

# From Universal Morphisms to Megabytes: A Baayen Space Odyssey



Editors: Krzysztof Apt, Lex Schrijver and Nico Temme

**From Universal Morphisms  
to Megabytes:  
A Baayen Space Odyssey**





# From Universal Morphisms to Megabytes: A Baayen Space Odyssey

Editors: Krzysztof Apt, Lex Schrijver and Nico Temme

CWI, Amsterdam, 20 December 1994,

on the occasion of the retirement of

Prof.dr. P.C. Baayen

from the Stichting Mathematisch Centrum





**Editors**

Krzysztof R. Apt  
Alexander Schrijver  
Nico M. Temme

**Executive editor**

Miente Bakker

**Printing and binding**

Rudy de Leeuw  
Jan Schipper  
Wim Tossijn  
Jos van der Werf

**Cover design**

Tobias Baanders

**Photographs**

Sjoerd Mullender  
Baayen Family Archives  
Photo Archive SMC

**Publication date**

December 20, 1994

Copyright © 1994 Stichting Mathematisch Centrum

Each separate contribution:

Copyright © 1994 the author(s)

Printed at CWI

Postbus 94079, 1090 GB Amsterdam

Kruislaan 413, 1098 SJ Amsterdam

Telephone +31 20 592 9333

Telefax +31 20 592 4199

ISBN 90 6196 450 4

## Contents

<i>Editorial</i>	ix
<i>Een woord vooraf</i> door G.Y. NIEUWLAND, voorzitter Curatorium SMC	xi
<i>Curriculum Vitae Pieter Cornelis Baayen</i>	3
<i>Promovendi Prof.dr. P.C. Baayen</i>	5
<i>Dat was volkomen abracadabra voor me, maar het leek me wel heel erg interessant</i> — Een interview met Cor Baayen, door Lex Schrijver	7

### Scientific Contributions

KO ANTHONISSE AND JAN KAREL LENSTRA <i>Operational Operations Research at the Mathematical Centre</i>	59
KRZYSZTOF R. APT AND FRANK TEUSINK <i>Comparing Negation in Logic Programming and in Prolog</i>	65
F. ARBAB AND I. HERMAN <i>The Manifold Coordination Language</i>	85
R. HARALD BAAAYEN <i>The randomness assumption in word frequency statistics</i>	125
HENK BARENDREGT <i>Discriminating coded lambda terms</i>	141
A. BENSOUSSAN <i>New Trends in Applied Mathematics</i>	153
JOHAN VAN BENTHEM <i>A New World Underneath Standard Logic: Cylindric Algebra, Modality and Quantification</i>	179
JAN BERGSTRA, JAN HEERING, AND JAN WILLEM KLOP <i>Introductory note to "Object-Oriented Algebraic Specification"</i>	187
J.A. BERGSTRA, J. HEERING, AND J.W. KLOP <i>Object-Oriented Algebraic Specification: Proposal for a notation and 12 examples</i>	188
O.J. BOXMA <i>Polling systems</i>	215
A.E. BROUWER <i>Finite graphs in which the point neighbourhoods are the maximal independent sets</i>	231
DICK C.A. BULTERMAN <i>A Framework for Adaptive Networked Multimedia</i>	235

ARJEH M. COHEN	
<i>Yet Another Lecture on the Icosahedron</i>	247
JEAN-ANTOINE DÉSIDÉRI, PIETER W. HEMKER, BARRY KOREN, AND MARIE-HÉLÈNE LALLEMAND	
<i>Research in Computational Fluid Dynamics, Stimulated by ERCIM</i>	269
JAN VAN EIJCK	
<i>A Natural Term Language</i>	287
PETER VAN EMDE BOAS	
<i>The full non-renameability result; a lost tale</i>	301
LOUK FLEISCHHACKER	
<i>Mathematics as the Paradigm for Metaphysics</i>	311
MICHEL HAZEWINKEL	
<i>The Wouthuizen Equation</i>	323
JAN HEERING AND PAUL KLINT	
<i>Prehistory of the ASF+SDF System (1980–1984)</i>	341
I. HERMAN, P.J.W. TEN HAGEN, AND G. REYNOLDS	
<i>Premo: An ISO Standard for a Presentation Environment for     Multimedia Objects</i>	347
P.J. VAN DER HOUWEN	
<i>On the History of Runge-Kutta Methods</i>	363
MICHAEL KEANE	
<i>The Essence of the Law of Large Numbers</i>	377
P. KLINT	
<i>Het Europese ESPRIT Programma: Een Persoonlijk Perspectief</i>	383
TOM H. KOORNWINDER	
<i>Special Functions Associated with Root Systems: Recent Progress</i>	391
A.A.M. KUIJK, P.C. MARAIS, AND E.H. BLAKE	
<i>Adaptive Spline-Wavelet Image Encoding and Real-Time Synthesis     on a VLSI Difference Engine for Image Generation</i>	405
WALTER M. LIOEN AND JAN VAN DE LUNE	
<i>Systematic Computations on Mertens' Conjecture and Dirichlet's     Divisor Problem by Vectorized Sieving</i>	421
JAN VAN MILL	
<i>Actions on the Hilbert cube</i>	433
HENRY MARTYN MULDER	
<i>The Expansion Theorem for Median Graphs</i>	437
ARD OVERKAMP AND JAN H. VAN SCHUPPEN	
<i>Control of Discrete Event Systems – Research at the Interface of     Control Theory and Computer Science</i>	453
ARJAN PELLENKOF, CÉSAR GALINDO-LEGARIA, AND MARTIN KERSTEN	
<i>Fast, Randomized Join-Order and Join-Method Selection Combined     with Transformation Based Optimization</i>	469

H.J.J. TE RIELE		
	<i>Job scheduling on a parallel shared memory bus computer</i>	485
ALEXANDER SCHRIJVER		
	<i>Rambling along paths, trees, flows, curves, knots, and rails</i>	493
ARNO SIEBES		
	<i>Data Mining: Exploratory Data Analysis on Very Large Databases</i>	535
N.M. TEMME		
	<i>Bernoulli Polynomials Old and New: Problems in Complex Analysis and Asymptotics</i>	559
A.S. TROELSTRA		
	<i>Kleene's Realizability</i>	577
FRITS W. VAANDRAGER		
	<i>Verification of a Distributed Summation Algorithm</i>	593
J.G. VERWER AND B.P. SOMMEIJER		
	<i>Stability Analysis of a Difference Scheme for Three-Dimensional Advection-Diffusion Problems</i>	609
PAUL VITÁNYI		
	<i>Randomness</i>	627



## Editorial

This book is offered to Cor Baayen, whose association with the Stichting Mathematisch Centrum (SMC) has lasted for over 35 years. From 1959 until 1965 Cor worked as a scientific researcher at the Department of Pure Mathematics. In 1965 he was appointed head of this department, and also professor of mathematics at the Free University in Amsterdam. In 1980 he became Director of the Stichting Mathematisch Centrum, which position he has held until now.

In the course of the time the ship he was the captain of changed its name — from *Mathematisch Centrum* to *Centrum voor Wiskunde en Informatica* — and traveled not only in time but also in space — from an old-fashioned and run-down school-building at the Tweede Boerhaavestraat to the friendly and comfortable manor in the Watergraafsmeer, below sea level. It also considerably grew, among others by further embracing computer science and by expanding its work in applied mathematics.

Cor steered this ship vigorously through ever-changing waters, sometimes with a full wind behind, sometimes against the wind. At his initiative a number of research areas were initiated or stimulated at CWI, like discrete mathematics, computational linguistics, computer algebra, cryptography, image analysis, performance analysis, interface technology. Thanks to his efforts the INSP-support became available for CWI, and the most successful and continuously growing conglomerate of European research institutes in mathematics and computer science — ERCIM — was created.

His enthusiasm for and interest in the research carried out at CWI can be illustrated best by the fact that he could effortlessly give overview lectures about the scientific work carried out at the institute. Started as a pure mathematician (with a thesis called “Universal Morphisms”), he has spent tireless efforts to get acquainted with the latest developments in mathematics and computer science — which he saw as essential for a Director of SMC —, so as to become authoritative in both disciplines alike.

We are very grateful to all those who contributed to the realization of this book. We thank Miente Bakker for managing the editorial process, Coby van Vonderen for substantial secretarial support, Sjoerd Mullender for making photographs, Tobias Baanders for designing the cover, Rudy de Leeuw, Jan Schipper, Wim Tossijn, and Jos van der Werf for printing and binding the book (the largest project ever of CWI’s printing division), and all authors for their articles.

We thank them all also for observing the short deadlines we imposed. The fact that they all met these deadlines shows the esteem they hold for Cor and expresses their appreciation for his dedication to the advancement of science. The breadth spanned by the contributions, from pure mathematics to applied computer science, reflects very well the space created and inspired by Cor for performing fundamental and applied research. The book thus offers the reader a science non-fiction odyssey through Baayen Space.

Cor, by putting this book with so many diverse contributions into your hands we hope to sustain your interest in mathematics and computer science. In your heart you have always remained a scientist. And scientists never retire. They just withdraw with yet another scientific book into their arm chair.

Also on behalf of all contributors to this volume, we wish you all the best in the next stage of your life. You were associated with the Mathematisch Centrum for well more than half of both your life and that of the Centrum. The Centrum bears your imprint and it remains yours!

*Krzysztof Apt*

*Lex Schrijver*

*Nico Temme*

## Een woord vooraf

G.Y. Nieuwland  
voorzitter Curatorium SMC

Prof.Dr P.C. Baayen zal op 20 december 1994, D.V. — degenen die hem kennen weten dat deze toevoeging voor hem betekenis heeft — afscheid nemen als Wetenschappelijk Directeur van de Stichting Mathematisch Centrum.

Hij heeft deze functie sinds 1980 vervuld en bepaalde in die periode nationaal en internationaal in veel opzichten het gezicht van de Stichting en haar instituut CWI.

Voor het wetenschappelijk directoraat bestaan in principe twee modellen: bij het eerste ligt het accent op de voorbeeldfunctie van de eigen wetenschappelijke prestatie van de leider van de organisatie, bij de tweede op zijn functie als stuurman. Ik vermoed dat ook de eerste rol Baayen goed gelegen zou hebben. Maar er viel voor hem weinig te kiezen: al kort na zijn optreden als directeur werd duidelijk dat met name het instituut van de Stichting respons diende te geven op de uitdaging die vanuit de maatschappij werd gesteld. Daarmee ging het CWI een traject in dat aan de stuurmanskunst van de directie tot dusver ongekende eisen stelde. In deze bundel komt die kant van Baayen's werk alleen impliciet aan de orde; de redactie heeft ervoor gekozen juist de sporen te boekstaven die hij daarnaast — haast schreef ik: desuietegenstaande — in wetenschappelijk opzicht heeft getrokken.

Dit boek biedt daarvan een fraaie staalkaart — in vier categorieën. In de eerste plaats zijn daar de bijdragen van vrienden-collega's uit Baayen's wetenschappelijke land van herkomst: het brede gebied waarop de logica en de fundamentele algebraïsche, topologische en combinatorische structuren van de wiskunde in interactie zijn. U mag dit ook lezen als een *acte de présence* van dat deel van de SMC dat plaats vindt op het universitaire erf.

Zijn eigen visies en uitgangspunten worden, behalve in een interview, in deze bundel vooral gereflecteerd in de bijdragen van zijn promovendi — die hun eigen weg gingen en laten zien daarop een frontpositie te hebben bereikt.

Dan is er een breed overzicht van de wetenschappelijke productie van het Centrum voor Wiskunde en Informatica — representatief voor het onderzoeksprogramma waarvoor Baayen zoveel jaren een eerste verantwoordelijkheid droeg. U treft daaronder uiteraard veel informatica aan: het vakgebied waarvan de ontplooiing in Nederland zoveel aan zijn wetenschappelijk leiderschap, kennis en inzicht te danken heeft.



Tenslotte een bijdrage met een bijzonder karakter, met een onderwerp dat twee van zijn grote liefdes verenigt: de wiskunde en de taal, geschreven door een auteur die niet alleen wetenschappelijk in relatie met hem staat.

Alles bijeen een boek waaraan velen plezier zullen beleven, document ook van een stukje Nederlandse wetenschapshistorie — maar allereerst naar de bedoeling van zijn auteurs: *liber amicorum*.





# Curriculum Vitae

## Pieter Cornelis Baayen

geboren 10 maart 1934 in Klaten (Java, Indonesië)

### Opleiding

- 1951 eindexamen gymnasium- $\beta$ , Christelijk Lyceum voor Zeeland, Goes
- 1954 kandidaatsexamen wiskunde en natuurkunde, Vrije Universiteit, Amsterdam
- 1957 doctoraalexamen wiskunde (uitgebreid) met meteorologie, Vrije Universiteit, Amsterdam
- 8 juli 1964 promotie tot doctor in de wiskunde en natuurwetenschappen, Universiteit van Amsterdam  
promotor: Prof.dr. J. de Groot  
titel dissertatie: Universal Morphisms

### Loopbaan

- 1956–1957 leraar wiskunde aan de dependance te Heemstede van het Tweede Christelijk Lyceum te Haarlem
- 1957–1959 verblijf aan de University of California at Berkeley, met een Fellowship van de International Cooperation Administration
- 1959–1995 verbonden aan het Mathematisch Centrum:  
1 oktober 1959: medewerker afdeling Zuivere Wiskunde  
1 december 1962: souschef afdeling Zuivere Wiskunde  
1 juni 1965: chef afdeling Zuivere Wiskunde en lid Raad van Beheer  
1 september 1980: wetenschappelijk directeur
- 1960–1965 docent Nutsseminarium voor Pedagogiek, Amsterdam (avondopleiding M.O.)
- 1962–1965 leeropdrachten Vrije Universiteit (functionaalanalyse, topologie)
- vanaf 1965 hoogleraar aan de Vrije Universiteit, Amsterdam:  
1 januari 1965: buitengewoon hoogleraar  
1 juni 1965: gewoon hoogleraar
- 1966–1967 gasthoogleraar University of Washington, Seattle

**Verder o.a.**

- 1978-1980 voorzitter Wiskundig Genootschap  
1979-1986 voorzitter Vereniging voor Christelijk Voortgezet Onderwijs te Utrecht  
1983-1985 vice-voorzitter Nederlands Genootschap voor Informatica  
1991-1994 president European Research Consortium for Informatics and Mathematics (ERCIM)

## Promovendi Prof.dr. P.C. Baayen

N.P. Dekker

Vrije Universiteit, 16 mei 1969

*Joint numerical range and joint spectrum of Hilbert space operators*

P. van Emde Boas

Universiteit van Amsterdam, 18 september 1974

*Abstract resource-bound classes*

(copromotor, naast prof.dr.ir. A. van Wijngaarden)

J. de Vries

Vrije Universiteit, 13 december 1974

*Reflections on topological transformation groups*

W.J. de Schipper

Vrije Universiteit, 20 december 1974

*Symmetric closed categories*

J. van Mill

Vrije Universiteit, 17 juni 1977

*Supercompactness and Wallman spaces*

A. Schrijver

Vrije Universiteit, 3 november 1977

*Matroids and linking systems*

H.M. Mulder

Vrije Universiteit, 12 september 1980

*The interval function of a graph*

L.E. Fleischhacker

Universiteit van Amsterdam, 24 september 1982

*Over de grenzen van de kwantiteit*

(copromotor, naast prof.dr. J.H.A. Hollak)

J.C.S.P. van der Woude

Vrije Universiteit, 16 december 1982

*Topological dynamix*

J. Vermeer

Vrije Universiteit, 19 mei 1983

*Expansions of  $H$ -closed spaces*

In 1995 wordt verwacht:

G. Alberts  
Universiteit van Amsterdam  
*Jaren van berekening*  
(met prof.dr. J.H.C. Blom)

# Dat was volkomen abracadabra voor me, maar het leek me wel heel erg interessant

Een interview met Cor Baayen  
door Lex Schrijver

*Laten we beginnen bij je geboortejahr 1934.*

Ik ben geboren in Klaten. Klaten ligt op Midden-Java, in de “vorstenlanden”, halverwege Solo (of Surakarta) en Yogya (of Yogyakarta). Solo is de zetel van de susuhunan, de “keizer”, en Yogya de zetel van de sultan, uit een zijtak van het vorstelijk huis, welke later veel machtiger is geworden dan de tak van de susuhunan. Ook nu heeft Yogya nog steeds een mate van onafhankelijkheid.

Mijn ouders waren beiden verbonden aan het Christelijk onderwijs. Mijn vader is als onderwijzer naar “Nederlandsch Oost-Indië” gegaan, heeft aktes erbij gehaald, en is hoofd geworden van een zogenaamde schakelschool, in Solo. Als ik het goed begrepen heb, is dat een school waar leerlingen die het goed gedaan hadden op het “onderwijs voor inlanders”, bijgewerkt werden om naar de middelbare school te gaan.

Hij heeft toen weer aktes erbij gehaald en is naar een Mulo gegaan, en vrij kort voor de oorlog, ook middelbare aktes, waarna hij leraar geschiedenis is geworden in Jakarta, of Batavia, zoals dat toen nog heette.

Ook mijn moeder is als onderwijzeres naar Indië gegaan (ze is ‘met de handschoen’ met mijn vader getrouwd), en heeft daar een tijd les gegeven, maar — zoals in die tijd gebruikelijk — toen het eerste kind kwam hield ze op met werken.

*Dat was jij?*

Nee, dat was ik niet. Dat is een eerder kind geweest, een meisje, dat is overleden voordat ik geboren ben, en begraven in Klaten. Daar ligt ook een broertje van me begraven, die is in de eerste oorlogsmaanden ziek geworden, na de landing van de Japanners. Er kon geen hulp verleend worden, hij had difterie, en hij was binnen een week overleden.

Hij was jonger dan ik. Ik ben de oudste, het tweede kind van mijn ouders, maar de oudste overlevende. Ik heb een jongere broer, dan een zuster, en dan dat broertje wat overleden is, en dan nog weer een zusje van voor de oorlog, en na de oorlog heb ik nog twee broertjes gekregen. Ik noem ze nog steeds broertjes hoewel de jongste daarvan nu toch ook al over de veertig is.





... uiteraard zijn de vroegste herinneringen het verste weg ...

### *Waar kwamen je ouders vandaan?*

Er is vrij veel bekend van het voorgeslacht van mijn ouders. Niet dat ik zelf ooit aan genealogisch onderzoek gedaan heb, maar voor mijn vaders voorgeslacht heeft een achterneef dat gedaan en van mijn moeders zijde heeft een broer van mijn moeder heel veel onderzoek gedaan.

Mijn vaders familie is terug te voeren tot Jan Janszoon Baaij, die in 1456 uit Antwerpen naar Bergen op Zoom kwam, en daarna is dit geslacht in Bergen op Zoom blijven wonen. Daar zijn ze in de tijd van de Doleantie meegegaan met de afscheiding uit de Hervormde kerk, naar wat later de Gereformeerde kerk geworden is.

Mijn moeders geslacht — zij heten Minderhoud, dus oorspronkelijk komen ze misschien wel uit Minderhout, vlak over de grens in België. De naam is heel rijk vertegenwoordigd in West-Kapelle op Walcheren. Mijn moeder is geboren op Zuid-Beveland.

Ik heb een grote kerkbijbel van mijn ouders, formaat statenbijbel, in leer gebonden en gedrukt in 1881, en aangeschaft door mijn overgrootvader. Die was toen lid van de Gereformeerde kerk en is later overgegaan, begrijp ik, naar de Gereformeerde Gemeente. Dat was de vader van mijn moeders moeder. Maar mijn moeders vader was weer gereformeerd. Traditioneel behoorden beide kanten van mijn onmiddellijke voorgeslacht tot de Gereformeerde kerk.

*Wat herinner je je nog van je eerste jaren?*

Toen ik nog geen jaar oud was zijn we naar Nederland gegaan, mijn ouders hadden er zo'n zeven jaar opzitten en kregen één jaar verlof. Zelf herinner ik me hiervan uiteraard niets.

Ik ben dus 1 jaar geworden in Nederland. Na die lange verlofperiode zijn we weer naar Indië gegaan. Ook die bootreis naar Nederland en terug herinner ik me niet, we waren een volle maand onderweg.



... die bootreis naar Nederland en terug herinner ik me niet ...

Van voor de oorlog herinner ik me niet zo erg veel. Ik heb na het concentratiekamp een tijd lang blokkeringen gehad waardoor ik een aantal dingen

heel lang niet meer geweten heb. Die zijn later na een periode van ziekte weer teruggekomen.

Ja, ik herinnerde me het land toen ik daar voor het eerst drie jaar geleden terug kwam en in Jakarta uit het vliegtuig stapte. Toen dacht ik: deze geuren herinner ik me; ik stond nog boven aan de vliegtuigtrap.

En toen ik voor het eerst in Yogya en Solo terugkwam had ik een heel sterk déjà vu gevoel, die witte kratonmuren bijvoorbeeld. Er zijn visuele en olfactorische herinneringen die meteen geactiveerd werden toen ik daar terugkwam.

Maar ik herinner me niet zo erg veel van het leven. Uiteraard zijn de vroegste herinneringen het verste weg.

*Merkte je iets van spanningen tussen inlanders en Nederlanders?*

Voor de oorlog denk ik dat ik politieke spanningen niet gemerkt zou hebben als die er waren. Rondom mijn ouderlijk huis waren die er niet. Mijn ouders hadden geregeld studenten in de kost die zoals dat heette uit de buitengewesten kwamen, Celebes, Sumatra, en die in Jakarta kwamen studeren of daar naar school gingen, en mijn ouders hadden daar goede relaties mee. We hadden ook personeel. Ik was erg bevriend met het zoontje van onze djongos, we speelden altijd samen. In en om ons gezin waren er geen spanningen, en politieke spanningen in het groot zijn mij als kind van toen zeven jaar outgaan.

*Toen kwam de oorlog.*

Ik herinner me dat wij, na het uitbreken van de oorlog in Europa, in 1941 als kinderen langs kennissen werden gestuurd, met het rijmpje “Volgend jaar 10 mei, is Nederland weer vrij”. Zo ging je rond om elkaar heil toe te wensen.

Het jaar daarop, in 1942, waren we inmiddels in heel andere omstandigheden gekomen. Want, zoals bekend, de aanval van de Japanners op Pearl Harbour was op 7 december 1941, en de landing van de Japanners op Java was in de nacht van 28 februari op 1 maart 1942. Toen verbleven wij al niet meer in ons huis in Jakarta. Mijn vader was als landstormer, zoals dat heette, als dienstplichtig soldaat opgeroepen, en had zijn gezin naar Klaten gestuurd, mijn geboorteplaats, waar een oom van mij hoofd van de Hollands-Chinese school was, en daar konden wij logeren.

En wat ik mij daarvan nog zeer wel herinner is, dat dat niet zo erg lang voor mijn verjaardag was, 10 maart. Ik denk, dat we, na Pearl Harbour, in de loop van januari-februari naar Klaten gegaan zijn, en ik zou op mijn verjaardag een grote meccanodoos krijgen; die stond al in huis. Die had ik al eens mogen zien, maar ik mocht er niet aan komen, want die was immers voor mijn verjaardag; en die heb ik nóóit gekregen. Dat is een van de grote trauma's in mijn jeugd. Ik heb het idee dat ik tegenwoordig technisch Lego koop voor mijn kleinkinderen omdat ik zelf destijds die meccanodoos niet gehad heb.

*De oorlog kwam ook net in je lagere schooltijd, neem ik aan.*



... ik was erg bevriend met het zoontje van onze djongos ...

Ik zat net kort in de derde klas toen de oorlog uitbrak. Dat heb ik in zoverre nooit ingehaald, dat ik nooit goed lagere schoolonderwijs gehad heb.

Wij zijn de oorlog begonnen in een kamp in Sumuwono, dat heb ik teruggevonden toen ik daar drie jaar geleden terug was. De barakken staan er nog, alleen die zijn geweldig klein geworden vergeleken met vroeger. Maar dat kamp is onmiskenbaar, door zijn ligging op een heuvel, met allemaal trappen.

In dat kamp hadden we het nog redelijk goed. Mijn moeder was zoals gezegd onderwijzeres, en met andere dames in het kamp organiseerde ze clandestiene klasjes. In een schuur, met achterkanten van kasten als schoolbord — het heeft wel het nadeel dat je het krijt haast niet kunt uitvegen. Er werden, hoewel misschien niet nodig, allemaal lakens gewassen en op lijnen gehangen en tussen die lakens werden dan klaslokaaltjes uitgespaard. En een van de kinderen werd op de uitkijk gezet, want het mocht eigenlijk niet. En als er dan een Japanezische kampbewaker langskwam of een heiho'er (dat waren Javaanse hulpbewakers), dan werd er een of ander afgesproken sein gegeven, en verspreidden we ons. Ik heb zo dus wel wat les gehad, op het kampschooltje.

Ik denk dat we na ongeveer een jaar van daaruit verplaatst zijn naar een van de beruchte kampen in Ambarawa; daar was een groot aantal concentratiekampen. En wij kwamen terecht in Kamp 7. Vanuit dat kamp zijn eind 1944 de

vrouwen en kinderen, waaronder ook mijn moeder en broer en zusjes, verplaatst. Er bleven zo'n 700 jongens achter van tussen de 10 en 13 jaar.

Als je tien jaar was, gold je voor de Japanner als volwassen, en als je van het mannelijk geslacht was, was je dus gevaarlijk en mocht je niet bij de vrouwen en kinderen blijven maar moest je naar een mannenkamp. Dat is niet gebeurd onmiddellijk nadat ik tien jaar geworden ben, 10 maart 1944, maar pas aan het eind van dat jaar. Ik heb dus een klein jaar in een mannenkamp gezeten, jongens en mannen, en daar werd niets meer aan onderwijs gedaan.

Oudere jongens waren al een keer eerder weggehaald en er kwamen zo'n 2000 mannen bij, dat waren zieken en invaliden uit omliggende kampen, dwangarbeiderskampen, krijgsgevangenenkampen, en die moesten wij, 700 jongens, verzorgen. In dat kamp heb ik van eind 1944 tot na de capitulatie, augustus-september 1945, gezeten.

Ik zat daar dus zonder te weten of mijn vader of moeder en broer en zusjes nog leefden, en zo ja, waar die dan wel zaten. Via het Rode Kruis kreeg je een enkele keer wel eens een levensteiken, maar er was heel weinig communicatie.

*Hoe kijk je op die tijd terug? Als jongen kun je natuurlijk een hele hoop dingen ook heel spannend vinden.*

Ik denk dat ik daar net iets te jong voor was. Enerzijds neem je het leven heel serieus als je hoofd van de huishouding bent, al is het maar een huishouding van één 10-jarig persoon, maar anderzijds was ik, denk ik, gewoon te jong om het als iets spannends te ervaren. Ik heb het veeleer als iets serieus, iets verantwoordelijks, en toch ook als iets beklemmends ervaren.

En vergeet niet dat je in toenemende mate uitgeput raakte door ondervoeding. Er waren ook heel veel sterfgevallen en het aantal doden per etmaal steeg in de loop van de maanden. En wij moesten als jongens al het werk doen, het corvee, de ziekenzorg, het begrafeniscorvee, maar ook alle beschikbare land ontginnen, ook nog varkens vet mesten voor de Japanner, en kippen en eenden verzorgen voor de Japanner, want die at wat beter dan wij en hield van een vers eitje op zijn tijd. Ik heb een tijd lang in het keukencorvee gezeten, groenten schoonmaken. Dat nam je heel serieus, maar aan het eind raakte je, tenminste ik, sterk apathisch door de ondervoeding. Ik herinner mij zeer wel dat als het corvee afgelopen was, om een uur of drie of zo, ik in het zonnetje tegen de muur ging zitten, in de tropenhitte. Je was zo ondervoed dat je zelfs de warmte van de tropenzon nodig had, als een soort energie-inbreng als het ware. Dan zat je daar maar te wachten tot er een gong ging dat er eten gehaald kon worden.

Ik ben die kampherinnering ook een hele tijd grotendeels kwijt geweest. Onderdrukt. Ik ben een keer overspannen geraakt, een tijd van slag geweest, 1969-1970. Toen heb ik een heel slechte tijd gehad, met veel onrust en nachtmerries, en toen zijn met brokken, mozaïekachtig, geleidelijk allerlei herinneringen teruggekomen die ik dus zo'n 25 jaar weggedrukt had, niet bewust overigens.

*En dat had je misschien ook nodig op dat moment?*

Ik denk het wel. Als je opgroeit en probeert weer normaal mens te worden dan moet je die dingen niet in je toegankelijke geheugen iedere keer tegenkomen. Dus ik denk dat dat vrij normaal is.

Achteraf ben ik van mening, dat geldt voor het hele gezin — daar heb ik het onlangs nog met mijn moeder over gehad —, dat wij met z'n allen, mijn broer en zusters, mijn ouders, die kampervaring heel constructief verwerkt hebben. Wij hebben daar voorzover ik kan nagaan geen trauma's, geen psychische nadelen van over gehouden. Mijn vader kon er met veel humor over vertellen. Die kon ook over allerlei nare dingen in het kamp zo vertellen dat je erom lachen moest, ook ik. Maar goed, ik ben te jong geweest om dat zelf te kunnen relativieren. Ik kan dat nog steeds niet, maar ik kan er wel met grote gelijkmoedigheid op terugzien. Ik heb daar echt geen problemen mee.

Het is overigens wel zo dat, toen ik voor het eerst naar Japan ging, dat toch een ervaring was die ik niet helemaal emotieloos doormaakte. Maar dat is ook gauw over.

*Hebben je oorlogservaringen je sneller volwassen gemaakt?*

Ik denk het eigenlijk wel, ja. Ik heb mij nooit meer kind gevoeld. Je bent verantwoordelijk geweest voor je eigen bestaan, en dat leg je nooit meer helemaal af. In tegenstelling tot mijn broer en zusters die allemaal jonger zijn dan ik en die bij mijn moeder gebleven zijn. Maar die hebben natuurlijk ook het nodige meegemaakt, die hebben ook geen normale jeugd gehad, maar die hebben de dingen toch weer anders verwerkt dan ik.

Ik denk achteraf dat het een waardevolle ervaring is geweest, op zijn manier. Maar het is wel zo dat ik me er heel lang niet in heb willen verdiepen. Dat ik niets wilde lezen over de oorlog en over het gebeuren in het kamp. Ook niet over de oorlog hier in Europa hoor, ik wilde daar eigenlijk niet mee geconfronteerd worden.

Enerzijds is er een puur psychisch automatisme geweest waardoor een heleboel dingen weggedrukt zijn, Freud zou wel kunnen verklaren waarom. Maar anderzijds wilde ik er ook zo weinig mogelijk bewust mee bezig zijn. Ook dat wordt op den duur milder. Op een gegeven ogenblik ben je toch nieuwsgierig wanneer het ook al weer precies was, en dan ga je weer een paar dingen opzoeken en nalezen. Ik heb eens een keer een boek gekocht over de krijgsgevangenenkampen in Indonesië. Dat was trouwens toen ik drie jaar geleden daar voor het eerst weer naar toe ging, toen wilde ik zoveel mogelijk van dit soort plaatsen terugvinden, en het zit me nog steeds dwars dat ik dat kamp in Ambarawa niet teruggevonden heb. Ik wist dat het er niet meer was, hoor, ik wist dat het helemaal verwijderd is, maar ik had zo'n gevoel van als ik die plek kan terugvinden dan moet ik dat toch kunnen herkennen. Maar ik heb daar geen enkel gevoel van herkenning gehad. Sumuwono wel, en daar ben ik blij om, dat ik dat



heb kunnen vinden, dat was heel plezierig, toch heel goed om daar weer eens te lopen, vooral omdat we met z'n tweeën waren, mijn zus en ik, zij herinnerde zich weer andere dingen dan ik, en ja, dat was goed.

Ouderen die het meegemaakt hebben, dat merk ik wel, praten er makkelijker over. Ik heb het toch, laat ik zeggen, heel animaal meegemaakt, je was aan het overleven en je leefde in deze omstandigheden, er was niets bijzonders aan, leuk was het niet, extra bedreigend, zo ervoer je het ook niet — je werd handig in het overleven, en je zorgde voor jezelf, en verder de orde van de dag.

Ik heb eens een keer 's nachts meegedaan aan een inbraak bij het varkensvoer, voor de varkens van de Jappen, dat was gemalen en geperste kaf van rijst, gabbah. Ik heb een stukje veroverd en verstoep, zoiets was riskant. Als je gesnapt werd dan kreeg je een behoorlijk zware straf van de Japanners. Maar ik ben niet gesnapt en dat heeft mij wat bijvoeding gegeven enige tijd, en dat beschouwde je helemaal niet als avontuurlijk, als leuk, interessant of dapper, of wat dan ook, nee, dat was gewoon een stukje overleven.

Terwijl ik merk van verslagen van ouderen — die hebben het veel bewuster meegemaakt, en hebben veel meer geweten wat ze deden. Twee of drie jaar ouder maakt al veel verschil. Ik bestond, ik overleefde.

*Je zag veel doden om je heen. Dat zegt je dan misschien ook heel weinig meer.*

Ja. Er was in het kamp een bord en daarop werden de namen opgeschreven van diegenen die de afgelopen 24 uur overleden waren. En of het precies zo gebeurd is dat weet ik niet, maar zo herinner ik het mij, de commandant had er op een gegeven ogenblik een premie op gezet dat als we voor het eerst een bepaald aantal haalden, 20 of zo, dan zouden we extra eten krijgen. Toen stonden we dus voor het bord: Ach, er zijn er weer maar 17 dood; en op een gegeven ogenblik: Ha! het zijn er 21, en toen kregen we een extra portie eten. En de dag daarop kregen we weer helemaal niets, want ja, die commandant moest met zijn budget rondkomen.

De verjaardag van Tanno Heika, de keizer, werd gevierd en dan kregen we vlees. Een of twee honden voor het hele kamp. Die gingen in de soep en als je geluk had dan vond je een draadje.

Ik heb gelezen dat de Japanse soldaten heel braaf waren in het opvolgen van orders. Aan het begin van de oorlog is er een bepaald bedrag, zoveel cent per dag, per gevangene vastgesteld voor voedsel, en daar hebben ze zich aan gehouden. Alleen, in de loop van de oorlog had je een inflatie van een paar honderd procent, en Tokyo is er nooit aan toe gekomen dat bedrag bij te stellen. En dus ging ons voedsel met factoren naar beneden, want het bleef tot het einde van de oorlog zoveel cent per dag. Aan het eind kon je er maar weinig meer voor krijgen.

*Hoe was het einde van de oorlog?*

Na de oorlog ben ik vanuit Ambarawa op clandestiene wijze liftend dwars door de linies heen, levensgevaarlijk, bij mijn moeder terecht gekomen, die in Semarang zat, in een vrouwenkamp. Pas na maanden zijn wij verenigd met mijn vader die in een krijgsgevangenenkamp in Bandung zat.

Dat was ook mijn eerste vliegtocht in zo'n groene leger-Dakota. Zo'n soort vliegende aluminium vuilnisbak, waar je op klapstoeltjes zat met een grote papieren zak, omdat iedereen zat over te geven en onpasselijk te zijn. Maar daar heb ik gelukkig nooit last van gehad, want ik vond het zelf wel prachtig. Ik geloof dat ik de enige van het gezin was die alles binnen hield.

Na de oorlog ben ik toen meteen maar in de tweede helft van de eerste klas HBS begonnen. Dat was misschien niet helemaal volgens de regels, maar mijn vader was weer les gaan geven, in Bandung. Hij was, aan een school met zo'n 600 leerlingen, de enige bevoegde geschiedenisleraar. Hij had een team van mensen die toch werkloos waren, een kapper, een boekhandelaar die alles kwijt was, en die gaf hij dan instructies in wat ze moesten doen. Eén of twee schoolboeken waren er, en die circuleerden dan de hele klas rond.

Mijn vader had dus nogal een sleutelpositie en wist mij, toen het gezin eindelijk herenigd was — dat had ook nogal wat voeten in de aarde —, op school ingeschreven te krijgen. Ik had toen een rapport waarbij ik volgens alle regels had moeten blijven zitten, maar ook daar werd soepel over gedaan, en mijn moeder heeft me geweldig bijgestoomd, iedere dag in de vakantie hard werken aan allerlei lessen, en zo ben ik in de tweede klas gekomen. Toen ging het al wat beter — in het land der blinden is eenoog koning, en ik had een goed thuisfront.

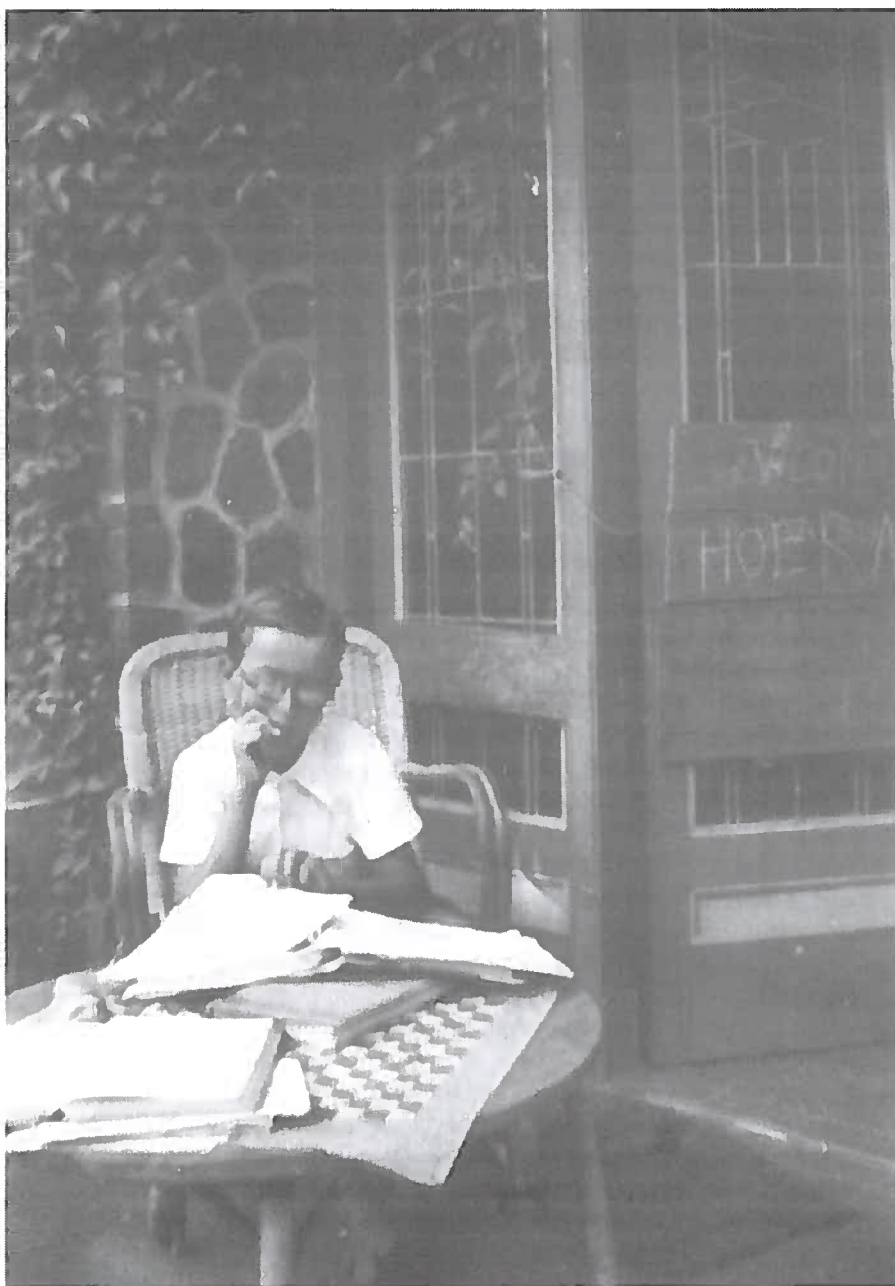
In die tijd merkte ik wel wat van de spanningen tussen inlanders en Nederlanders. Dwars door Bandung liep de demarcatielijn. Ik heb daar de granaten door de straat horen gieren, dan sloegen ze een eindje verderop in, en dan kwam er weer een.

Je kon dus ook de stad niet uit. We zijn één keer naar de Dago-waterval geweest met een Nederlandse militair, er waren militairen van de politionele acties en mijn ouders stelden daar hun huis onmiddellijk voor open. Dus die kwamen bij ons koffie drinken en eten en gezelligheid zoeken, en een keer heeft zo'n militair, gewapend, ons meegenomen naar de Dago-waterval. Toen was dat gevaarlijk terrein, daar kon je pemuda's tegenkomen, daar kon je niet naar toe zonder militair met geweer. Ik was overigens te jong om dat echt als spannend te ervaren. Dat was gewoon een randvoorwaarde van het bestaan.

Toen probeerde men in Bandung het Christelijk Lyceum weer op te richten, aan wat toen de Dagoweg heette. Ook daar ben ik drie jaar geleden teruggeweest en heb de school herkend, ook die was veel kleiner geworden.

Maar om die school destijds weer opnieuw te kunnen oprichten was er een minimum aantal leerlingen nodig en mijn vader, grootmoedig als altijd, gaf mij op voor die school aan de Dagoweg. Toen zat ik ineens in het Gymnasium, dat was bijna een uur lopen van waar wij woonden — heen, en een uur terug, iedere dag. Nou daar zat ik nog maar net een paar maanden op en toen gingen we naar Nederland.





... iedere dag in de vakantie hard werken aan allerlei lessen ...

### *Waarom zijn jullie teruggegaan naar Nederland?*

Mijn vader was met hart en ziel leraar, maar hij is in het kamp praktisch blind geworden en kon niet alles wat een gezonde, valide leraar kon. Op een gegeven ogenblik gaf dat een conflict. De leiding van de school vroeg meer van hem dan hij kon opbrengen. Toen heeft hij zich laten keuren en werd hij onmiddellijk afgekeurd en naar Nederland gestuurd met de eerstvolgende boot.

Dat was november 1947, en die tocht op de "Oranje" was een geweldige belevenis, één van de hoogtepunten in mijn leven. Een lijnboot als de "Oranje" was een prachtig stuk techniek, erg indrukwekkend. Wij, mijn jongere broer en ik, hadden de vrijheid om overal naar toe te gaan. Mijn zusje daaronder was net te jong en moest bij de kinderopvang blijven. Maar wij konden over de hele boot. Maar het allerbelangrijkste was dat er genoeg te eten was. We waren geweldig ondervoed uit het kamp gekomen, en dat eerste anderhalf jaar na de oorlog was nog steeds een periode van grote zuinigheid, en van, laten we zeggen, ook geen bijzonder rijke voeding. En daar aan boord heb ik voor het eerst bewust appels en peren gegeten, en havermout en van alles en nog wat, dat was een heus paradijs.

Wij zijn als hele kleine ondervoede magere scharminkeltjes naar Nederland gekomen, en dat was toch al eind 1947. We kwamen aan in het begin van de winter, 30 november of zoiets, staat me bij, want de volgende dag zag ik voor het eerst hagel, geen idee wat dat was. Er lag ineens wit grind, dat er de vorige nacht niet geweest was. Ik kreeg dan ook prompt binnen twee maanden een zware longontsteking. Nadat ik die overleefd had ben ik in een jaar van de kleinste van de klas tot de allerlangste van de klas doorgeschoten en heb een heleboel ingehaald in lengte en breedte en massa.

### *Waar heb je toen gewoond en ben je naar school gegaan?*

We woonden in Bergen op Zoom, maar ik zat op school in Goes. Aan die school, het Christelijk Lyceum voor Zeeland, heb ik heel goede herinneringen. Er kwamen leerlingen uit heel Zeeland. Het was boeiend, ik ging graag naar school.

Er was één nadeel aan de situatie. Vanuit Bergen op Zoom moest ik 's morgens al om 6 uur met de stoomtrein mee want de trein van 8 uur was te laat, dan kwam ik pas om 9 uur op school. En ik kwam 's avonds laat terug en ik had natuurlijk mijn huiswerk.

Ik kwam niet toe aan een sociaal leven in Bergen op Zoom, behalve zondags bij mijn grootmoeder, koffie met ontbijtkoek, en dat was ook niet zo erg sociaal op mijn niveau. Terwijl ik in Goes ook buiten de vriendenkringen bleef, want ik kwam pas tegen schooltijd en ik ging meteen na school weg om de eerste trein te halen, want die treinen reden maar eens in de twee uur, dat was niet zo erg leuk. Ik ben dus een beetje zonder speelvrienden, zonder vrienden met wie je sociaal contact hebt, opgegroeid. Ik had wel vrienden op school maar die

woonden dan ook heel ergens anders. Een goede vriend woonde in Yerseke en die zag ik alleen maar op school.

Het Christelijk Lyceum voor Zeeland was in die tijd een kleine school, pas gestart, nog zonder erkenning, en dat was een beetje behelpen, wat lokatie betreft en wat docenten betreft. We zaten toen ik daar op school kwam (dat was halverwege de derde klas) in een oud gebouw, ik denk dat dat een oud weeshuis was of zo.

We waren als derde klas de hoogste klas, en we bleven dat, mijn hele schoolperiode, en dat gaf ons een bijzondere verantwoordelijkheid. Als we ons hadden misdragen tegenover een leraar — dat wil zeggen, dat vond die leraar, die ging zich beklagen — dan werden we bij de rector geroepen en die zei dan: “Jullie zijn de hoogste klas en jullie moeten het voorbeeld geven, daarom zal ik jullie nu geen straf geven maar ik reken erop ...”. Ja, het was een heel aparte sfeer.

Door mijn lacunaire ondergrond zakte mijn rapport weer volledig in elkaar, en toen herhaalde zich het spelletje. Ik had volgens alle regels moeten blijven zitten, met een 3 voor Latijn, een 4 voor Grieks en een onvoldoende voor dat, maar de rector in Goes had zelf in Indië gezeten en die wilde me wel een kans geven.

Weer een zomer heel hard geblokt en in de vierde klas heb ik toen geleidelijk aan mijn cijfers op weten te halen, en met de overgang van 4 naar 5 waren alle onvoldoendes verdwenen, en daarna is de zaak naar een redelijk niveau gebracht.

In die laatste klassen, 5 en 6 gymnasium, zaten we met z'n zevenen in de klas. Pas in mijn eindexamenjaar, in 1951, heeft de school erkenning gekregen. Er was nog een tijd lang sprake van dat wij op een andere school examen zouden moeten doen. Net op tijd is dat in orde gekomen.

Het was ook voor de leraren voor een deel een opoffering. De school was niet gesubsidieerd dus ik denk ook niet dat ze hetzelfde salaris gehad zullen hebben als de leraren aan een wel erkende en gesubsidieerde school. Maar het was heel intiem, doordat het klein was en doordat de leraren enthousiast waren. Ze probeerden je echt wat over te brengen. Ik heb daar heel positieve herinneringen aan.

*Welke vakken vond je leuk op school?*

Wiskunde vond ik erg leuk, ik had les van de heer Maas; die was K5-er maar een heel solide en betrouwbare leraar. Als gymnasiast kreeg ik ook analytische meetkunde, uit het boek van Schreck, als ik het goed heb. Daar kwamen coördinaat-transformaties aan de orde, maar alleen maar translaties, en dan wilde ik op een gegeven moment weten wat er gebeurde bij rotatie. Nou meneer Maas kwam met zijn exemplaar van Barrau en had daar een bladwijzer bij gedaan waar de coördinaat-transformaties stonden en leende dat aan mij uit, en dat bestudeerde ik dan, dat had hij voor K5 moeten doen. Hij kon het me niet uitleggen maar hij kon wel zijn boek uitlenen en dat vond ik erg leuk.

En zo waren er meer leraren die echt wat voor je deden. De leraar Duits, die

mij een boek over filosofie uitleende en die mij op het spoor zette van het oud-Duitse Nibelungenlied en dergelijke. En de leraar Nederlands, meneer Cornet, die geweldig enthousiast was, die ook altijd bereid was om zich te laten afleiden om over zinnige culturele onderwerpen met de klas te praten. Ik heb daar erg veel van geleerd, vooral door zijn enthousiasme. Door natuurkunde werd ik erg geboeid, door de leraar Hoogteijling.

Ook kwam er een leraar, de heer Mulder, die pas zijn ingenieursexamen in Wageningen had gehaald, zowel voor biologie als voor scheikunde, ook een enthousiast iemand, waar ik graag bij op les kwam. Biologie was een van mijn lievelingsvakken, dat was mijn enige 10 bij het eindexamen. Ik was eigenlijk ook van plan om biologie te gaan studeren, maar die enthousiaste leraar natuurkunde heeft me overgehaald om naar de VU te gaan voor natuurkunde, daar had hij zelf gestudeerd.

Het was een school die heel weinig had. Proeven konden er nauwelijks gedaan worden. Ik herinner mij nog dat er eens een fles kwik was aangeschaft, en toen kwam de werkster, die wou de fles oppakken en die had helemaal niet in de gaten dat dat kwik was. Ze schrok zo van dat gewicht en liet hem vallen en daar brak-ie. Gelukkig in de gootsteen, en uit de elleboog kon nog een deel van het kwik gered worden. Maar dat was bijna het hele practicumkapitaal dat daar werkelijk 'through the drain' ging.

Maar ach, mijn zwakke vakken waren de talen. Daar heb ik ook erg lacunair les in gehad en daar ben ik nooit sterk in geworden. Ik heb nooit goed Frans geleerd bijvoorbeeld, want de eerste leraar Frans, die is aan tbc overleden — daar stierf je toen nog aan, kort na de oorlog —, toen hebben we een tijd zonder leraar Frans gezeten, en toen kreeg de leraar geschiedenis opdracht om Frans te geven want die had in België gewoond, dus die was relatief deskundig. Dat was de heer Van Dijk, zijn vrouw is de bekende schrijfster van streekromans geweest, Nellie van Dijk-Has. Ik heb van mijnheer Van Dijk leuk geschiedenis gehad, maar ik heb geen Frans geleerd. Toen kregen we daarna een juffrouw Frans, die was nauwelijks ouder dan de oudste jongen in de klas, misschien nog jonger, die had toch wel enige moeite met orde in de klas. Kortom, Frans is nooit mijn sterkste vak geworden.

Maar Grieks vond ik leuk onder de talen, en dat vind ik nog steeds een heel leuk vak, en van Latijn heb ik ook nog wel wat opgestoken. En voor de exacte vakken hoefde ik niks te doen.

### *En gymnastiek?*

Nee, daar was ik de miskleun. Dat kwam ook, de eerste jaren ging ik vanuit Bergen op Zoom in Goes op school, dat is maar een kilometer of 40, denk ik, maar in die tijd ging er nog een stoomtrein eens in de twee uur. En die deed er een uur over, ja, en dan kon je soms twee uur later vertrekken als je het eerste uur mocht missen. En vooral als het eerste uur gymnastiek was, dan wist ik het voor elkaar te krijgen dat ik daar toestemming voor kreeg. Mijn

gymnastiekopleiding laat dus te wensen over.

*Na je eindexamen, in 1951, ben je gaan studeren aan de Vrije Universiteit.*

Ja, daarna ben ik naar de VU gegaan. Mijn vader zag aankomen dat ik zou gaan studeren, probeerde de zaak economisch in de hand te houden, en had inmiddels gesolliciteerd naar een positie in Alphen aan de Rijn, en in 1951 zijn we daarheen verhuisd.

Toen wij daar woonden is Avifauna opgericht; dat gaf in het dorpsleven nogal wat commotie. Ik ben daar geweest. Ik ben altijd een groot liefhebber van dieren, in het bijzonder van vogels geweest. Dus Avifauna was best de moeite waard. Het was alleen nauwelijks te betalen voor ons. Mijn ouders moesten na de oorlog behoorlijk zuinig zijn, want die zijn volledig berooid, zonder iets, uit Indië teruggekomen en moesten wèl zes kinderen groot brengen.

Ik herinner me verder nog boekhandel Haasbeek, die bestaat nog steeds. Haasbeek trok het land rond en kocht overal winkeldochters op en verkocht die voor een verlaagde prijs. In die boekhandel heb ik heel wat uren rondgehangen en heb daar ook heel wat zakcentjes naartoe gebracht om toch maar weer een boek aan te schaffen.

*Dat heeft de basis gelegd voor je boekencollectie?*

Die basis is al in Indië gelegd, vlak na de oorlog. De Japanners hadden een rare gewoonte. Ze sorteerden alles en sloegen dat dan weer op. Dan had je dus een straat daar was een aantal huizen helemaal volgestouwd met stoelen, in de volgende straat stonden de tafels, die ameublementen waren uit elkaar getrokken. En weer in een volgende straat stonden de bedden, en zo was er een groot huis volgestouwd met boeken. Alle mensen waren geïnterneerd, alle blanken, die huizen stonden leeg, en de Japanners hadden alle huisraad opgeslagen, maar eerst wel even sorteren. Alle boeken waren bijeen gebracht in een verdiepingshuis (zoals we dat noemden in Indië). En die boeken moesten geregistreerd worden, en ik heb als jochie in Bandung zakgeld verdiend door van de boeken de titels op te schrijven, lijsten te maken van die boeken, en ach, ik mocht wel eens een boek meenemen van de toeziende man die daar de leiding gaf. Daar komen mijn eerste boeken vandaan. Nog niet zo veel maar daar begon het mee. Ik had dus al een paar boeken toen ik uit Indië kwam.

*Wat had je grootste interesse? Fictie, non-fiction, ...*

Dat was gemengd. Zowel De Drie Muskietiers als De Wonderen der Wereld, en een boek over kunstgeschiedenis. Van de heer Haasbeek, de boekhandelaar in Alphen aan de Rijn, heb ik bij mijn doctoraal nog een boek gekregen, en dat waren de Analects van Confucius. Ik heb het nog steeds. De vertaling van de wijze woorden van Confucius. Daar was ik ook altijd in geïnteresseerd, religies, gewoonten, met name religieuze gewoonten van andere volkeren. Ik heb



als student de Koran van begin tot eind, alle sura's, doorgelezen. Ik had de Nederlandse vertaling gekocht, met aan de ene kant, zo hoort het bij een goede vertaling, Arabisch waarvan ik alleen de Arabische cijfers heb leren ontcijferen, en aan de andere kant Nederlands, in kolommen naast elkaar.

Ik heb mijn hele leven een zwakke plek gehad en dat zijn maagzweren. En daar heb ik voor het eerst voor gekuurd in 1954. Maar toen betekende dat nog 6 à 7 weken plat en niet bewegen en heel laffe kost. Lezen met behulp van een plankje en zo, je mocht niet overeind komen. De theorie was toen nog dat dat nodig was om zo'n maagzweer te genezen. En in die tijd heb ik de Koran uitgelezen. Ik had hem al, een vertaling van het Almadija Genootschap. De vertaling van Kramers heb ik gekocht zodra die uitkwam. Drie Nederlandse en twee Engelse vertalingen heb ik van de Koran. Ik heb hem toen doorgelezen en hele discussies gehad met de predikant die op ziekenbezoek kwam.

Je hebt nu heel vaak kerkelijke discussies en zo. Dat mensen de vraag stellen van "Waarom is de ene godsdienst meer waar dan de andere?". Nou die vraag kwam toen bij mij ook al op en ik heb die vraag ook aan de predikant gesteld, van "Ja, waarom moet ik de bijbel voor waar houden en de Koran nou niet?". Uiteindelijk is dat natuurlijk geen zinvolle vraag, maar laten we daar nu niet op in gaan. Maar dat soort discussies had ik toen met de predikant.

Ik kan wel stimulansen aanwijzen. In de eerste plaats mijn Indische verleden, waar je natuurlijk opgroeide in een cultuur die Islamitisch is. In de tweede plaats die school in Goes waar ik naar toe ging. Dat was een Christelijke school die erg veel werk maakte van zijn Christelijke karakter. Op een Christelijke school heb je vaak godsdienstles, maar die school in Goes had verschillende religieuze vakken. We kregen een vak Kerkgeschiedenis waar ik echt veel van geleerd heb. We kregen een vak Bijbelkennis en een vak Zendingswetenschappen, en bij dat vak werden ook de grote godsdiensten van de wereld behandeld. Dus op de middelbare school maakte ik al kennis met andere godsdiensten, onder deskundige leiding. Er waren dominees die les gaven, maar dit was een dominee die er echt wel verstand van had. Met de hoofdlijnen van het Hindoeïsme, Boeddhisme, de Islam, het begon met animisme en dynamisme, het was keurig netjes systematisch opgebouwd. Dat boeide me, dat vond ik interessant, en ben ik altijd interessant blijven vinden.

Ik heb geprobeerd in de loop van de tijd naar de bronnen terug te gaan. Ik heb materiaal over het Boeddhisme, Hindoeïsme, het heeft me altijd geïnteresseerd. Het interesseert me nog steeds hoe andere mensen, andere culturen denken, ik probeer mij in te leven hoe mensen uit zo'n cultuur hun bestaan beleven, ik probeer daar enige empathie voor te ontwikkelen, daar hoort de Islam zeer beslist bij.

*Hoe verliep je studie?*

Ik heb tot mijn kandidaats vanuit Alphen aan de Rijn als bus-student gestudeerd aan de VU. Dat betekende dat ik ook geen lid was van een of andere

studentenvereniging, ik was nihilist zoals dat toen heette, maar dat heb ik ook overleefd.

Zoals ik al eerder zei, had ik heel lang biologie willen gaan studeren, maar het is mijn leraar natuurkunde geweest die gemaakt heeft dat het natuurkunde werd. En ik heb daar nooit spijt van gehad, al bleek dat later wiskunde te worden.

Je kon op twee manieren kiezen voor natuurkunde in die tijd: letter A en letter D. Letter A was wiskunde en natuurkunde met sterrenkunde. Letter D was natuurkunde en wiskunde met scheikunde. Ik heb voor A gekozen. Sterrenkunde leek me ook wel leuk en wiskunde vond ik ook leuk. Hoewel scheikunde ook niet bepaald een van mijn slechte vakken was.

De VU was toen nog een kleine universiteit. Als ik het goed heb waren wij in 1951 voor A met zo'n 13 studenten. En dan was er nog een aantal studenten D (waaronder Maarten Maurice), en wat verder weg zat E en zo. Tussen A en D zat niks. Verschillende mensen uit die tijd kom ik nog steeds tegen op de VU. Nel Velthorst, Guus Somsen, dat zijn allemaal studiegenoten.

*Wat herinner je je nog van de hoogleraren natuurkunde?*

Ik herinner me de colleges van Sizoo, dat was een inspirerend docent, boeiende man, was in die tijd bestuurder van TNO. Hij nam geen tentamens af, dat liet hij zijn assistenten doen, maar hij gaf nog wel college.

Ik herinner me ook de colleges van de theoretisch natuurkundige Jonker, maar die heb ik pas leren kennen na het kandidaatsexamen. Ik herinner me zeer wel een van de jongere, zeg maar, adjudanten, van Sizoo, die ook tentamens voor hem afnam en dat was Jan Blok. Die is ook al weer jaren geleden overleden. Het enige tentamen waarvoor ik ooit gezakt ben was bij Jan Blok, atoomtheorie. Toen mocht ik nog geen Jan zeggen uiteraard, de afstand tussen docenten en studenten was groter dan nu.

In mijn tijd was Andriessen de portier van het VU-gebouw in De Lairessestraat, een echte Amsterdammer, een heel zware man, en die bewaakte de lift, want die was alleen voor hoogleraren en wij als student probeerden toch wel eens via de lift te gaan. Maar als Andriessen je in de kraag kon grijpen dan stuurde die je de trap op hoor!

*Wie gaf sterrenkunde?*

Grosheide. Dat deed hij heel consciëntieus; ik vind dat hij dat goed deed. Ik heb zijn sterrenkunde altijd begrepen. Zijn meetkunde heb ik vaak pas achteraf begrepen. Pas later door zelf boeken over lineaire algebra te gaan bestuderen ontdekte ik dat ik dat bij Grosheide ook al geleerd had. Ik kon het wel reproduceren maar ik kon het kennelijk niet in een verband plaatsen. Grosheide was heel erg formeel. Hij gebruikte de kern-index-methode, systematisch, kennelijk kun je zo'n methode leren correct te hanteren zonder dat je weet wat je doet. En dat hebben meer mensen gemerkt aan colleges van Grosheide. Maar Grosheide

gaf boeiend sterrenkunde, dat wil zeggen, ik werd er door geboeid.

*Wanneer heb je besloten toch wiskunde te gaan doen en hoe ben je daartoe gekomen?*

Dat is in de loop van de voorkandidaatsstudie gebeurd. Toen ik eenmaal kandidaats gedaan had was het duidelijk dat ik verder zou gaan met wiskunde als hoofdvak, in het bijzonder door de colleges van Koksma en Mullender.

Ik koos wel natuurkunde als bijvak, in eerste instantie, maar ik heb toen een aanvaring gehad met de practicumleider, die vond ik niet zo erg plezierig. Ik had denk ik ook de pech dat ik mijn kandidaats deed op een moment waarop geen anderen een bijvak natuurkunde begonnen, en ik kreeg dus een opdracht om in mijn eentje een apparaat in elkaar te zetten. Dat ging van: hier is wat materiaal, hier is een soldeerbout, en daar is het magazijn, en maak maar een univibrator. Nou ja, de natuurkunde-hoofdassistent, de heer Hamers, verwachtte dat ik daar dag en nacht aan zou werken want ik legde daarbij beslag op kostbare apparatuur: oscillografen e.d., en ik vond 2 à 3 middagen in de week wel voldoende, want ik wilde ook mijn wiskunde bijhouden. En dat heeft op een gegeven moment tot een botsing geleid; toen is mijn apparatuur weggehaald onder het mom van: Je bent er toch nooit! En ik ben nou eenmaal zo, ik accepteer een hele tijd verschillen van mening, maar dan word ik dwars, dus toen ben ik naar de heer Hamers toegegaan en heb gezegd: Ik zie van mijn studie natuurkunde af.

*Dus je keuze voor wiskunde is terug te voeren op die meccanodoos die je niet gekregen hebt?*

Ja vast. Een oude frustratie, die blijft doorwerken. Toen ben ik als een van de eersten toegelaten tot het uitgebreid wiskunde. Formeel was er die mogelijkheid, dus hoofdvak wiskunde met slechts één bijvak. Maar Koksma en Grosheide en Mullender waren daar niet voor. Dat gaf maar eenzijdige studenten, je moest bijvakken doen, liefst natuurkunde want daar komen tenslotte alle differentiaalvergelijkingen vandaan, waar je als wiskundige zoveel plezier aan beleeft. Maar ik heb voor elkaar gekregen dat ik uitgebreid wiskunde mocht doen, door te beloven dat ik nog meer zou doen dan een dubbele portie, met meteorologie als bijvak. Mechanica zat er ook bij, zal wel onderdeel van het verplichte wiskundepakket geweest zijn.

*Welke herinneringen heb je aan de wiskundecolleges?*

Koksma was een geweldig enthousiast en inspirerend docent, had vanuit zijn enthousiasme een heel leuke manier om met je om te gaan. Tentamens deed je toen nog bij hoogleraren thuis en tentamen bij Koksma was een ervaring, daar ging je met vrees en beven naar toe. Maar als ik daar op terugkijk, die man was ook dan vormend met je bezig, je leerde daar op het tentamen. Grosheide was heel systematisch en verlangde van je dat je het precies zo terug kon vertellen



als hij het je verteld had, tenminste zo beleefden wij dat.

Aan de colleges van Mullender denk ik met plezier terug. Hij heeft altijd iets speels gehad, je zou het ook iets slordigs kunnen noemen, maar speels en een beetje slordig horen vermoedelijk bij elkaar. Die kon heel geniaal met zijn vak omgaan maar liep weleens vast in zijn epsilons en delta's.

De VU was in die tijd klein, en verschillende colleges werden gecombineerd, maar voor diegenen die A gekozen hadden werden nog eens op een apart college, ik meen op de woensdagmiddag, de puntjes op de 1 gezet. Dus op het brede college werd wel eens een aantal dingen geponeerd, en dan kreeg je op het aparte college bij voorbeeld de sneden van Dedekind.

Ik wilde aan het eind van het eerste jaar al meteen tentamen doen, ik was kennelijk nogal ijverig en was goed bij. Ik heb toen meegemaakt dat ik tijdens het college analyse, zeg maar voor het brede publiek, Mullender kon helpen die was vastgelopen in een bewijs. Dat bewijs had hij ook al gedaan op het aparte college en dat had ik al geleerd, dus ik kon hem vertellen hoe het verder moest. Ik geloof nog steeds dat dat mij geweldig geholpen heeft. Want toen ik eenmaal bij Mullender tentamen kwam doen, toen kwam ik daar vrij vlot doorheen. Ik zie me nog zitten met bibberende handen, met zo'n kopje thee dat mevrouw Mullender binnenbracht, vol prachtige concentrische kringetjes: toen was ik geloof ik al geslaagd ...

Koksma had de zeer aantrekkelijke gewoonte om colleges te geven voor alle jaren — voorkandidaats, nakandidaats, tot vijfde-jaars toe. College verzamelingenleer, college groepentheorie, en dat waren heel leuke colleges, dat deed hij met heel veel flair, met veel improvisatie ook, maar dat was boeiend.

In die tijd was het onderwijs aan de VU nog tamelijk conservatief. Het college groepentheorie kwam niet verder dan het allereerste beginstukje, een bladzijde of zestig in Van der Waerden Deel I, en misschien nog wel minder. En een van de argumenten waarmee ik destijds voor elkaar kreeg dat ik uitgebreid wiskunde doctoraal mocht doen was dat ik Koksma aanbood om geheel Van der Waerden te doen als tentamen. Ik denk dat ik toen meer in Van der Waerden gelezen had dan waar hij ooit aan toe gekomen was. Ik heb daar veel van geleerd. Ik vond het een schitterend boek.

*Werd er topologie gegeven in die tijd?*

Ouderejaars vertelden met een zekere nostalgie dat Grosheide eens een caput — en ik verstond hen maar niet, 'tautologie' verstond ik — gegeven had, maar ik begrijp dat hij een keer een college topologie heeft gegeven, ik neem aan voornamelijk algebraïsche topologie. Maar dat was dus een legende, dat was ééns een keer gebeurd. Ikzelf was al aan het eind van mijn studie toen er bij de nieuwe boeken in de bibliotheek een boek lag van Kelley, *General Topology*. Ik heb daarin zitten kijken en dat was volkomen abracadabra voor me, maar het leek me wel heel erg interessant. Ik heb dat boek toen gekocht en gelezen, dat was mijn eerste kennismaking met topologie, maar toen was ik al bijna

afgestudeerd.

*Waren er nog meer vakken die jou in het bijzonder aantrokken?*

Ik vond eigenlijk het hele wiskundeprogramma leuk. Als gymnasiast had ik geen beschrijvende meetkunde gehad, en dat moest je dus inhalen, net zoals de HBS-ers wat analytische meetkunde moesten inhalen. Mullender gaf opdrachten voor beschrijvende meetkunde, en ik herinner me nog dat ik als tentamenopdracht een regelmatig twaalfvlak moest tekenen in drie verschillende projecties, dus volledig geconstrueerd. Dan moet je om te beginnen de regelmatige vijfhoek construeren en daarna verder met centrale projectie en orthogonale parallelprojectie en zo. Dat moest keurig op een groot papier in inkt, dat werd dan ingeleverd en daarna besproken. Het leverde uiteindelijk een handtekening op. Ik had nooit leren tekenen, ik heb toen voor het eerst een trekpen gehanteerd. Ik heb boeken van Van Veen en zo gelezen, over beschrijvende meetkunde, en ik vond dat leuk en boeiend en heb dat met plezier gedaan.

Eigenlijk vond ik alles leuk. Maar wat mij bijzonder aantrok was algebra, verzamelingenleer, en na het kandidaats ook weer de capita van Koksma, bijna-periodieke functies, heel boeiend. Tegenwoordig heb je het over de Bohrcompactificatie en dan trek je het in de harmonische analyse. Maar dat gebeurde toen nog echt op reële getallen zonder generalisatie naar topologische groepen. Dat vond ik een heel boeiend college. Koksma gaf een aantal colleges waarin iedere keer eenzelfde structuur aan de orde kwam, een Banach-ruimte of Banach-algebra die volledig was, en dat heb je bij de bijna-periodieke functies zo. Waar had je dat nog meer bij? Bij Fourier-transformaties en Fourier-reeksen, en dat behandelde hij dan systematisch op dezelfde manier. Je had een eenduidigheidsstelling en een volledigheidstelling en de ongelijkheid van Parcefal en dat kwam dus in verschillende contexten terug. Heel weinig efficiënt maar wel heel leerzaam. Je gaat dan inderdaad zien dat er een gemeenschappelijke structuur, een abstracte structuur, ligt achter allerlei concrete wiskunde. En dat heb ik boeiend gevonden.

*Statistiek, daar heb je het nog niet over gehad.*

Nee, dat werd nauwelijks gegeven. Van Rooijen was buitengewoon hoogleeraar en gaf verzekeringswiskunde. Dat heb ik bij hem gelopen. Hij gaf ook van tijd tot tijd colleges statistiek en numerieke wiskunde, maar die heb ik nooit gelopen. Ik heb dus helemaal geen statistiek gehad. En met zijn numerieke wiskunde ben ik ook nooit geconfronteerd. Wel heb ik nog eens een college demografie van hem gelopen. Hij werkte bij een verzekeringsmaatschappij, de "Hollandsche Societeit", dus demografie was zijn specialiteit en zijn kernbelangstelling. Maar statistiek en waarschijnlijkheidsrekening en ook numerieke wiskunde zijn helemaal aan mij voorbij gegaan.

*Ook omdat het je minder interesseerde?*

Van Rooijen gaf niet zoveel colleges en het was het ene jaar dit en het andere jaar dat en ik ben niet zo erg lang als student blijven hangen. Ik heb in 1954 mijn kandidaats gedaan en in 1957 mijn doctoraal. Ik had in 1952 een meisje ontmoet waarmee ik zou trouwen, ben in 1954 verloofd en heb de hele zomer hard gewerkt in een jamfabriek om voldoende te verdienen om verlovingsringen te kopen. En ik ben in 1956 voor de klas gegaan, leraar geworden, om een basis te leggen voor de bruiloft.

En ja, op een gegeven ogenblik zette ik er nogal de vaart in, ik was van ons jaar de eerste die afstudeerde. Dat betekende dat ik vermoedelijk nauwelijks de gelegenheid gehad heb nog veel extra colleges bij Van Rooijen te volgen.

*In welke jamfabriek werkte je?*

Pfeiffer of De Pijper of zo, hij bestaat al lang niet meer. De fabriek stond aan de Leidse Rijn, en die jamketels ... Er werd iemand op de uitkijk gezet en die ketels werden dan met dat vieze Rijnwater omgespoeld, voordat de volgende portie erin gekookt werd. En bovendien, als je zag hoe die frambozenmandjes van de veiling kwamen, bedekt met een laag schimmel en rupsen en zo, en die gingen gewoon met levende have het vuur op, er ging vervolgens toch een heleboel sulfiet bij. Het was zeer leerzaam, ik heb jarenlang geen jam willen eten.

*Daarna ben je ook uit Alphen aan de Rijn vertrokken?*

Na mijn kandidaatsexamen in 1954 ben ik op kamers gaan wonen. Ik zat op een zolderkamertje op de Tweede Kostverlorenkade, vlakbij de Wiegbrug. Heel plezierig om je eigen baas te zijn.

*Wie herinner je je nog van je medestudenten?*

Ik noemde Maarten Maurice al (later hoogleraar wiskunde aan de VU), Wim Kuyk (later hoogleraar o.a. in Antwerpen), zijn vrouw Minke Zuidema, die kwam een jaar later, maar daar ben ik nog samen mee naar school gegaan in Solo, samen in een andong, in zo'n paardewagentje daar. Een jaar eerder Piet Born, heeft natuurkunde gestudeerd, gepromoveerd en is naar Pakistan gegaan en heeft daar een Christelijk College opgebouwd. Daar is hij nu net bezig afscheid te nemen.

Later uit de kandidaats-fase: Nico Habermann, die gepromoveerd is bij Dijkstra in Eindhoven en later naar de VS gegaan, Hoofd Computer Science bij Carnegie Mellon geworden. Ik heb hem daar nog opgezocht, de laatste keer dat ik daar zijn gast was. Toen ben ik ook te gast geweest bij Dana Scott thuis, met mevrouw Habermann en Nico. Laatstelijk was Nico de Computer Science Advisor van de National Science Foundation. Maar hij is een jaar geleden, veel te vroeg, heel onverwacht, overleden.

Wim Blokluis moet ik zeker noemen, die is later naar Den Helder gegaan,

naar de Opleidingsschool voor de Marine. Hij was een oudere studiegenoot die ons zo nu en dan tot kalmte maande als we weer eens wat te enthousiast met de bordenwissers gingen gooien, die wist al dat dat niet hoorde. Eyt Algra, moet ik zeker ook noemen, die heb ik al heel lang niet gezien, ook aan hem heb ik heel plezierige herinneringen. Een goede vriend die ik nooit meer gezien heb sinds we afgestudeerd zijn.

En iemand die ik dan ook nog even moet noemen dat is Han Schippers, de zoon van Professor Schippers, de theoloog. Die jongen had tbc, en in de periode nadat hij uit het sanatorium was en genezen verklaard heb ik hem leren kennen. Wij hebben toen met een aantal studenten — Maarten Maurice en Wim Kuyk waren daar ook bij, maar Han Schippers prominent ook — een werkgroep formele logica opgezet, en zijn daar een aantal boeken gaan bestuderen, en nodigden daar de hoogleraren uit, en die waren zo sportief dat ze ook kwamen. Koksma, Grosheide en Mullender. We bestudeerden dus een boek van Carnap en een boek van Curry. We hadden wel Koksma om advies gevraagd, en die had Beth om advies gevraagd en had dat overgebracht. En daar heb ik voor het eerst kennis gemaakt met de formele logica. Dat vond ik heel erg boeiend.

Maar Han Schippers is heel jong overleden. Waar ik mij hem ook altijd om blijf herinneren is omdat hij een heel sympathiek iemand was. Ik heb hem niet lang gekend, hij deelde met ons de belangstelling voor dat soort abstracte zaken als logica maar hij heeft mij ook geïntroduceerd in andere zaken, *Gargantua et Pantagruel* van Rabelais heb ik van hem geleend in de vertaling van Ernst van Altena, daar was ik heel erg door geboeid in die tijd. Ik heb het later zelf aangeschaft.

Dat zijn zo een paar namen die bij me opkomen.

*In 1957 ben je afgestudeerd. Wat heb je toen gedaan?*

Ik wilde graag promoveren bij Koksma, hoewel ik meer tentamens had gedaan bij Mullender dan bij Koksma. Maar ik had gesolliciteerd naar een baan als leraar in Haarlem, en ik had Koksma gevraagd of hij referenties wilde geven. Toen reageerde hij nogal terughoudend, misschien moet ik zeggen nogal fel. Hij vond het niks voor mij en bood me bij die gelegenheid een assistentplaats op het Mathematisch Centrum aan, waarvan Koksma toen Directeur was. Ik ben toen naar mijn vader gegaan, van "Wat moet ik nou doen?". Die baan in Haarlem werd me aangeboden en ik kon assistent worden op het MC. En mijn vader zei: "Je moet natuurlijk een *echte* baan nemen". Hij was in de crisistijd begonnen en een school was tenminste een serieuze, betrouwbare werkgever.

Maar Koksma heeft me van de school geplukt, en gezorgd dat ik een beurs naar de VS kreeg. Er was toen nog een programma van het State Department, voor uitwisseling met Europa, de International Cooperation Administration, in het kader van de wederopbouw van Europa.

Ik heb twee jaar in Berkeley gezeten. Ik heb toen college gelopen bij Van der Corput, die zat toen in Berkeley, maar vooral bij Tarski en Henkin. Ik heb

erg veel steun van Henkin gehad. Hij was heel sociaal gericht en ving mensen die van buiten kwamen op, deed daar wat voor. Kelley was met sabbatical, die was er niet, maar Loève, de waarschijnlijkheidstheoreticus, gaf college topologie uit het boek van Kelley.

Ik heb in Berkeley de moderne wiskunde geleerd. Ik heb daar heel veel colleges gelopen. Ik kwam daar eigenlijk met een fellowship met de bedoeling dat ik daar onderzoek zou leren doen of mee zou doen met het onderzoek. Maar ik heb daar heel veel colleges gelopen, en daar de moderne benadering van de wiskunde leren kennen die ik op de VU niet tegengekomen was, algebra, moduletheorie, topologie, veel logica. Mostowski kwam daar in sabbatical, gaf axiomatische verzamelingenleer, heel boeiend, heel aimabele, knappe man ook. Daar heb ik Dana Scott voor het eerst gezien, ik ken hem langer dan hij mij kent. Hij hield een verhaal op het seminar van Tarski. In die tijd was er ook een keer een congres waar ik als toehoorder bij zat, op het gebied van logica en grondslagen. Montague heb ik daar ook ontmoet. Dat is voor mij een heel belangrijke ervaring geweest.

*Na je tijd in Berkeley ben je op 1 oktober 1959 aan het MC begonnen. Hoe verliep je promotieonderzoek? Hoe werd dit beïnvloed door je tijd in Berkeley?*

Toen ik terugkwam uit Californië had ik een heel wat andersoortige wiskunde in mijn bagage dan toen ik er naar toe ging. Wat ik in Berkeley geleerd heb is de moderne opzet van de wiskunde, de meer abstracte opzet, het gebruik van het Lemma van Zorn, de meer algebraïsche, meer structurele benadering van zaken, in plaats van alles precies doorrekenen. Koksma gaf toch veel meer colleges complexe functietheorie, differentiaalvergelijkingen, op de ouderwetse manier, waarbij je de zaak helemaal doorrekent, en begint met éénachttiende epsilon en aan het eind komt het precies allemaal goed, en daar was hij geniaal in.

Mijn bedoeling was om bij Koksma te gaan promoveren. Maar Koksma is ik meen eind 1960 ernstig ziek geworden en een tijd lang uitgeschakeld geweest. Koksma is toen ook opgevolgd als Directeur van het MC door Van Wijngaarden.

Koksma kon mij toen niet coachen, en in die periode ben ik in het gravitatieveld van De Groot terecht gekomen en ben ik aan problemen gaan werken die De Groot mij voorhield. Ik was erg geboeid door laten we zeggen de Bourbaki-wiskunde, die ik uit boeken en geschriften in die tijd leerde, en De Groot als topoloog deed natuurlijk onderzoek in die richting.

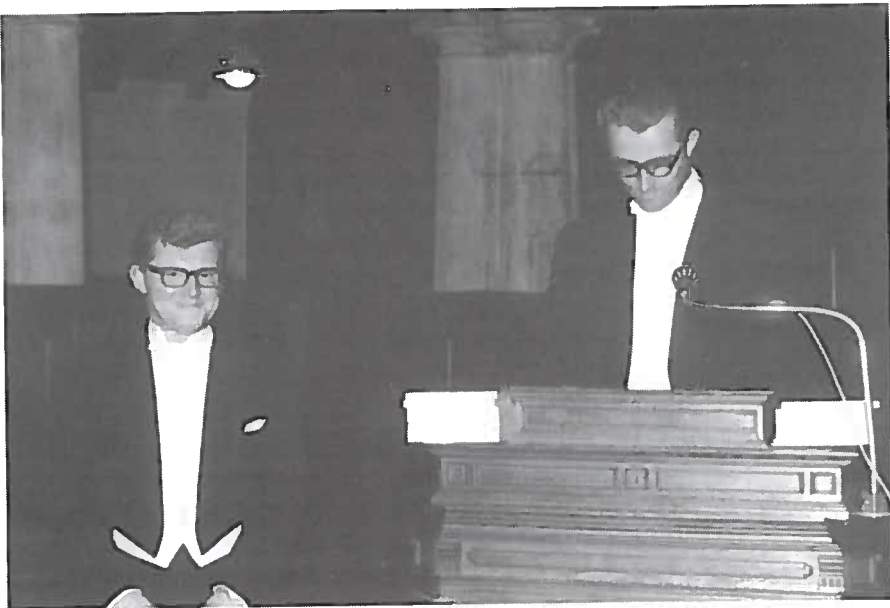
*Dus als Koksma je meer had begeleid was je misschien in een meer klassieke richting gegaan dan waarin je nu bent gegaan.*

Ja, maar Koksma had er zeker oog voor dat het inmiddels op een andere manier moest en kon. Dat blijkt al uit de opdrachten: probeer die ideeën van Weyl over gelijkverdeling nou eens te generaliseren naar andere lokaal compacte groepen dan de reële getallen. Dat is natuurlijk al een moderne benadering. Dus Koksma wilde me wel degelijk in een andere richting.



Ik heb eerst geprobeerd op mijn eentje op het gebied van gelijkverdeling wat te doen en op Koksma's suggestie de ideeën van Weyl te generaliseren naar topologische groepen. Dat is me toen niet gelukt. Later heb ik samen met Gilbert Helmberg op het gebied van gelijkverdeling wat gedaan. Dat was nadat Koksma weer voldoende hersteld was. Er is toen een colloquium gelijkverdeling geweest en Helmberg was een van de sprekers. Hij was toen een jaar gastmedewerker op het MC en ik heb later samen met Gilbert een artikel geschreven, waarbij ik graag ruitelijk toegeef dat Gilbert daar heel wat meer expertise en kennis in ingebracht heeft dan ik. Hij had ook een kleine voorsprong op mij ...

Maar later heb ik ook een aantal dingen, waar we samen niet uitkwamen, opgelost, in de zomer van 1964, bij een bezoek aan Zdenek Hedrlín in Praag. Daarvoor, in diezelfde zomer ben ik ook gepromoveerd.



... in diezelfde zomer ben ik ook gepromoveerd ...

### *Hoe was de sfeer op het Centrum?*

Leuk, heel plezierig. Ik heb met Van Herk op een kamer gezeten en dat was leerzaam. Heel interessante man, een keel-, neus- en oorarts die altijd wiskunde had willen studeren, maar zijn ouders vonden dat je als dokter een betere boterham kon verdienen. Hij was bijna bezeten van de Riemann-hypothese. In die richting was hij iedere keer aan het graven. Hij was hoogleraar in Bandung

geweest. Aanvankelijk was hij op het Mathematisch Centrum aangesteld op een plek waar op dat moment plaats was en dat was op de Afdeling Statistiek, en daar werd hij aan het werk gezet maar dat had niet zijn hart. Zijn hart was de Riemann-hypothese, de getaltheorie, de priemgetalverdeling. Toen is men lankmoedig geweest en heeft men gezegd, goed, de helft van je tijd mag je aan je dierbare wiskunde besteden, en dat betekende dan ook de helft van de tijd gestationeerd bij ZW. En de andere helft van zijn tijd moest hij dan toch het werk van Statistiek blijven doen, consultaties e.d., want als medicus was hij daarvoor natuurlijk bij uitstek geschikt. Veel van de consultaties bij Statistiek kwam van medici en andere onderzoekers die proeven opzetten en experimenten wilden doen en die statistisch wilden interpreteren. En die moesten dan geholpen worden om niet van tevoren hun uitkomsten al te projecteren in de wijze waarop ze hun experiment deden; Van Herk sprak hun taal.

Van Herk had ook een eigen 'theory of everything' waarbij hij, geloof ik, het aantal dimensies van de fysische realiteit kon uitrekenen, en dat was weer een heel ander aantal dan wat je bij andere auteurs aantreft. Later is Van Herk hoogleraar in St. Andrews in Schotland geworden.

In de lunchpauze werd er gebridget. Ik werd gewoon geronseld, ik kon helemaal niet bridgen, nou, dat heeft Lekkerkerker me dan maar geleerd. Hij was een goed bridger, hij heeft zich vrees ik nogal eens aan mij moeten ergeren want ik ben nooit een goed bridger geworden. Maar die traditie die toen begonnen is hebben we jarenlang voortgezet. Gert-Jan Förch is toen een tijdlang medewerker geweest bij TW, bekend auteur van bridge-handboeken, zeer deskundig. Als partner bleef hij altijd heel keurig en hoffelijk. Al was het nog zoveel troep wat ik daar voor hem neerlegde, ik werd heel hoofs bedankt, 'thank you, partner'; ja dat was een levensles!

We hadden een heel leuke traditie, we gingen heel amicaal met elkaar om, met name de mensen van TW en ZW. De Rekenafdeling was een groep apart, hoewel Dirk Dekker vaak bij ons kwam meebridgen, maar die was natuurlijk in feite een oud-ZW-er, bij De Groot gepromoveerd. En Statistiek was ook een groep apart, daar hadden we niet zo veel contact mee.

### *En de Algemene Dienst?*

De Algemene Dienst, ja, Mevrouw Oosting, hoofd van de huishouding. En de heer Van Ommen, die de koffie rondbracht. Die had een heel gore stofjas die vermoedelijk in geen jaren gewassen was, waarmee hij achter de stencilmachine stond en die onder de inkt zat en zo. In die stofjas zaten dan de lepeltjes, en dan kwam hij binnen, geklopt werd er nooit, de deur werd opengestoten en daar zat ik dan met mijn benen op mijn bureau, die trok je dan gauw naar beneden (of niet, als je zag dat het Van Ommen was). Dan werd er koffie neergezet en dan graaide hij in die grauwe zak en dan kwam er een lepeltje uit, of een klontje dat had hij ook los in zijn zak. Dat hoorde er allemaal bij en dat was wel leuk.

Bep Reckman was natuurlijk een zeer bekende figuur, ook een zeer invloedrij-



ke figuur, en weleens een wat onvoorspelbare figuur. Ze kon wel eens emotioneel zijn. Had ze ook wel aanleiding toe.

Ook als je goed met haar overweg kon dan kon het je gebeuren dat je op het verkeerde moment bij haar kwam en dan werd je dat ook wel duidelijk gemaakt. Ze nam nooit een blad voor de mond. Dus als ze vond dat je je niet gedroeg zoals zij passend vond dan liet ze je dat wel weten.

Mevrouw Monasch herinner ik mij op de bibliotheek, op de tijdschriftenafdeling. Ik zat altijd veel op de bibliotheek, ik had toen nog de tijd om de nieuwe boeken door te kijken en tijdschriften door te bladeren, en te studeren. Ik kende dus de bibliotheecarissen allemaal. Mevrouw Monasch zat op de tijdschriftenafdeling, ze was politiek uiterst links en was dus in het bijzonder gericht op alle ruilingen met Oost-Europa. En daar heeft de bibliotheek heel veel aan gehad want allerlei Russische, maar ook Roemeense en Georgische en Azerbeidjaanse, en misschien wel Tsjetsjeense tijdschriften, hadden wij hier en die waren nergens anders in West-Europa. Wij hadden een complete collectie. Mevrouw Monasch ging bij voorkeur achter het IJzeren Gordijn met vakantie en zocht academies op en vertelde ze hoe stom ze waren dat ze niet met ons ruilden en bracht dan weer ruilingen tot stand.

Meestal lag ze overhoop met wie op dat moment bibliothecaris was, wie dat ook was. Maar ik kon het goed met haar vinden op de een of andere manier. Ze maakte nogal eens een praatje met me en dat vond ze kennelijk wel leuk. Ik mocht haar wel, het was een pittig vrouwtje.

*Hoe heb je Koksma op het MC leren kennen?*

Koksma zette me meteen aan het werk, hij gooide me meteen in het diepe. Die kreeg als secretaris van de Akademie voor de Indagationes artikelen aangeboden en die gaf hij dan aan mij en daar moest ik hem dan een rapport over geven en dat soort zaken. Dat ging soms over dingen waar ik nog niets van wist. Maar de Afdeling Zuivere Wiskunde bestond toen uit twee personen, de andere was Gerrit Lekkerkerker, en die heeft mij geweldig gecoached, daar heb ik erg veel van geleerd. Dus als Koksma mij dan weer een hap gaf die eigenlijk mijn kunde te boven ging dan zei Lekkerkerker heel wijs en relativerend: "Dat doe je zo en zo", op een beetje socratische manier.

Koksma schakelde me onmiddellijk ook in bij colloquia, het was toen nog belangrijk dat er ieder jaar een groot colloquium was waar een van de afdelingen voor verantwoordelijk was en waar in principe iedereen uit het land uitgenodigd was. Koksma heeft colloquia georganiseerd over  $p$ -adische getallen en over gelijkverdeling. Er werden van tevoren syllabi uitgereikt. Dan werd je, als Koksma aan de beurt was als spreker, op zijn kamer geroepen, daar in de Boerhaavestraat, een kamer beneden, vlakbij de ingang. Daar liep hij te ijsberen — dat deed hij altijd als hij diep nadacht, ook op een tentamen, als hij je dan vragen stelde dan liep hij achter je rug heen en weer; dan had je het gevoel van wat komt die nou weer doen, uit welke hoek zal hij me bespringen.

Dan liep hij dus te ijsberen en telefoneerde, het ene telefoontje na het andere, Barning, secretaris van de directie, rende in en uit en gaf hem stukken, Koksma las al telefonierend de stukken, en tussendoor dicteerde hij zinnen voor zijn syllabus. Ik zat dat allemaal ijverig op te schrijven, en werd dan weer naar boven gestuurd om dat uit te gaan werken.

Koksma kon drie dingen tegelijk doen. Daar is hij, denk ik, ook aan doodgegaan. Hij is uit hetzelfde jaar als mijn vader, 1903, en is in december 1964 overleden, dus hij was 61 jaar. Koksma heeft het meest van de oprichters gedaan aan de realisatie van de Stichting en het instituut. Hij is van de stichters ten onrechte het meest verwaarloosd.

Er is veel meer erkenning voor Van Dantzig als de visionaire man, de man van de toepassing van de wiskunde, op alle mogelijke vakgebieden. Van der Corput was de man met gezag, de senior, hij was de promotor van Koksma. Van der Corput kende Van der Leeuw nog als collega in Groningen, en had dus het oor van de minister, want Van der Leeuw was in 1945 Minister van Onderwijs, Kunsten en Wetenschappen.

Maar het werk werd grotendeels door Koksma gedaan. Het Internationale Congres van 1954 is door Koksma georganiseerd (met een stel adjudanten zoals Jaap Seidel), hij dirigeerde alles, maar de voorzitter van het congres was niet Koksma maar Schouten. En in de stukken kom je Koksma nauwelijks tegen. Het is Schouten als voorzitter van het congres. Maar het werk, de hele organisatie was in handen van Koksma.

En zo was het ook onder de oprichters van het MC. Koksma deed geweldig veel werk. De basis voor de bibliotheek is gelegd door Koksma, hij was de inspecteur van de boekerij van het Wiskundig Genootschap, secretaris van de Koninklijke Akademie, had overal zijn netwerk zitten, hij gebruikte dat om dingen voor elkaar te krijgen, had geweldig veel informatie en een geweldige werkkraft.

*Vond Koksma zichzelf ook ondergewaardeerd?*

Nee, hij was een heel bescheiden man, hij werkte heel hard maar vond dat ook zijn vanzelfsprekende plicht. Hij was lid van de KNAW en van de Fryske Akademie, zat in allerlei schoolbesturen, deed kerkeraadswerk, deed heel veel.

*Is hij altijd een voorbeeld voor je geweest?*

Ja, Koksma is een voorbeeld voor me geweest in een heleboel opzichten. Dat komt ook omdat ik de facto zijn opvolger was aan de VU. Dat is een hele verantwoordelijkheid geweest. Het is met name ook Koksma geweest die mij aangetrokken heeft op de VU, met zijn collega's uiteraard, want Koksma deed dat in goed overleg. Eerst voor leeropdrachten (vanaf 1962), en later heeft hij mij voorgedragen voor een buitengewoon hoogleraarschap. En toen hij overleed in december 1964 heb ik zijn colleges overgenomen en gecontinueerd. Zo was hij bijvoorbeeld bezig met een college variatierekening, daar had ik nog nooit iets



... Koksma is een voorbeeld voor me geweest in een heleboel opzichten ...

van gedaan. Maar ik heb toen het dictaat van Jan de Vries geleend en heb de Kerststilstand van de colleges gebruikt om dat door te werken en wat vooruit te werken, en ik heb Koksma's college afgemaakt. Ik vertelde op donderdag alles wat ik wist en dan rende ik naar huis en dan ging ik weer een week, alle ogenblikjes die ik kon vinden, studeren, om dan de volgende week weer drie uur college te kunnen geven.

Ook heb ik de tentamens afgenomen van de mensen die bij hem eerder college hadden gelopen, zoals bijna-periodieke functies en zo. Ik ben dus in zijn voetsporen getreden, ik ben altijd beschouwd als zijn opvolger. En ja, dan heb je toch het gevoel dat je andermans profetenmantel op de schouder hebt gekregen en zo iemand wordt je alleen daardoor al tot voorbeeld.

*Je bent in 1965 niet alleen benoemd tot hoogleraar aan de VU, maar ook tot chef van de Afdeling Zuivere Wiskunde van het MC.*

Ja, najaar 1964 stierf niet alleen Koksma, maar ook werd De Groot door ziekte getroffen. Ik moest dus zowel op de VU het werk van Koksma overnemen, als op het MC de leiding van de afdeling Zuivere Wiskunde. Ik heb toen echt héél hard gewerkt. Dat is een heel zware tijd geweest, maar wel boeiend want het lukte. Ik was natuurlijk ook een stuk jonger, je kunt dan ook meer hebben. Ik heb heel veel kunnen doen. Maar er bleef weinig tijd over voor eigen onderzoek.

Ik heb de facto eind 1964 de leiding van de afdeling ZW gekregen, woonde ook de vergaderingen van de Raad van Beheer bij. Maar alleen het Curatorium kon mij benoemen tot lid van de Raad van Beheer en dit vergaderde pas in 1965. Toen ben ik ook benoemd tot chef en lid van de Raad van Beheer. Dus in 1964 ben ik al begonnen als chef de belangen van de afdeling waar te nemen,

maar in 1965 is dat geformaliseerd.

*Hoe groot was de afdeling toen je chef werd?*

Niet zo erg groot. Maarten Maurice was denk ik vertrokken na zijn promotie, hij had een halve baan op het MC en een halve aan de VU. Wim Kuyk was hier nog. Dirk Kruyswijk was hier inmiddels, Ietje Paalman was hier en ik denk dat Jan van der Lune al assistent was, ik weet niet of Peter van Emde Boas er al was als assistent, ik denk het eigenlijk wel, of misschien al adjunct-medewerker of zo. Jos van de Slot is ook een leerling van De Groot, die zal er ook al geweest zijn. Dan ben je al een heel eind. De meeste anderen zijn later gekomen.

*Waar lag je zwaartepunt in die tijd? Je werkte 4 dagen in de week op de VU en 1 dag in de week op het MC.*

Vanaf 1 juni 1965 ja. Vrijdags was ik hier, maar het meeste werk gebeurde op de VU.

*Maar je hebt het MC altijd heel belangrijk gevonden, verhoudingsgewijs meer dan een vijfde baan?*

Zeker, maar ja, zoals ik al zei, Koksma was overleden, er moest een heleboel werk gedaan worden aan de VU, ik had daar een heleboel werk. Maar vrijdags was ik hier, en dan ook volledig hier, en dan probeerde ik hier de zaak op te bouwen, niet zonder succes want de afdeling ZW is behoorlijk gegroeid in de jaren dat ik daar chef was.

*Je bent in die tijd ook een jaar naar Seattle geweest.*

Ja, dat had De Groot nog voor elkaar gekregen, dat ik kort na mijn promotie een jaar naar Seattle zou gaan. Toen overleed Koksma, en is mij gevraagd om Koksma te vervangen en is het uitgesteld tot 1966. In 1966-1967 heb ik in Seattle gezeten.

Ik was erg geïnteresseerd in het werk van Edwin Hewitt. "Abstract Harmonic Analysis", het tweedelige boek van Hewitt en Ross. Ik had het eerste deel helemaal doorgewerkt, er zaten ook heel wat topologische groepen in, prachtig boek! Het tweede deel is twee keer zo dik, en daar ben ik nooit echt mee klaar gekomen.

Ik heb daar ook weer genoten van het college geven, ik had een heel goed rapport met de studenten. De enige keer in mijn leven dat ik aan het eind van een college een applaus kreeg; dat is me hier in Nederland nooit gebeurd. En ik heb daar het seminar van Hewitt bijgewoond. Ik moet achteraf zeggen dat mijn voorkennis onvoldoende was. Ik had meer van maattheorie en harmonische analyse van maten moeten weten om daar ten volle van te kunnen profiteren.

Maar ik heb daar ook wel weer de kans gehad een aantal seminars en zo te volgen. Branko Grünbaum die daar aan convexiteitstheorie deed, Victor Klee.

Ik heb daar veel opgestoken maar onvoldoende zelf resultaten bereikt vind ik achteraf. Ik was toch in sommige opzichten wellicht iets te ambitieus geweest in wat ik wilde.

Ik heb toen wel de tijd gevonden om met De Groot — dat wil zeggen, ik schreef, maar we publiceerden het onder gemeenschappelijke naam — een overzichtsartikel over mijn proefschrift te schrijven, over een aantal resultaten daaruit waar De Groot ook aan bijgedragen had. Ik heb zo die tijd in Seattle vooral gebruikt om een aantal dingen te consolideren die ik al had.

Maar de bedoeling was natuurlijk dat ik er met nieuwe ideeën vandaan zou komen en dat is onvoldoende gelukt vind ik achteraf.



*... eigenlijk de eeuwige student ...*

*Heb je daar te weinig aandacht aan gegeven?*

Nou ik denk dat mij daar parten speelde wat mij mijn hele leven parten heeft gespeeld, dat ik te veel tegelijk wilde en een te brede belangstelling heb, te weinig focus op één onderwerp. Ik ging naar Grünbaum en ging naar Klee en ging naar Hewitt en ik zat weer college te lopen bij wijze van spreken. O ja, ook Namioka gaf een seminar over  $K$ -theorie en ik ging naar een seminar over Lie-algebra's. Ik vond het allemaal even interessant, maar ik had minder mijn

aandacht breed moeten uitspreiden en mij meer moeten focussen op een of twee onderwerpen.

*Waarom?*

Dan had ik zelf meer research kunnen doen. Dan had ik misschien een wat minder brede basis gehad voor later, bijvoorbeeld voor mijn functie op het MC, maar dan was de kans groter geweest dat ik zelf wat originelere bijdragen had geleverd. Ik vind dat ik daar een kans gemist heb.

Maar ik heb er wel weer een heleboel geleerd hoor! Mijn inzicht in wiskunde is daar weer gegroeid.

*Je bent dus eigenlijk de eeuwige student.*

Ja, een beetje wel.

*Vind je dat je te weinig gebruik hebt gemaakt van je capaciteiten?*

Ik heb eens de illusie gehad dat ik meer eigen, origineler, creatief werk kon doen en er is een aantal redenen waarom dat niet van de grond is gekomen. Een daarvan is een brede belangstelling, waardoor ik mijn aandacht vrij breed verdeelde. Een andere reden komt voort uit de problemen in mijn persoonlijke leven, die al heel lang, 25 jaar, veel energie en veel tijd van mij gevraagd hebben. Waardoor ik mij ook niet zo kon concentreren als wenselijk ware geweest. Achteraf gezien zijn die problemen er mede de oorzaak van geweest dat ik eind 1969 een tijdlang uitgeschakeld geweest ben, afgeknapt, een aantal maanden mijn werk niet heb kunnen doen, en daarna heeft het een tijd geduurd voordat ik een deel van mijn oude energie terug had. Eigenlijk heb ik altijd het gevoel gehad dat ik die nooit volledig terug gehad heb. Ik heb na 1970, vaak, zo gekscherend maar niet helemaal badinerend, gezegd: "Ik heb weer net gedaan alsof ik nuttig gewerkt heb".

*De ziekte van 1969-1970 ging ook terug naar je tijd in het kamp?*

Het was een combinatie van zaken. Het ziekteproces bij mijn vrouw begon in die tijd duidelijk manifest te worden. Ik had het geweldig druk gehad. Ik was op de faculteit voorzitter van de commissie Computers en daar waren spanningen, het was de tijd vlak voor de oprichting van SARA. De natuurkundigen hadden een computer en de wiskundigen vonden dat zij dan tenminste ook een computer mochten. Er moesten leerplannen ontwikkeld worden voor onderwijs in de informatica. Ik ben toen ook voor korte tijd voorzitter van de wiskundegroep geweest. Ik had een zware belasting zowel op het MC als op de VU. Problemen thuis, de gezondheid van mijn vrouw, dat totaal is mij teveel geworden. December 1969 kreeg ik daar een zware griep bij en op de een of andere manier is er toen iets geknapt en in die periode zijn allerlei herinneringen uit Indonesië en het kamp teruggekomen. Maar de directie aanleiding was gewoon ordinaire



overbelasting door combinatie van mijn twee banen en mijn huiselijke situatie. Dat is alweer 25 jaar geleden.

Ik ben binnen een half jaar weer aan het werk gegaan maar ik kon eigenlijk nog niks doen. Ik ben toen echt goed van de kaart geweest.

*Je hebt aan de VU gelijk gewerkt met Maarten Maurice, die ook topoloog is. Heb je dat als een probleem gezien?*

Nee, ik heb trouwens veel meer colleges gegeven dan topologie. Cassels zou naar de VU komen en toen heb ik een jaar algebraïsche getaltheorie gegeven. Ik heb recursieve functies gegeven, logica, topologische groepen, harmonische analyse, het boek "Integrals and Operators" van Segal en Kuntze heb ik een keer op college behandeld. Ik heb vooral nakandidaats heel veel onderwerpen gegeven ook buiten de topologie. Ook de eerste informaticacolleges op de VU heb ik gegeven. Ik heb automatentheorie gedaan.

Dat zat ook een klein beetje in mijn leeropdracht, die was veel breder dan topologie, dat was gewoon de zuivere wiskunde, en dat heb ik geïnterpreteerd als het introduceren van de moderne wiskunde. Naarmate er specialisten kwamen die het konden overnemen schoof ik weer door.

*En dat is je goed bevallen?*

Ik heb er zelf heel veel van geleerd. De topologie liet ik voor een belangrijk deel aan Maarten over. Maar we hebben ook dingen samen gedaan, we hebben een keer samen college categorietheorie gegeven. We verdeelden ook wel de taak. Ik deed topologische groepen, dat liet Maarten liggen. Maarten deed meer aan homotopietheorie en zo, en dat liet ik dan weer liggen. Dat was geen enkel probleem.

*College geven heb je ook altijd graag gedaan.*

Dat heb ik altijd heel graag gedaan, ja. Dat mis ik het meeste. Toen ik mijn huidige functie accepteerde was ik net mijn onderzoek weer aan het optuigen. Dat heb ik niet kunnen volhouden, dat is blijven liggen, maar ik heb nog jaren college gegeven en toen ik dat ook moest opgeven vond ik dat echt sneu. En dat had een dubbele reden.

Enerzijds de toeneinende werkdruk hier op het CWI en anderzijds, het laatste college dat ik gaf was logica voor informatica-studenten en ik vond het jammer om dat op te geven, dat mis ik nog steeds.

Ik heb altijd met het grootste genoegen ook voorkandidaats-colleges gegeven, en speciale colleges zoals Boole'se algebra's, daar haalde je de goede studenten mee uit. Jan van Mill die op tentamen met veel onconventionele bewijzen kwam omdat hij bij lineaire algebra, groepentheorie, of zo, dingen had geleerd die hij probeerde toe te passen. Het werkte niet, maar het feit dat hij het probeerde was al zo ongewoon dat je meteen dacht, dat is iemand die denkt voor zichzelf.



Het was veel werk maar ik heb de tentamens altijd mondeling afgenomen dus ik kreeg ze allemaal langs, ook diegenen die daar niets van brouwden. Maar dat persoonlijk contact met de studenten, mis ik ook. Ik persoonlijk vond de mondelinge tentamens leuk.

*Hoe was de sfeer op de VU toen je daar als hoogleraar zat?*

Ik heb de sfeer daar altijd fijn gevonden, collegiaal, plezierig, je kon elkaar aanspreken op toch een grotendeels gedeelde levensbeschouwing. Het was een groep mensen die een flink stuk achtergrond deelden, in leeftijd dicht bij elkaar zaten, elkaar als student gekend hadden. Maar er is natuurlijk ook een gemeenschappelijk uitgangspunt.

Koksma, Mullender en Grosheide vond ik echt heel nobele en heel humane mensen die heel wijs en heel coöperatief bezig waren. Ik heb grote bewondering voor deze mensen. Ik kwam vorige week op de dies Mullender nog tegen en dan realiseer je je hoeveel warme gevoelens je voor die mensen hebt. Er waren eigenlijk nooit spanningen.

Ik was overigens van geen enkele taak uitgesloten. Ik had een volle onderwijslast, bestuurlijke taken e.d. Andere hoogleraren hadden een dag om thuis te werken en ik had een dag op het MC.

*Maar je wilde waarschijnlijk ook niet minder? Of was het zo dat je aandrang op vermindering van je taken?*

Nee. Ik heb altijd een misschien wat doorgeschoten plichtsgevoel gehad.

*Woekeren met je talenten?*

Ja. En misschien toch ook een soort schuldgevoel. Ik zei het al, ik heb het gevoel dat ik in mijn leven meer eigen creatieve bijdragen had moeten leveren en als je daar op de een of andere manier niet aan toe komt, voel je je verplicht dat te compenseren. Dat heeft bij mij beslist ook wel een rol gespeeld.

*Het is natuurlijk ook een keuze van je geweest.*

Je kunt vervolgens dat als excuus gaan gebruiken, maar het is inderdaad ook een keuze geweest. Ik heb ook allerlei dingen gedaan op verzoek van studenten hoor. Ik ben college logica gaan geven op verzoek van studenten. Het werkgroepen-idee heb ik destijds voorgesteld, en het is ingevoerd. Ik heb daar nog steeds heel goede herinneringen aan, werkgroep modale logica, verzamelingenleer. Ik vond die werkgroepen bijzonder boeiend, met vijf, zes studenten, die al gevorderd waren en dan samen literatuur doornemen, ze helpen een verhaal te houden en zo. Maar dat betekende ook dat je als docent vrij breed georiënteerd moest blijven.

*Zag je dat als een zware taak?*

Nee, dat vond ik leuk. Ik was er zelf enthousiast voor. Ik meende te kunnen constateren dat dat overkwam. Ik heb daar vaak positieve reacties op gehad.

*Je hebt ook een aantal promovendi gehad.*

Ja, een stuk of tien. Behalve voor de eerste, Nico Dekker, die begeleid is door Rien Kaashoek, heb ik voor alle andere toch wel flink verantwoording genomen. Ook al kwamen ze zelf met een idee, ik was gesprekspartner. Peter van Emde Boas kwam terug uit Cornell met ideeën, maar ik ben, denk ik, de enige die zijn proefschrift ooit volledig gelezen heeft.

*De manier waarop jij mensen hebt gestimuleerd bestond er ook vaak uit de mensen op het goede, bij de persoon passende, spoor te zetten, vaak een modern spoor. Is dat een bewust beleid van je geweest of ging dat vanzelf?*

Nou je weet dat ik op het MC bewust de discrete wiskunde ingevoerd heb, ter vervanging van de topologie. Het begon met een colloquium met Seidel en Van Lint als sprekers. Mensen aantrekken en ook tegen de topologen zeggen: "Ja mensen, jullie hebben een aantal jaren kunnen werken, maar nu is het tijd om iets anders te zoeken".

En ik vind dat dat ook moest. Er zijn maar beperkte middelen in een instituut als het onze. Je moet een vakgebied een tijdlang stimuleren maar op een gegeven ogenblik kun je zeggen: dat wordt aan de universiteiten gedaan. Jan Aarts in Delft, Maarten Maurice aan de VU, het MC moet wat anders gaan doen. En ik heb toen de discrete wiskunde bewust gekozen en de formatieruimte, geleidelijk maar toch, overgeheveld van topologie naar discrete wiskunde. Andries Brouwer had een la vol met topologie-resultaten liggen, maar is hier aangetrokken om discrete wiskunde te gaan doen. Jij ook. Ik wist er zelf toen niet zo erg veel van af. Maar ik dacht: dat is belangrijk, dat gaan we doen! En ik heb aan de VU daarover jarenlang college gegeven, geholpen door een uitstekende syllabus van Martyn Mulder.

*En logica?*

Dat is altijd een hobby van me geweest.

*Maar je hebt ook mensen in die richting gestimuleerd of aangetrokken.*

Ja, Krzysztof Apt, Danny Leivant, Theo Janssen, Loek Fleischhacker.

*Als chef ZW zat je ook in de Raad van Beheer van het MC.*

De oude Raad van Beheer vergaderde iedere vrijdag bij Van Wijngaarden op de kamer. Daaraan heb ik ook heel kostelijke herinneringen. Die Raad van Beheer is later omgezet in de Beleidsraad, waar ik persoonlijk van vind dat die veel minder goed functioneerde. Een versterking van de Directie en verzwakking



... koppig dwars tegen Van Wijngaarden in ...

van de positie van de andere afdelingschefs, waardoor het ook vrijblijvender werd. Met daarin toch een heel constructieve relatie tussen de afdelingschefs, zowel in de Raad van Beheer als in de Beleidsraad.

Ik heb altijd een heel plezierige relatie met Van Wijngaarden gehad. Misschien wel omdat ik ook wel eens behoorlijk met hem in botsing ben gekomen. Ik heb eens een heel duidelijk conflict met Van Wijngaarden gehad over een bepaalde zaak. Ik ben hem toen niet uit de weg gegaan en ik heb de indruk overgehouden dat hij dat respecteerde, en dat we daarna ontspannener met elkaar omgingen dan daarvoor. Daarvoor was de relatie met hem wat formeler. Ik ben eens een keer, het betreft een derde, koppig dwars tegen Van Wijngaarden ingegaan en ik respecteer en waardeer het dat dat de relatie met Aad van Wijngaarden alleen maar ten goede is gekomen. Ik heb daar heel goede herinneringen aan.

*Heb jij toen je zin gekregen?*

Het is een wapenstilstand geworden, en door slijtage is het probleem verdwenen.

### *Hoe zag jij het functioneren van Van Wijngaarden?*

Dat is een delicate vraag. Van Wijngaarden was een inspirerend man met een warme en echte belangstelling voor de wiskunde, zij het dat zijn wiskundige expertise eenzijdig was. Je kunt hem numericus noemen. Hij heeft systematisch de numerieke wiskunde opgezet en er college in gegeven. Zijn hart ging naar getaltheorie, denk ik. Hij was natuurlijk ten zeerste deskundig, hij had een diep inzicht in recursiviteit en hoe dat werkte zonder dat hij nou de theorie van de recursieve functies als zodanig in de vingers had. Maar hij wist gewoon hoe het werkte en hoe je ermee kon manipuleren.

Als directeur was hij wel duidelijk afhankelijk van de mensen om hem heen voor wat betreft de wiskunde. Hij had natuurlijk in Jan Hemelrijk een plaatsvervangend directeur, die ook al wat ouder was, heel nuchter en bestuurlijk heel goed. En die eerste jaren dat ik lid was van de Raad van Beheer had hij de inbreng van Hemelrijk, Lauwerier en mij wel nodig. Hij was informaticus, druk met de IFIP, druk met Algol 68, daar was hij de motor van. De 2-niveau grammatica waarin Algol 68 is gedefinieerd is zijn idee en dat heeft hij moeten bevechten in die werkgroep.

Daarnaast had hij het natuurlijk ook in zijn persoonlijk leven niet gemakkelijk. Wat ik daarmee wil zeggen is, dat Van Wijngaarden mede aangewezen was op de inzichten en ondersteuning van de andere leden van de Raad van Beheer. Hij was overigens ook zo dat hij je zijn vertrouwen schonk. Ik heb me altijd zeer junior gevoeld naast hem, maar hij liet dingen in vertrouwen aan mij over. Dat maakte het samenwerken met hem dus ook plezierig.

En als persoon was hij een heel aimabele, vriendelijke en breed geïnteresseerde man met wie je over alles kon praten en die overal een opinie over had, maar dan ook een beargumenteerde opinie. Iedereen die hem gekend heeft is vol anecdotes. Ik dus ook, maar daar zal ik nu niet mee komen.

*In 1980 werd je Directeur van de Stichting Mathematisch Centrum. Die functie heb je niet geambieerd. Waarom niet?*

Daar is een aantal redenen voor. Ik heb ze me recentelijk niet afgevraagd. Ik zal daar niet erg systematisch op reageren, maar daar is een aantal redenen voor.

In de eerste plaats ben ik geen wiskundige van topniveau. En voor het nationale instituut voor wiskunde en informatica zou je bij voorkeur iemand moeten hebben als Jack van Lint of zo. Iemand van de beste, de hoogste klasse die je in Nederland in de wiskunde hebt.

Dat is één overweging. Een tweede overweging is dat het een baan is met een heleboel management. En ik vind onderwijs en onderzoek veel leuker dan management. Dat is een van de redenen waarom ik destijds, toen men mij onder druk zette, uiteindelijk gezegd heb, ja, maar dan is een noodzakelijke voorwaarde dat er een aparte directeur Beheerszaken komt. Ik ben niet van

plan om al dat management echt voor mijn verantwoording te nemen, dan doe ik het zeker niet. Ik zag dus op tegen al het management-werk, maar eerlijk gezegd ook tegen het representatieve karakter van het werk. Ik voel me altijd beter als ik midden of achter in de zaal mag zitten, en vandaar uit mag luisteren en meedoen. Voorin zitten, laat staan met de voorzittershamer in de hand, dat is iets dat ik helemaal niet leuk vind. En het was van te voren al duidelijk dat dat uiteraard wel van me verwacht zou worden.

Het is me niet op het lijf geschreven, vind ik, om voorzitter te zijn, of sociale contacten te leggen en te onderhouden. Dat doe ik niet van nature, als het moet doe ik het wel en tot mijn eigen verbazing meen ik te kunnen constateren met succes. Ik heb een nuttig en breed netwerk van contacten opgebouwd, met mensen bij wie je ook echt kunt aankloppen. Maar het is niet mijn ambitie. Het is ook niet iets waar ik mezelf in uitleef.

Er waren dus allerlei aspecten aan de baan die me helemaal niet aantrokken. Nog afgezien van het feit dat er op dat moment, in 1980, een behoorlijke druk was. De middelen liepen terug, het was de periode van Bestek '81. Er moest bezuinigd worden. Ik weet nog wel dat op de Beleidsraad, waar wij als afdelingschefs probeerden posities veilig te stellen, Van Wijngaarden zei: "Ja, 2 halen, 3 betalen". Je ziet ook de formatie teruglopen. In 1981 zijn er minder mensen dan in 1980, in 1982 nog minder, het was een dalende lijn.

Daar stond natuurlijk tegenover dat ik me heel actief had ingezet — dat was min of meer mijn portefeuille binnen de Beleidsraad en daarvoor de Raad van Beheer — voor de integratie van het MC in het Nederlandse gebeuren. Ik was de contactpersoon in de Raad van Advies vanuit de Raad van Beheer. Ik was lid van de Nederlandse Commissie voor de Wiskunde en had me daar erg ingezet voor het idee van één Stichting voor de Wiskunde en niet twee. Dat was in die tijd nog een reële optie, een stichting voor het instituut en een aparte stichting voor de landelijke activiteiten. Er lag ook wel duidelijk een taak, namelijk proberen te voorkomen dat het MC zou afglijden en proberen de relaties met het onderzoek aan de universiteiten te versterken. Maar opnieuw, als je erover nadenkt hoe je dat moet doen, dan is je eerste reactie, dat weet ik eigenlijk ook niet, en vervolgens: ik zal daar veel mensen voor moeten aanspreken en zo, en dat is nou niet direct wat ik ambieerde.

*Waarom heb je het uiteindelijk wel gedaan?*

Nou, hoe zat die benoemingscommissie in elkaar? Seidel, iemand die ik goed kende en zeer waardeerde, Zandbergen, iemand die ik ook al heel erg lang ken, Tijdeman zat er in, Korevaar zat er in, Wessels en Van Est zaten er in. Nou dat waren dus allemaal mensen die ik respecteer en die ik meer of minder tot mijn vrienden reken. Die zijn ook niet gelijk met mij begonnen — in de loop van de jaren kom je zo nu en dan toch wel signalen tegen waaruit je achteraf kunt constateren hoe het proces geweest is — ik weet dus dat ze niet bij mij zijn begonnen. Maar wel bij mij zijn uitgekomen en zijn begonnen met druk op mij

uit te oefenen. En toen ben ik met mensen gaan praten. Met Van Wijngaarden, die tegen mij zei: "Als jij mijn opvolger wordt dan heb ik er vrede mee dat ik er eerder mee stop". Nou, zo iets maakte indruk.

Ik ben met alle afdelingschefs gaan praten. Ik herinner mij die gesprekken nog heel scherp. Ik herinner me zelfs bijvoorbeeld de plek nog waar ik met Hans Lauwerier praatte. Dat was in de grote collegezaal in de Boerhaavestraat. Nadat iedereen weg was, heb ik nog een tijdje met hem zitten praten en hij zei: "Ik denk dat je het heel goed zult doen, maar het zal een hele belasting voor je zijn, en ik weet niet of je dat aan kunt".

Ik heb een lang gesprek gevoerd met Van Lieshout, met Van Zwet, met Van Lint. Ik heb met verschillende mensen gesproken en daaruit groeit het gevoel dat je het misschien toch niet mag weigeren. Zonder uitzondering zeiden de mensen dat ze er vertrouwen in hadden, of zelfs positiever dat ze het toejuichten. Jack van Lint heeft duidelijk geprobeerd mij te vertellen dat ik het moest doen. En heeft me gewaarschuwd dat er aan de universiteiten donkere tijden kwamen.

En ja op een gegeven ogenblik denk je: hier kan ik niet onderuit. Het was wel zo dat in die periode, voorjaar 1980, mijn vrouw weer een moeilijke episode had, ze was niet thuis. Onze dochter was toen 10 jaar oud en die had op de lagere school te maken met het oefenen voor het landelijk proefwerk en zo. Daar moest ik nogal wat extra aandacht aan geven, die was een beetje defaitistisch op dat moment. Ik had het toen niet gemakkelijk en ook niet erg veel tijd om alles tegen elkaar af te wegen. Ik had ook niet de gelegenheid om met mijn vrouw de zaken door te praten, want die was niet aanspreekbaar. En dat maakte het niet makkelijker.

Maar ja, uiteindelijk ben ik voor de druk bezweken. "And the rest is history."

*Vond je het ook niet een "uitdaging"?*

Ook wel, natuurlijk. Het instituut van Van der Corput, Koksmas en Van Wijngaarden, daar leiding aan geven, dat is wel degelijk een uitdaging. En ik had natuurlijk wel veel nagedacht over wat je zou kunnen en moeten doen. De NCW had een werkgroep ingesteld om na te denken over de toekomstige structuur van de tweede geldstroom, twee stichtingen of één stichting, en hoe moest die dan in elkaar steken. Jan Nuis was lid van die werkgroep, Bob van Lieshout was lid. In die werkgroep hebben we nagedacht, en ik heb toen voor het eerst een diagrammetje getekend dat je eigenlijk nog steeds, in aangepaste vorm, aantreft in Annual Report en jaarverslag. Een stichting met een stichtingsbureau en een instituut met afdelingen en werkgemeenschappen daarbuiten, die ook door het bureau bediend worden, een wetenschapscommissie en zo. Ik had wel ideeën, en door de benoeming te aanvaarden was ik in de gelegenheid om die ideeën te toetsen, uit te werken, dat was een uitdaging.

*Zou het je teleurgesteld hebben als ze je niet gevraagd hadden?*

Dat is een vraag waar ik nog nooit over nagedacht heb. Nou, bij mijn eerste

uitnodiging bij de benoemingscommissie werd mij niet gevraagd of ik directeur wilde worden. De commissie wilde met een aantal mensen in het land praten. En de commissie heeft mijn mening gevraagd. Als ze dat niet hadden gedaan dan was ik wel teleurgesteld geweest, want ik vond dat ik kon meepraten over de zaak. We hebben toen gesproken over waar het naartoe zou moeten met het MC, met het instituut, met de Stichting.

En op een gegeven moment hebben ze mij gevraagd wat ze zouden moeten doen als ze geen goede kandidaat zouden kunnen vinden. Ik heb toen gezegd dat ze Seidel zouden moeten benoemen tot Directeur, als tussenpaus, waarop Seidel bijna explodeerde want die vond dat helemaal geen goed voorstel. Als ze me niet hadden uitgenodigd voor dat gesprek dan zou ik me teleurgesteld gevoeld hebben ja. Ik wilde wel meepraten over de toekomst.



... waarop Seidel bijna explodeerde ...

Maar ik was totaal niet teleurgesteld dat ze anderen vroegen voor de functie. Ze hebben mensen gevraagd waarvan ik het nu nog jammer vind dat die het toen niet gedaan hebben. Dat was veel beter geweest, van het begin af aan.

*Je kijkt niet met louter voldoening terug op je tijd als Directeur.*



Het is gemengd, er zijn hoogtepunten, er zijn dieptepunten. Er zijn perioden waarin ik met geweldig veel enthousiasme en een zekere zwier bezig was. En er zijn perioden waarvan ik het gevoel heb dat ik met lood in de schoenen door een moeras aan het baggeren geweest ben.

Als je terugkijkt heb je altijd perspectivische vertekeningen. Als je kijkt naar een landschap dan zie je de voorgrond het duidelijkst terwijl de achtergrond misschien belangrijker is. Als ik terugkijk, dan zie ik terug op het laatste jaar dat ik als uiterst onaangenaam en grauw heb ervaren. Terwijl op de achtergrond jaren zijn waarin ik met enthousiasme heb gewerkt en naar mijn gevoel ook met succes.

Als ik zou moeten proberen de zaken in perioden in te delen dan is er de periode, de eerste paar jaar, waarin ik de stiel moest leren. Ik moest nog heel veel leren. Ik heb veel van Barning geleerd, hoe je stukken maakt, hoe je dingen opzet. Ik heb veel van Nuis geleerd, de terriërachtige vasthoudendheid, het doorzettingsvermogen. Ik heb veel gehad aan contacten met iemand als Seidel als voorzitter van het Curatorium.

Maar het is geen geheim dat ik er op een gegeven ogenblik uit heb willen stappen. Dat ik heel serieus geprobeerd heb terug te gaan naar de VU, maar daar was in feite mijn positie, hoewel formeel nog beschikbaar, niet echt meer vrij. Het is geen geheim dat er persoonlijke spanningen ontstaan zijn tussen Nuis en mij. Ik heb altijd grote waardering gehad voor Jan Nuis. Voor 1980 had ik al intensief met hem samengewerkt maar op een gegeven ogenblik zijn we door onze persoonlijkheden op elkaar gebotst. We hebben geleerd dat te hanteren, allebei, maar er is toch iets van de collegialiteit, de vriendschap van voor die tijd, door beschadigd en dat betreurt ik nog steeds, dat is erg jammer. Er zijn ook anderen in meegezogen en dat is niet goed. Daar kijk ik dus niet met voldoening op terug.

Die eerste jaren waren ook financieel moeilijke jaren. Toen is er een periode van contacten gekomen, Van Spiegel, directeur-generaal Wetenschapsbeleid van het Ministerie van Onderwijs en Wetenschappen, is daarin heel erg belangrijk geweest, aan hem heb ik erg veel gehad. Want jarenlang ben ik, 3 à 4 keer per jaar, met Van Spiegel gaan praten, die nam daar ook de tijd voor, gaf me allerlei aanwijzingen, deed suggesties. Dat heeft ook geleid tot opname van het CWI in het Informatica Stimuleringsplan (INSP). Dat heeft een groei van het CWI mogelijk gemaakt. Daar kijk ik met voldoening op terug. We hebben een tijd gehad van grote bezuinigingen, Nota-Beiaard, Taak-Verdeling en Concentratie, Selectieve Krimp en Groei, het ene na het andere, overal werd bezuinigd. Maar in die tijd is het CWI tegen de verdrukking in gegroeid. Ik denk niet alleen in omvang maar ook in kwaliteit. We zijn er verbazend goed in geslaagd voor de informatica echt goede mensen aan te trekken, in een hoog tempo.

Mij is verweten — van de kant van NWO met name, maar van die kant niet alleen — dat tijdelijke middelen — vijf jaar lang twee miljoen maar ook niet meer dan dat — gebruikt zijn om vaste mensen aan te stellen. Ik ben nog steeds van mening dat ik dat terecht gedaan heb. Anders kun je geen centre of excellence

in de informatica worden. Je kunt de informatica niet opbouwen van 20% tot 50% van de onderzoeksinspanning van je instituut zonder ook vaste mensen aan te stellen. Dat kun je gewoon niet met tijdelijke mensen. Je moet nieuw onderzoek binnenshuis halen en daarvoor heb je onderzoeksleders nodig. Ja, toen de continuering van de INSP-steun maar voor een deel mogelijk bleek zaten wij met verplichtingen, vaste aanstellingen die we niet konden en trouwens ook niet wilden afstoten. Ik heb wel eens het gevoel gehad dat op een niet helemaal reële en faire wijze aan de Directie verweten is dat ze die verplichtingen heeft aangegaan.

Jaren van heel hard werken, van heel veel contacten, van ‘voor wat hoort wat’. Wij kregen geld, maar mij werd gevraagd om van alles en nog wat te doen. Ik zat in allerlei commissies en werkgroepen en zo. Keihard gewerkt in die jaren, maar toen kwam er een periode van reorganisatie. Dat is een heel nare periode geweest. Ik weet dat er mensen in het instituut zijn die daardoor beschadigd zijn maar ik ben er in zekere zin zelf ook door beschadigd.

Het had geen reorganisatie moeten worden of hoeven worden denk ik. Ik denk dat we in de weg van de geleidelijkheid — maar misschien is het heel erg naïef en argeloos wat ik zeg — dat we in de weg van de geleidelijkheid de dingen ook voor elkaar hadden kunnen krijgen. Maar zodra het op een gegeven ogenblik echt reorganisatie heette moest er ook een reorganisatieplan komen. Moesten er op een gegeven ogenblik formaties worden aangegeven, en dus ook wat er werd afgestoten. Moesten er in de laatste fase namen worden genoemd. Dat heeft allemaal veel te lang geduurd. Ik zou kunnen uitleggen waarom, maar daar begin ik niet aan. Zo’n proces moet je in een half jaar doen en het heeft wel twee jaar geduurd. Dat is traumatiserend geweest voor een heleboel mensen. Ook voor mij. Dat is dus ook niet een periode geweest waarnaar ik met genoegen terugkijk.

*Maar zeg je nu eigenlijk niet dat de reorganisatie jou uit de hand gelopen is? Het heette op een gegeven moment nu eenmaal reorganisatie, en toen moest er een plan komen, en toen moesten namen genoemd worden.*

Wij moesten de tering naar de nering zetten. Dus wij moesten bezuinigen en aangezien loonkosten onze grootste uitgavenpost vormen, betekende dat een reductie in personeelsomvang. Als je dat kunt doen in de vorm van een herstructurering zonder gedwongen ontslagen, in overleg (maar je zult zo nu en dan toch op mensen druk moeten uitoefenen) dan kun je dat in alle argeloosheid toch soepeler doen dan wanneer iets formeel reorganisatie heet. De ondernemingsraad komt tegenover de directie te staan, het onderhandelen begint, met adviseurs, vakbonden aan de ene kant, juristen aan de andere kant, van wat kun je nou wel zeggen en wat niet. Ieder woord wegen op een goudschaaltje. Zodra je formeel aan het reorganiseren bent zijn er kaders en zijn er spelregels waar je je aan moet houden. En die spelregels zijn bedoeld om mensen te beschermen, maar ze hebben een terugslag, namelijk op een gegeven moment moet je dan

ook formeel aangeven wat er te gebeuren staat. Je moet een reorganisatieplan hebben waarin staat dat bepaalde functies worden opgeheven en dat dus mensen overcompleet zijn en dan moet je namen gaan noemen.

*Was het niet mogelijk om alsnog wat op de rem te gaan staan?*

Lex, ik denk niet dat ik op de rem had kunnen gaan staan. Maar ik moet hieraan toevoegen — en dan wil ik helemaal niet mijn eigen verantwoordelijkheden ontkennen — dat op een gegeven ogenblik door de ondernemingsraad, in het overleg, is afgedwongen, misschien is dat niet helemaal het juiste woord, vastgesteld, dat gesproken moest worden van een reorganisatie, en op dat moment kun je niet meer terug. Het overleg met de OR is met grote zorgvuldigheid en met grote vasthoudendheid gevoerd; ik ga er van uit dat het kennelijk niet anders kon.

*Maar je wilt toch niet zeggen dat het eigenlijk de schuld is van de OR dat er een plan kwam en daardoor ontslagen?*

Nee, nee, dat zeg ik helemaal niet. NWO heeft een grootscheepse actie achter de rug en ze hebben kans gezien dat herstructurering te noemen en te blijven noemen. Dat is ook een reorganisatie geweest. Er zijn geen gedwongen ontslagen bij gevallen, maar er zijn posities verdwenen, maar dat is een herstructurering geweest.

Wij moesten herstructureren. Ik denk dat er geen schuldige is aan te wijzen, ik denk dat het voor een deel een autonoom proces is. Wat ik herstructurering had willen houden, is formeel een reorganisatie geworden en zodra het dat is moet je een adviesbureau in huis halen en moet je plannen maken en dan is er een stoomwals zonder rem die de helling afgaat en die je niet meer kunt tegenhouden.

*Maar moet een reorganisatie altijd tot gedwongen ontslagen leiden?*

De herstructurering die nodig was moest wel leiden tot een verminderde formatie. Dat was een absolute voorwaarde van NWO. Een voorwaarde voor het vangnet dat NWO ons wilde geven.

Ik denk dat dit niet de goede plaats en gelegenheid is om dat te gaan analyseren. Ik wil alleen zeggen dat ik het erg jammer vind dat na een aantal jaren waarin het CWI zich heel goed ontwikkeld heeft, tegen de verdrukking in is gegroeid, dat echt traumatische proces van reorganisatie kennelijk onvermijdelijk was. Ik vind dat jammer, en opnieuw: het heeft mensen beschadigd maar het heeft ook mij beschadigd. Ik ben erg gevoelig voor goede relaties met mensen en sommige relaties zullen nooit meer zijn wat ze geweest zijn, want ik ben wel verantwoordelijk en die verantwoordelijkheid ga ik ook niet uit de weg. Juist daarom til ik er ook zwaar aan.

*Hoe kijk je terug op de, soms wat gespannen, relaties met NWO, SION, de*

### *wiskundige buitenwereld?*

Je noemt weer even een aantal dingen in één adem. De contacten met NWO, dat is een vreemde zaak in zoverre dat een groot deel van die contacten niet door mij maar door Nuis werden onderhouden. Dat zat gewoon in de taakverdeling. Eens per jaar was er een hearing van GB-E over de noden en de daaruit volgende budgettaire behoeften van de stichtingen. Daar zat je met alle E-stichtingen en GB-E rond de tafel en iedereen mocht even wat zeggen en iedereen vertelde natuurlijk hoe slecht het met ze ging en dat zij in ieder geval niet gekort moesten worden in subsidie volgend jaar. Nou, dat was geen reëel contact.

Contacten met het Algemeen Bestuur zijn er nauwelijks, afgezien van een werkbezoek. Nogmaals, de eigenlijke contacten werden door de directeur Beheerszaken onderhouden. Dus daar heb ik niet zoveel directe inbreng in gehad. En daar heb ik dus ook niet te veel triomfen of teleurstellingen te melden. Verder zijn er contacten geweest echt op hoog bestuurlijk niveau, tussen voorzitter en voorzitter en dat soort zaken. Opunieuw ben ik daar meer iemand aan de zijlijn, dan dat ik daarbij rechtstreeks betrokken ben.

De relatie met SION is altijd wel turbulent, vurig, impulsief, emotioneel geweest, maar dat heb ik altijd boeiend gevonden. Ik heb geen nare nasmaak of vervelende herinneringen aan de relatie met SION. Je moet wel leren met mensen om te gaan. Met voorzitter Van de Riet moest je op een andere manier omgaan dan met voorzitter Hertzberger. Uiteindelijk kan ik het met allebei goed vinden. Ik beschouw Reind van de Riet als een van mijn goede vrienden. Ik ben iets minder persoonlijk bevriend met Bob Hertzberger, maar ik kan het uitstekend met hem vinden, ook al zijn we het zo nu en dan niet met elkaar eens. En de spraakmakende informatici, er is niemand met wie ik het gevoel heb dat er spanningen zijn tussen hem en mij. Wel eens meningsverschillen maar die zijn zonder meer bespreekbaar. Dus, de relatie met SION staat natuurlijk vanzelfsprekend onder druk, dat zit hem in de constructie. Dat zit hem in het feit dat SION de tweede geldstroom-stichting is voor de informatica, en de helft van de tweede geldstroom informatica gaat via het CWI. En in het feit dat SION geen formele zeggenschap over het CWI heeft.

*Ze hebben daarom misschien het gevoel dat ze buitengesloten worden.*

Dan moet je dus een constructie zien te vinden dat ze niet buitengesloten worden. Ik heb altijd gezegd dat je een instituut niet als integraal instituut kunt besturen als er via twee geldkranen door twee partijen meegestuurd wordt, dat kan niet. Je moet als instituut één bestuur hebben en geen twee. En je moet als instituut een vast basissubsidie hebben. Maar vervolgens moet het mogelijk zijn om SION een echte, reële greep op het onderzoek op het MC te geven. Kennelijk is dat nog steeds niet volledig overtuigend gelukt. Nog niet zo heel erg lang geleden heb ik een hele dag in Amersfoort mogen voorzitten waar door informatici binnen en buiten het CWI gesproken werd over strategische plannen,

en ik had het idee dat we toen aardig op weg waren om elkaar te vinden. Daar is de klad in gekomen doordat andere dingen de aandacht zijn gaan opeisen. Het laatste jaar met name. Ik geloof nog steeds dat het mogelijk is om een echte, inhoudelijke, reële betrokkenheid van het SION-bestuur met het onderzoek hier te realiseren zonder dat de integriteit van het CWI daaraan wordt opgeofferd. En het is meer een uitdaging om dat nu eindelijk eens een keer goed te doen dan dat ik vind dat ik daar met gemengde of nare gevoelens op terug moet kijken. Het is een klus die ik niet afgemaakt heb.

*Maar waarom is er nooit gekozen voor één stichting wiskunde en informatica, waarmee je toch veel sterker staat?*

SION wilde dat niet. De informatici wilden dat niet. ZWO, destijds nog, heeft de psychologische fout gemaakt die ouders moeten leren niet te maken. ZWO heeft er namelijk erg op aangedrongen. Niet voor niets is SION tien jaar lang een stichting in oprichting gebleven. ZWO bleef zijn erkenning onthouden want ZWO vond dat SION en SMC samen moesten in één stichting. Nou ja dat is het varkensprincipe. Als je een varken vooruit wil hebben, dan moet je aan zijn staart trekken, want als je tegen zijn achterste gaat duwen dan loopt hij je achteruit omver.

*Waarom wilde SION dat niet?*

Ik heb een vergadering hier meegemaakt, toen SION nog opgericht moest worden. Ik heb toen aangeboden, helemaal van harte, om aan werkgemeenschappen informatici alle ruimte te geven binnen de SMC. Alles wat aan bureau en ondersteuning beschikbaar is. Het was Blaauw, hij was nog geen curator, die op zijn bedachtzame en rustige manier zei dat hij 'geen voorstander was van inwoning bij de wiskunde, want inwoning duurt als regel veel te lang'. Blaauw, echt een van de meest bedachtzame, wijze en coöperatieve informatici, was er duidelijk voorstander van om een eigen organisatie te ontwikkelen. De jongeren waren het daar alleen maar des te meer mee eens. Het zat er gewoon niet in. Ik heb geprobeerd dat aan Van Lieshout duidelijk te maken. De informatici willen het niet, geef ze dan de ruimte zoals ze het zelf willen. En ik ben nog steeds van mening dat dat de enige verstandige manier is. We hebben als Stichting MC dan ook iedere keer geadviseerd aan ZWO/NWO om SION wel te erkennen en ik vind nog steeds dat wij niets anders konden doen. Ik zou het overigens geweldig mooi vinden als er in de toekomst toch nog eens een fusie zou komen.

Het AB van NWO heeft opnieuw, bij zijn herstructurering, wel degelijk — laten we zeggen — gehoopt dat SION en SMC zouden samengaan. Ze hebben het iets minder expliciet uitgesproken dan destijds ZWO. Maar ja, uiteindelijk vind ik dat men moet respecteren wat de onderzoekers in een bepaald gebied zelf willen. En je kunt veel beter twee goed samenwerkende stichtingen hebben die na verloop van tijd concluderen dat het toch efficiënt zou zijn om meer dingen samen te gaan doen, dan dat je mensen dwingt tot, wat de Fransen een



*cohabitation* noemen.

*Blaauw zag het als inwoning, maar dat is toch iets anders dan samenwoning?*

Zo is het in Frankrijk gebruikt met betrekking tot de socialistische president en de huidige en een vorige regering. Zo bedoelde ik het dus, in die termen. Een accommodatie van twee partijen die nolens volens met elkaar verder moeten.

*Er is in jouw directeurs tijd ook steeds meer druk ontstaan vanuit de wiskundige buitenwereld op het CWI, ook kritiek.*

Die kritiek is er altijd geweest. Ik ervaar het niet dat de druk groter is dan vroeger. Het is natuurlijk wel zo dat het totstandkomen van onderzoeksscholen noopt tot een herbezinning over wat dan precies de taak en de positie van het CWI is. De onderzoeksscholen hebben financiering nodig, die kijken natuurlijk ook naar de tweede geldstroom. En die vragen zich ook af in hoeverre taken, met geld en al, van het CWI misschien naar onderzoeksscholen zouden kunnen overgaan in de loop van de tijd. Dat soort geluiden hoor ik wel.

Het antwoord van SMC en het CWI moet zijn: een herbezinning op eigen taken, en vervolgens duidelijk maken aan beleidsmakers in onderzoeksscholen, bij NWO, en waar dan ook, dat die eigen taak zo waardevol is, zo ondersteunend, complementair. Dat op zichzelf vind ik ook een uitdaging. Dat vind ik niet iets om benauwd over te zijn, ook niet iets om over te chagrijnen. Vooral de laatste twee jaar werd er in een bepaalde hoek van de Wetenschapscommissie weleens uitgeprobeerd hoe ik, hoe het CWI, op bepaalde dingen zou reageren. Ik vind dat dat moet kunnen. Het is misschien niet altijd even plezierig, maar goed. That's in the game. En als CWI moeten we een voldoende duidelijk zelfbeeld hebben en voldoende goed argumenteren waarom we dat zelfbeeld hebben. Dat we ons daar tegenover staande kunnen houden. Opnieuw vind ik dat meer iets dat de zaak spannend maakt, dat de zaak levend houdt, dan dat ik daar nare gevoelens van heb. Nare gevoelens heb ik van beschadigingen van personen en persoonlijke relaties, niet van belangentegenstellingen. Daar kun je zakelijk over praten, daarover kun je het ook heel goed met elkaar oneens zijn en vervolgens samen een biertje gaan drinken.

Persoonlijke relaties die beschadigd worden, dat is veel erger. En dat heb ik in deze veertien jaar van tijd tot tijd meegemaakt. Van vrij in het begin af aan, in verschillende context, en dat maakt dus dat ik met gemengde gevoelens op de tijd terugkijk. Het is echt niet alleen maar een leuke tijd geweest.

Er zijn wel hoogtepunten en leuke ogenblikken geweest. Ik heb heel veel voldoening beleefd aan de oprichting van ERCIM en de daaruit voortkomende samenwerking. De eerste periode daarvan was uitermate constructief. Seegmüller en Bensoussan en ik, met zijn drieën probeerden wij te zorgen dat de onderzoeksgroepen elkaar leerden kennen, dat de instituten wat meer van elkaar op de hoogte raakten, dat we wat van elkaars cultuur gingen begrijpen. Een aantal van de mensen beschouw ik echt als mijn vrienden. Met hen heb ik een heel



... informatica en toegepaste wiskunde zijn voor de toekomst van ons en onze kinderen tenminste even belangrijk als ruimtevaart en hogere energie fysica ...

plezierige relatie opgebouwd. ERCIM groeit nu zo snel dat het moeilijk is om goed vast te houden aan wat je ermee wilt bereiken.

*Wat wil je ermee bereiken? Ik neem aan dat het niet alleen om de persoonlijke relaties gaat.*

Nee. Wat ik wil bereiken is samenwerking op organisatieniveau, waar mogelijk contractonderzoek, uitwisseling op het gebied van fundamenteel onderzoek tussen de wiskundigen, met name de toegepaste wiskundigen, en de informatici in Europa. Om op die manier voor ons vak te kunnen opkomen. Wat de fysici al jaren met groot succes doen rondom CERN e.d. Wat de astronomen en de ruimte-onderzoekers met groot succes doen om nationale en internationale ondersteuning te krijgen, de publieke opinie te actualiseren voor hun onderzoek. Dat moeten wij voor informatica en toegepaste wiskunde ook kunnen. Want informatica en toegepaste wiskunde zijn voor de toekomst van ons en onze kinderen tenminste even belangrijk als ruimtevaart en hogere energie fysica. Onze wereld wordt gestructureerd door allerlei toepassingen van computers en van informatieverwerking. Onze toekomstige beschaving wordt daardoor gedomineerd. Wij moeten daarin een eigen stem ontwikkelen en dat kunnen we alleen maar als we gaan samenwerken en als we niet iedere keer met elkaar concurreren of niet eens weten wat er aan de overkant van de grens gebeurt. Ik wil dus gewoon een grotere samenhang, een grotere cohesie, een groter aggregatie-niveau



voor het onderzoek in de informatica. Een groter, ook informeel netwerk. Het is pas een succes als je ook de onderzoekers in de industrie, in het bedrijfsleven daarbij betreft. De mensen die bezig zijn bij de banken om informatiebeveiliging toe te passen. Bij de PTT's, bij industrieën op het gebied van de IT, die er in Europa nog wel zijn, ook buiten Philips, Siemens en Bull. Die moeten van elkaar weten wat voor mogelijkheden er zijn, welke mensen je daarbij kunt inschakelen, waar je nou in Brussel, Straatsburg, of bij de nationale regeringen moet lobbyen. Dit is een onderwerp waar ik echt in geloof.

*Het is jammer dat je je werk door je ziekte niet goed hebt kunnen afronden.*

Ja. Een jaar geleden ben ik opnieuw ziek geworden. Kennelijk heb ik daar aanleg voor. Eigenlijk onder dezelfde omstandigheden als in 1969-1970. Ontwikkelingen in mijn werk die ik moeilijk vond, gecombineerd met ontzettend vervelende, nare, traumatische ontwikkelingen in mijn persoonlijke leven, en die combinatie is kennelijk niet zo gezond voor me.

*Emotioneert het werk je ook?*

Soms ja. Ik ben daar heel emotioneel bij betrokken ja.

*Te veel volgens jou?*

Nou dat weet ik niet, vermoedelijk functioneer je efficiënter als je niet emotioneel wordt maar zonodig wel kunt spelen dat je het bent. Maar ik ben niet zo goed in simuleren en als je emotioneel wordt dan kun je daar ook energie aan ontlenen. En ik heb dat in het verleden ook wel eens bewust gebruikt. Vechtend voor het CWI. In Den Haag.

*Lag je ook wakker van je werk?*

Ja, daar kan ik echt wel van wakker liggen. Kijk als ik lees, ik geloof gisteren in de krant, dat Minister Wijers vindt dat het technologisch onderzoek veel te veel navelstaarderig bezig is (dat zijn mijn woorden), veel te waardevrij bezig is en het moet allemaal anders en er moeten Centres of Excellence komen, dan heb ik op dat moment de neiging emotioneel te worden van: verdikkeme nog aan toe, Deetman en Van Aardenne samen hebben bij het INSP-gebeuren uitdrukkelijk het MC de opdracht gegeven nationaal Centre of Excellence for Computer Science te worden. In de rapportage aan de Kamer heeft Deetman die opdracht nog vier jaar daarna herhaald. Ik moet naar Wijers toe en ik moet hem daar mee confronteren, en ik moet hem daaraan herinneren.

*Je bent waarschijnlijk in dat opzicht ambitieus, je wilt alle dingen goed doen.*

Je hoeft niet alles even goed te doen maar je moet het wel allemaal bijhouden. Je kunt je niet veroorloven en zeggen: We laten de wiskunde een tijdje aan zijn

lot over en we gaan verder met de informatica, zo werkt dat niet. Of nu gaan we eens een tijdje aan de landelijke activiteiten werken, het instituut komt de volgende keer wel weer. Nee, dat moet je allemaal tegelijk proberen actief te houden. In de aandacht van de mensen en in je eigen aandacht te houden.

*Zijn er concrete dingen waarvan jij zegt: dat had ik echt beter moeten doen?*

Ja maar die heb ik niet op een rijtje. De relatie met SION is niet af. Er zijn natuurlijk dingen, ik zeg het nu in een bijna lege platitude, die ik met de ervaring, het inzicht en de kennis die ik nu heb beter had kunnen doen, als ik die kennis eerder had gehad. Spanningen in de relatie met Jan Nuis, waren niet nodig geweest als ik eerder begrepen had waar bepaalde gevoeligheden liggen. Ik betreur die spanningen ten zeerste en achteraf zeg i dat had niet hoeven. Maar het was misschien ook onvermijdelijk want pas door er doorheen te gaan merk je hoe het anders had gekund.

*Vergelijkbaar met de reorganisatie?*

Ja, vergelijkbaar met de reorganisatie. Er zijn dingen die niet af zijn. Laat ik één ding noemen waarvan ik vind dat dat ook niet goed is zonder dat ik bereid ben nu te analyseren hoe dat zou kunnen komen. De kernwiskunde in dit instituut is te zwak geworden. De afdeling AM is te klein, te smal, is bijna geheel teruggevallen op twee onderwerpen, algebra, met name computer-algebra, en niet-lineaire dynamica, met name mathematische biologie-modellen. Dat is te smal voor dit instituut. Dat is jammer, dat is niet goed.

*Zie je dat als een taak voor een opvolger?*

Ik denk dat het voor het instituut heel goed zou zijn als de kernwiskunde, de grondmethodologieën, verbreed en versterkt zouden worden. Voor computer-algebra — ik sta in zoverre aan de wieg van CAN dat ik voor het eerst een suggestie heb gedaan op een bijeenkomst om een beroep te doen op die pot expertise-centra. RIACA is terug te voeren op een brief die ik aan NWO heb geschreven destijds. Willen CAN en met name RIACA een succes zijn dan moeten we ze ook wiskundig, vanuit fundamenteel onderzoek, vanuit het CWI ondersteunen. Daarvoor is een sterkere groep nodig dan er nu bij AM zit. Dus enerzijds moet er geïnvesteerd worden in de algebra, niet alleen de computer-algebra maar ook de theoretische algebra en de algoritmiek, ten behoeve van CAN en RIACA. Anderzijds moet de basis verbreed worden. Er zou hier eens een goed project meetkunde moeten komen, echte goede differentiaal-meetkunde bijvoorbeeld.

*Omdat wij dat nodig hebben?*

Omdat dat een uitstraling zal blijken te hebben. Als je differentiaal-meetkunde kiest, ontstaat er als het goed is samenwerking met controltheorie. Op

een gegeven moment ontstaat er ook samenwerking met de biomathematica en de niet-lineaire modellen. Dan blijkt je daar ook symbolisch rekenen voor te kunnen gebruiken, en raakt CAN erbij betrokken en je krijgt een nieuwe groep mensen in het land die met enige interesse volgen of het experiment lukt of niet.

*Vind je het achteraf, alles overziende, jammer dat je geen biologie bent gaan studeren?*

Ik heb nooit spijt gehad van wiskunde. Ik vind het misschien achteraf jammer dat ik niet meer wiskunde heb gestudeerd. Maar biologie, ja en nee. De laatste tien boeken die ik gelezen heb gingen over biologie.

*Dit sluit misschien aan bij de vraag: Wat ga je na je afscheid doen?*

Een medicus met wie ik niet zo lang geleden sprak zei tegen mij dat ik heel duidelijk nog met het gezicht naar het verleden sta en met mijn rug naar de toekomst. En ik denk dat dat waar is. Ik weet nog niet wat ik na mijn afscheid ga doen. Ik weet dat ik op de VU nog welkom ben. Ik heb daar een heel kleine aanstelling, maar het is wel een vaste aanstelling, en ik hoop dat ik die mag houden en dat ik daar nog iets kan opbouwen, onderwijs, of wellicht zelfs betrokken kan raken bij onderzoek, bij afstudeerders of wat dan ook. Maar dat is geen weektaak. Ik moet mijn baan hier afronden. Ik ben bezig mijn afscheidsverhaal op papier te zetten en dan moet ik verder hier alles leeg gaan halen en mijn archieven gaan opruimen of weg doen, dan ga ik vervolgens een andere woonplaats zoeken buiten de Randstad. Ook om persoonlijke redenen. Ik moet nog beginnen te zoeken, ik heb nog geen flauw idee waar, dat gaat ook tijd kosten. Dus ik ben in ieder geval voorlopig nog bezig met het afhechten, het afronden van het verleden. En ik kan eigenlijk pas goed nadenken wat ik daarna ga doen als ik die deur achter me gesloten heb.

Wat eigenlijk veel te weinig in dit gesprek aan de orde is geweest en waarvoor ik verwacht veel meer tijd te zullen hebben, zijn mijn kinderen en kleinkinderen. Zij hebben altijd heel heel veel voor mij betekend, en zijn ook een belangrijke steun voor mij geweest zowel in mijn professionele als in mijn privéleven.

Ook hoop ik meer tijd te hebben voor een aantal liefhebberijen, zoals de natuur, de tuin, fotograferen (in het bijzonder bloemen en vogels), geschiedenis, poëzie, (middeleeuwse) muziek.

*Je zei: ik ben nog welkom op de VU. Ga je er van uit dat je op het CWI niet meer welkom zou zijn, of is het meer dat je zelf niet wilt?*

Ik zal hier voorlopig niet komen. Ik ben ervan overtuigd dat ik op allerlei niveaus welkom ben maar ik voel me toch op een andere wijze niet thuis. Ik voel me ongemakkelijk. Ik heb een erg ongemakkelijk jaar achter de rug, en die ervaring wil ik niet voortzetten.

*Maar je bent waarschijnlijk al een aantal jaren niet op de eerste en tweede*

*verdieping geweest. In de bibliotheek bijvoorbeeld, waar je vroeger altijd graag kwam.*

Ik kom zelfs niet meer in de kantine. Er is een situatie ontstaan waarin ik mij hier erg ongemakkelijk voel. En ik denk niet dat ik die verbreek door bijvoorbeeld naar de bibliotheek te gaan. Laten we het er maar op houden dat er bij mij nog steeds een paar moertjes los getrild zijn en die moeten weer aangeschroefd worden.

*Ik denk dat velen je nog vaak hier hopen te zien.*

Een zekere mate van koppigheid is mij niet vreemd. Ik ben niet van plan om, als ik hier de deur achter mij dichtgetrokken heb, binnen afzienbare tijd terug te komen, daar heb ik mijn redenen voor.

*Het 50-jarig jubileum, zien we je dan?*

Een van de levenswijsheden die de Angelsaksische taal ons geleverd heeft is: Never say never. Laat ik het antwoord op die vraag dus maar schuldig blijven.

*Wordt het een verrassing?*

Ik meen het antwoord te weten maar ik spreek het niet uit.

*Cor, hartelijk bedankt voor het interessante gesprek.*

Dit interview werd afgenomen op 27 oktober 1994, met medewerking van Miente Bakker. Met veel dank aan Coby van Vonderen voor het vervaardigen van de transcriptie.



# Scientific Contributions





# Operational Operations Research at the Mathematical Centre

Ko Anthonisse and Jan Karel Lenstra

This reprint is dedicated to Cor Baayen on the occasion of his retirement as scientific director of CWI. It was written in 1982, when CWI and NWO were still Institute MC and ZWO. Just before, the Foundation MC had confirmed its national position by becoming the 'Netherlands Foundation for Mathematics'. Shortly afterwards, the Institute MC changed its name to CWI and started to play a leading role in the national computer science stimulation program. These events, which are milestones in the history of the MC, were largely due to Baayen's efforts.

The paper was published before in the *European Journal of Operational Research* (Volume 15, 1984, pp. 293–296) and, in its original Dutch version, in *Kwantitatieve Methoden in het Management*, edited by C.B. Tilanus, O.B. de Gans and J.K. Lenstra (Het Spectrum, Utrecht, 1983, pp. 252–258). We are grateful to Elsevier Science Publishers B.V. for their permission to reprint it here. We have chosen to make only a few editorial changes, and hope that the paper reflects the spirit of the time of its first publication.

This note deals with consultation in operations research at the Mathematical Centre in Amsterdam. After a short description of the activities of the MC, in particular of its Department of Operations Research and System Theory, three practical projects are described.

## 1. INTRODUCTION

The Mathematical Centre (MC) was established in 1946 as a nonprofit foundation for the promotion of mathematics and its applications.

The Institute MC has six scientific departments: pure mathematics, applied mathematics, numerical mathematics, mathematical statistics, operations research and system theory, and computer science. Among the supporting non-scientific departments, the library and the printing office play a role of national importance. The MC is sponsored by the government through the Netherlands Organization for the Advancement of Pure Research (ZWO). Next to pure research, the MC also carries out consulting activities in the private and public sectors. The budget for 1980 amounted to more than thirteen million Dutch

guilders, of which about 85% was provided by ZWO. The institute employs around 150 people.

The Foundation MC was recently appointed to coordinate, stimulate and evaluate mathematical research at universities to the extent that it is being financed by ZWO. To this end, a number of research communities in several branches of mathematics were created. This new task is mentioned here to emphasize the central position of the MC within Dutch mathematics, although it is not of primary relevance to the subject matter of this paper: consultation in the area of operations research.

The Department of Operations Research and System Theory is engaged in the investigation of mathematical models and methods that could support optimal actions in decision situations. The motivation originally came from problems in economics and industrial engineering, and today is also found in communication and control and even in the political and social sciences.

These investigations entail the study of a wide range of mathematical subjects, such as complexity theory, combinatorics, probability theory and differential geometry. The unifying element is the potential applicability of the models and methods under investigation. Consequently, the department tries to become involved in projects that lead to original and advanced applications in areas in which it has expert knowledge. Such projects can vary from answering specific questions explicitly to participating in development research, with the purpose of making new theory applicable in practice.

The involvement in practical projects is an essential part of the department's scientific policy. The current research projects and the main application areas are:

- *combinatorial optimization*, i.e., the determination of optimal distribution systems, depot locations, room assignments, timetables, production plans, cutting patterns, and other discrete structures;
- *analysis and control of information flows in networks*, such as computer networks, telecommunication systems and networks of queues;
- *system and control theory*, in particular prediction, filtering, nonlinear control, system identification and time series analysis.

Experience has shown that consultative activities often lead to innovative applications as well as to intriguing mathematical problems and results. This will be illustrated below on three practical projects. They were carried out by Antoon Kolen, Ben Lageweg, Leen Stougie, Koos Vrieze, and the authors.

## 2. PLAYING FOR KEEPS

A consortium of four international contractors had completed a large dredging contract. An inventory of equipment was left over, consisting of 268 items, including crane ships, barges, rock breakers and smaller items such as pipes and spare parts. An independent consultant had established a price for each item. The total value of the inventory was about \$24.8 million. Since the inventory

could not be sold locally, it was agreed among the partners that each would buy a quarter of the lot. It was also agreed that the allocation of the items to the partners would be determined by an auction, as it was virtually impossible to decide on this by straightforward negotiations.

The auction consisted of 25 rounds. In each of the first 24 rounds each partner would be allotted \$0.25 million to buy items or to save money for subsequent rounds. It was not allowed, however, to save an amount exceeding the price of the most expensive unsold item. In the last round the remaining budgets must be spent. The order in which the companies should buy from the inventory in the first round would be determined by drawing lots:  $S_1 = (1, 2, 3, 4)$ . The order in the subsequent rounds followed from that in the first one: in the second round  $S_2 = (2, 3, 4, 1)$ , in the third  $S_3 = (3, 4, 1, 2)$ , and in the fourth  $S_4 = (4, 1, 2, 3)$ . Then a second cycle of four rounds would follow:  $S_2, S_3, S_4, S_1$ . The third cycle would start with  $S_3$ , and so on.

Our client, one of the partners, had composed a listing of the items with the price of each item, its attractiveness for the company and guesses of the preferences of the others. The attractiveness was defined as a classification into categories *A* (very attractive) to *E* (scrap). Category *A* contained three expensive crane ships; five to six rounds of saving would be necessary to acquire one. Due to the extended production times for new cranes it was expected that each partner would try to obtain at least one of these.

A program was developed to keep track of purchases and savings of each partner and to provide information, if requested, such as lists of attractive items that could be bought in the present or the next round. This program was run on the company's computer and used on-line by the delegation at the auction.

Our assignment was to develop a strategy to obtain as much attractive equipment as possible. An analysis of the inventory and the rules of the game provided useful information. It was impossible to avoid buying from category *E*; minimization of this was selected as the primary objective. The drawing of lots was intended to provide equal opportunities but did not do so; e.g., the company which draws 2 is preceded by company 1 in most rounds and thus is at a disadvantage if both prefer the same items. Another problem was the end-game. It was possible that only expensive items would be left at the end and no partner has sufficient savings to buy. Even if this situation did not occur, at least \$0.8 million worth of material would be left at the end, without rules for allocating it.

For this auction, various strategies are conceivable. With the help of a simulation program five strategies were investigated, each with and without going after the cranes. The auction was simulated for over 50 combinations of these basic strategies for the competitors and assumed preferences for the items, and the results were analyzed.

According to the simulations, the outcome for the company would depend to a large extent on the drawing of the lots. At the beginning of the game as little as possible should be saved. It was most advantageous to buy an item requiring several rounds of saving in one of the last few rounds. The round in which to

start saving depended upon the outcome of the lottery. While saving, one might not spend more than the competitor with the closest amount of savings. Much attention should be paid to items priced \$0.25 million or less, which do not require saving. Finally, the game would probably reach a deadlock, so the partners should define rules for the endgame.

With the help of the bookkeeping program and the selected strategy, our client succeeded in spending only 10% of the budget in category *E* and very high percentages in categories *A* and *B*. This is in spite of an unfavourable starting position and an arbitrary allocation of the leftovers in the last round. The estimates of benefits ranged between \$0.25 and \$1.50 million. An objective estimate is impossible since the real preferences of the other partners are unknown. One of them purchased different items than was expected. In general, the others tried to optimize in each round, whereas our client pursued an overall optimum.

This consultation was not remarkable by the problem and the results alone, but also because we were given only ten days. Nevertheless, as intensive and effective preparation for the auction was feasible.

### 3. AFTER THE LAST RIDE

The public transportation service in one of the major Dutch cities, which operates 16 tramway lines and 270 tramcars, was interested in an optimal allocation of trams to depots. On each line, a number of trams runs between both endpoints. After the last ride at night, each tram goes to one of seven depots. Each depot has a limited capacity. A ride to the depot costs a certain amount that, among other things, depends on the length of the ride. The problem is to allocate trams to depots in such a way that the total costs of the depot rides are minimized.

Three variants of this problem were of interest. The first one represents current practice: all trams of the same line are allocated to the same depot. This stimulates contact among the drivers on one line. The second variant can yield savings: the trams of a line that make their last ride to the same endpoint have to go to the same depot. The lower costs were to be weighed against the inconvenience of dividing personnel. The third variant is the cheapest one: each tram can go to any depot. This was not considered to be a feasible alternative, but it would set an informative lower bound on the minimum total costs.

Out of the seven depots mentioned above, only three do exist in reality. Two of the four fictitious depots are locations at which a depot could be built. Before deciding to do this, one wanted to have a definite estimate of the potential savings as a function of the capacities to be chosen. The other two fictitious depots are in fact two new routes to an existing depot. Building those routes could diminish noise pollution at night, again depending on the capacities to be chosen — i.e., the number of trams that would be allowed to use the routes. One was interested in the relation between operating costs and usage of one or both new routes. All this led to 40 combinations of depot capacities for each variant. Therefore, a total number of 120 problems had to be solved.

The mathematical formulation of these problems was obvious. The third variant is nothing but a linear transportation problem, for which standard techniques yield integral solutions; there is no concern that a tram will be split over several depots. The first and second variants have side constraints to enforce that all trams of the same line (or of the same endpoint) go to the same depot; this required the introduction of a 0-1 variable for each of the  $16 \times 7 = 112$  line-depot combinations (or for each of the  $32 \times 7 = 224$  endpoint-depot combinations).

The resulting integer linear programming problems were solved by the APEX system of Control Data, which is available on the Cyber 175 – 750 of SARA (Foundation Academic Computing Centre Amsterdam). This program computed good to very good solutions at reasonable costs.

The computations were organized as follows. For each variant, our general LP matrix generator produced an input file for one combination of capacities. After that, a special procedure handled each case by substituting a combination of capacities in the input file, calling the APEX system, and adding the solution to an output file. Finally, a report generator made manageable surveys of all solutions.

The problems have also been solved for possible future situations involving increased capacities or modifications of the network.

As has been stated before, one had no intention to simply implement the best solution in practice. Many of the variants and situations considered are unrealistic, but this very feature does contribute to the value of the collected results as an aid to decision making. The transportation service is now investigating the possibility of changing to an allocation rule according to the second variant; this would yield substantial savings.

#### 4. NASTY CLIENTS

A Dutch firm, primarily engaged in the retail trade, had decided to diversify and had acquired a large number of summer cottages. A client can make a reservation at any of the firm's branches and is immediately told whether a cottage is still available for the period (s)he is applying for. Only at a later stage it is determined in which cottage each accepted client has to spend the holidays. This procedure led to a couple of questions.

Does there exist a simple rule that indicates whether a client can be accepted? Yes, there does: cottages can be assigned to clients in their desired periods if and only if, at any time, the number of clients is no larger than the number of cottages. How about a method that assigns the accepted clients to a minimum number of cottages? This exists as well: assign the clients to cottages in order of their starting times, giving priority to cottages used before.

As early as 1954, more complicated versions of these problems were solved by G.B. Dantzig and D.R. Fulkerson, founding fathers of operations research, as witnessed by the existence of the Dantzig Prize and the Fulkerson Prize. The questions asked were closely related to our research in machine sequencing and scheduling, so the answers could be given offhand during the first (and, as



it turned out, the last) contact with our potential client.

While leaving, a trivial complication crossed his mind: a client can reserve a specific cottage by paying Dfl.25 upon application and is then preassigned. This has a dramatic effect on the problem's computational complexity. So far, we had identical cottages and nonidentical clients; but now the cottages are nonidentical as well. The question whether a client who expresses no preference can be accepted boils down to the following problem: is it possible to pack  $n$  given time intervals (the unassigned clients) into  $m$  other given time intervals (the idle periods of the cottages)?

This is a beautiful combinatorial problem, but it has not been dealt with previously in the literature. The above necessary and sufficient condition for acceptance remains valid only under the (false) assumption that a client would be willing to move into another cottage now and then. It recently turned out that the problem is solvable in polynomial time for each fixed  $m$  and NP-complete for arbitrary  $m$ . The complexity theorist is delighted by such a classification. However, the polynomial method is not practicable for realistic values of  $m$ , and NP-completeness does not imply absolute insolvability. It seems very well possible to develop an algorithm that resolves most cases fairly quickly — although one can always construct an artificial instance that keeps the computer running until holidays are over.

We never heard from our client again: the complications caused by the nasty clients are probably trivial indeed and do not prohibit the application of the existing methods.

Is this an example of a consultation that *failed*? No, it rather is the *reverse* of a consultation. A practitioner saddled us with a problem that, after all, was of no concern to him. Continuing research on this problem is a task of the Mathematical Centre. It is primarily motivated by our professional curiosity, but also by possible practical demands in the future.

NOTE ADDED IN PROOF

On September 1, 1983, the Institute MC changed its name to CWI (Centre for Mathematics and Computer Science).

# Comparing Negation in Logic Programming and in Prolog

Krzysztof R. Apt

CWI

and

*Faculty of Mathematics and Computer Science  
University of Amsterdam, Plantage Muidergracht 24  
1018 TV Amsterdam, The Netherlands*

Frank Teusink

CWI

•

Many aspects of Artificial Intelligence can be clarified and made rigorous by using tools and concepts originating in mathematical logic. Cor Baayen has stimulated this research programme at CWI. This paper provides an example of this form of work and is offered to him at the occasion of his retirement from CWI. The second author is a PhD student employed by SION. His coauthorship is a tribute to Cor Baayen's successful efforts of ensuring a smooth cooperation between CWI and SION.

Mathematical logic has played a useful role in clarifying concepts and ideas advanced in Artificial Intelligence. However, for specific applications it is often needed to modify and extend well-known logic formalisms, sometimes in an unusual way.

A case in point is the treatment of negation in Prolog. To properly render its meaning and compare formally its use to that in logic programming we had to extend the customary logic programming formalism by allowing variables standing in atom positions (so called *meta-variables*) and adopting ambivalent syntax.

To define the computational process of Prolog one needs to define formally backtracking, which is an algorithmic concept. We found a simple account of it by means of a single operation on finite ordered trees. To deal with the cut operator one more operation is needed.

After taking care of these matters we establish a formal result showing an equivalence in appropriate sense between these two uses of negation – in Prolog and in logic programming. This result allows us to argue about correctness of various known Prolog programs which use negation by reasoning about the corresponding logic programs.



This paper is a shorter version of a chapter from *Meta-programming in Logic Programming*, K.R. Apt and F. Turini (editors), The MIT Press, (in preparation).

## 1 INTRODUCTION

During the last 15 years, a lot of attention was devoted to the study of negation in logic programming. No less than seven survey articles on this subject were published. Just to mention two most recent ones: Dix [Dix93] and Apt and Bol [AB94].

The main reason for this interest is that in the logic programming setting negative literals can be used to model non-monotonic reasoning. The computation process of logic programming provides then a readily available computational interpretation. This is not the case with other approaches to non-monotonic reasoning. This computation process is called SLDNF-resolution and was proposed by Clark [Cla78]. Negation is interpreted in it using the “negation as finite failure” rule. Intuitively, this rule works as follows: for a ground atom  $A$ ,

$$\begin{aligned} \neg A \text{ succeeds iff } A \text{ finitely fails,} \\ \neg A \text{ finitely fails iff } A \text{ succeeds,} \end{aligned}$$

where “finitely fails” means that the corresponding evaluation tree is finite and all its leaves are marked as failed.

However, SLDNF-resolution is not a practical way of computing and usually one resorts to Prolog when seeking for a computational interpretation. But in Prolog negation is implemented in a different way, namely by the predicate (or synonymously relation symbol) `neg` defined internally by the following two clauses:

$$\text{neg}(X) \leftarrow X,!,\text{fail}. \quad (1)$$

$$\text{neg}(X) \leftarrow . \quad (2)$$

where “!” is the cut operator and `fail` is a Prolog built-in with the empty definition.

The intuition behind this definition is perhaps best revealed by first introducing the `if_then_else` predicate defined as follows:

$$\text{if\_then\_else}(P, Q, R) \leftarrow P,!,Q.$$

$$\text{if\_then\_else}(P, Q, R) \leftarrow R.$$

`if_then_else` is intended to model within Prolog the customary

$$\text{if } P \text{ then } Q \text{ else } R$$

construct of imperative programming languages. Then `neg` can be equivalently defined by

$$\text{neg}(X) \leftarrow \text{if\_then\_else}(X, \text{fail}, \square).$$

where  $\square$  is the empty query which immediately succeeds. So intuitively,  $\text{neg}(X)$  can be interpreted as “if  $X$  succeeds then fail else succeed”.

It is usually tacitly assumed that logic programming and Prolog ways of dealing with negation are “equivalent”, in the sense that SLDNF-resolution combined with the leftmost selection rule (henceforth called LDNF-resolution) properly reflects Prolog’s way of handling negation. Upon closer scrutiny this assumption is far from being obvious. The above definition of the  $\text{neg}$  predicate and its use in programs calls upon a number of features which are present in Prolog, but absent in logic programming, and for which a formal treatment is lacking. These are:

- the use of meta-variables, that is variables which occur in an atom position, like  $X$  in the first clause,
- the use of meta-programming facilities that arise when applying this definition of  $\text{neg}$ , so in constructs of the form  $\text{neg}(A)$  where  $A$  is an atom, or a query in general.

Additionally, two better understood, though not necessarily simpler to handle, features of Prolog need to be taken care of, namely:

- the ordering of the program clauses,
- the use of the cut operator “!”.

The aim of this paper is to relate precisely these two uses of negation: in logic programming and in Prolog. To do this we appropriately tune the definition of the SLDNF-resolution given in Apt and Doets [AD94] to our present needs and formally define “Prolog trees” in the presence of the cut operator. Then we prove a result that shows an appropriate equivalence between these two definitions of negation.

The outcome of this study is that we can now interpret various results about correctness of general logic programs executed by means of the LDNF-resolution (see e.g. Apt [Apt94]) as correctness results about the corresponding Prolog programs that use negation.

## 2 SYNTACTIC MATTERS

### 2.1 General Logic Programs

To relate general logic programs to Prolog programs we have to be precise about the syntax. Fix a first-order language  $\mathcal{L}$ . To make this comparison possible we assume that

- a general program is a *sequence* and not a *set* of general clauses,
- the predicates  $!$ ,  $\text{neg}$  and  $\text{fail}$  are not present in the language  $\mathcal{L}$ .

A *general clause* is defined in the usual way (see e.g. Lloyd [Llo87]), so as a construct of the form  $A \leftarrow L_1, \dots, L_n$ , where  $A$  is an atom and  $L_1, \dots, L_n$  are literals, i.e. atoms or their negations, all in the language  $\mathcal{L}$ . And a *query* is a finite sequence of literals. In the context of logic programming the negation connective is written as “ $\neg$ ”.

## 2.2 Prolog Programs

Prolog programs here considered are intended to be the programs that allow us to model the negation by means of the predicate **neg** defined by the clauses (1) and (2). However, the syntax of clause (1) creates a number of problems, even if we ignore the cut operator “! ”.

First of all, the use of the meta-variable  $\mathbf{X}$  in clause (1) violates the syntax of the first-order logic. This use of  $\mathbf{X}$  in the resolution process leads to further complications. Take an  $n$ -ary function symbol  $\mathbf{p}$  in the language  $\mathcal{L}$  and let  $s_1, \dots, s_n$  be some terms. Consider now the query  $\mathbf{neg}(\mathbf{p}(s_1, \dots, s_n))$ . During Prolog computation process it resolves using the clause (1) to the query  $\mathbf{p}(s_1, \dots, s_n), !, \mathbf{fail}$ . Now in the first query  $\mathbf{p}$  occurs in a position of a function symbol, whereas in the second one  $\mathbf{p}$  occurs in a position of a relation symbol. So every function symbol needs also to be accepted as a relation symbol.

Also conversely: take an  $n$ -ary relation symbol  $p$  with some terms  $s_1, \dots, s_n$ , and consider the general clause  $p(s_1, \dots, s_n) \leftarrow \neg p(s_1, \dots, s_n)$ . Its desired translation into a Prolog clause is  $\mathbf{p}(s_1, \dots, s_n) \leftarrow \mathbf{neg}(\mathbf{p}(s_1, \dots, s_n))$ . In the head of the latter clause  $\mathbf{p}$  occurs in a position of a relation symbol, whereas in its body in the position of a function symbol.

As in both cases  $\mathbf{p}$  was arbitrarily chosen, we conclude that to render the resolution process meaningful we need to accept that the classes of function symbols and of relation symbols in the underlying language coincide.

This is clearly in violation with the (usually tacit) assumption that in the first-order language, say  $\mathcal{L}$ , fixed above, the classes  $F_m$  and  $R_n$  of, respectively, its function symbols of arity  $m$  and its relation symbols of arity  $n$  are pairwise disjoint for  $m, n \geq 0$ . In short, the use of the clause (1) cannot be properly accounted for by just referring to the first-order logic.

A simple solution to the above mentioned two problems is to modify the syntax of the language  $\mathcal{L}$  by allowing

- *meta-variables*, so variables that can occur in atoms positions, both in the queries and in the clause bodies,
- *ambivalent syntax*, so – in this case – by assuming that the classes of function and relation symbols coincide.

The latter can be achieved by extending  $\mathcal{L}$  to a language in which for each  $m \geq 0$   $F_m \cup R_m$  are the classes of both its function symbols and relation symbols. Thus in this language terms and atoms coincide.

Additionally, we assume that

- the predicates **!**, **neg** and **fail** are present in the underlying language,
- **!** is a built-in 0-ary predicate (with a meaning to be explained later), and no clause uses it in its head,
- **neg** is a built-in predicate defined by the clauses (1) and (2), so no other clause uses it in its head,
- **fail** is a built-in 0-ary predicate with the empty definition, so no clause uses it in its head.

The last two assumptions ensure that **neg** and **fail** are indeed defined internally in the desired way. For the purposes of syntax the cut operator “!” is viewed here as a 0-ary predicate with the empty definition. This might suggest that its meaning coincides with that of **fail**. However, this is not the case. Its real, operational, “meaning” will be defined in Section 4 by means external to the resolution process.

So in the resulting language, apart of the customary atoms, also **!**, **fail** and meta-variables are admitted as atoms (henceforth called *special atoms*).

Now, a *Prolog program* is defined as a sequence of Prolog clauses preceded by the clauses (1) and (2). In turn a *Prolog clause* is a construct of the form  $A \leftarrow B_1, \dots, B_n$ , where  $A, B_1, \dots, B_n$  are atoms in the language  $\mathcal{L}$ , and  $A$  is not a special atom. And a *Prolog query* is a finite sequence of atoms. For brevity, in the examples of Prolog programs, we drop the listing of the clauses (1) and (2). Finally, we denote sequences of atoms or literals by bold capital letters.

Note that at this stage we use two notions of an atom – one within the language  $\mathcal{L}$  and another in its ambivalent extension just defined. From the context it will be always clear to which of these two languages we refer.

### 2.3 Restricted Prolog Programs

The translation of a general program to a Prolog program is now straightforward and as expected: we just replace everywhere a logic programming literal  $\neg A$  by Prolog’s atom **neg**( $A$ ) and prefix the resulting program with the clauses (1) and (2). In short, the logic programming negation connective “ $\neg$ ” is traded for the built-in predicate **neg**. Similarly, a general query is translated to a Prolog query by replacing everywhere  $\neg A$  by **neg**( $A$ ).

This translation process maps every general program (resp. general query) onto a Prolog program. However, not every Prolog program (resp. Prolog query) is the result of translating a general program (resp. general query). Indeed, in general the cut operator “!” can be used in any Prolog clause, not only (1).

Let us now characterize the Prolog programs (resp. Prolog queries) which are the result of the above translation of general programs (resp. general queries). We call them *restricted Prolog programs* (resp. *restricted Prolog queries*). To this we translate “back” every Prolog program (resp. Prolog query) onto a general program (resp. general query) by replacing everywhere **neg**( $A$ ) by  $\neg A$ ,

and omitting the clauses (1) and (2) that define the `neg` predicate. Then a Prolog program (resp. Prolog query) is restricted if the outcome of this reverse translation is a syntactically legal general program (resp. general query). For example the Prolog query `neg(q),q` is restricted because its reverse translation is  $\neg q, q$ , whereas neither `neg(q(neg(a)))` nor `p(q),q` is restricted because their respective reverse translations violate the syntactic assumptions concerning general programs.

Of course, it is possible to define the class of restricted Prolog programs and queries directly, though the resulting definition is rather tedious.

We now define a *resolvent* of a Prolog query as follows.

**DEFINITION 2.1** Consider a non-empty Prolog query  $A, \mathbf{M}$  and a Prolog clause  $c$ . Let  $H \leftarrow \mathbf{L}$  be a variant of  $c$  variable disjoint with  $A, \mathbf{M}$  and let  $\theta$  be an mgu of  $A$  and  $H$ . Then  $(\mathbf{L}, \mathbf{M})\theta$  is called a *resolvent* of  $A, \mathbf{M}$  and  $c$  with an mgu  $\theta$ .  $\square$

The only unusual feature in the present setting is, that now the mgu's also bind the meta-variables. Also, note that the selected literal is always the leftmost literal.

It is worthwhile to mention that a resolvent of a restricted Prolog query w.r.t. a restricted Prolog program is not necessarily a restricted Prolog query. This is due to the use of clause (1), which introduces a cut atom. Thus, the Prolog queries generated in a computation of a restricted Prolog query are not necessarily restricted Prolog queries. However, the Prolog queries so generated do have one important property: they do not contain meta-variables. To prove this fact we need a stronger property.

**DEFINITION 2.2**

- An atom  $A$  is called *unsafe* if one of the following holds:
  - $A$  is a meta-variable,
  - $A$  is `neg(X)` where  $X$  is a variable,
  - $A$  is `neg(neg(s))` where  $s$  is a term.
- A Prolog query is called *meta-safe* if none of its atoms is unsafe.  $\square$

For example,  $X$ , `p(X)` is not meta-safe because its leftmost atom is a meta-variable, `neg(X)` is not meta-safe because the argument of `neg` is a meta-variable, and `neg(neg(p(X)))` is not meta-safe because the argument of the outermost `neg` predicate is itself a `neg` predicate.

Note that restricted Prolog queries and bodies of the restricted Prolog clauses are meta-safe.

**LEMMA 2.3** *Let  $Q$  be a meta-safe Prolog query and  $P$  a restricted Prolog program. Then all resolvents of  $Q$  are meta-safe.*

**Proof:** Let  $Q$  be of the form  $A, \mathbf{L}$ , and let  $(\mathbf{M}, \mathbf{L})\theta$  be a resolvent of  $Q$ , with

an input clause  $c$  and mgu  $\theta$ . As  $Q$  is meta-safe, we know that  $L\theta$  is meta-safe. We prove that  $M\theta$  is meta-safe as well. Three cases arise.

**Case 1** :  $c$  is clause (1).

Then  $M\theta$  is of the form  $B,!,fail$ , where  $A$  is of the form  $\text{neg}(B)$ . But  $Q$  is meta-safe, so  $B$  is neither a meta-variable nor of the form  $\text{neg}(B')$ . So  $M\theta$  is meta-safe.

**Case 2** :  $c$  is clause (2).

Then  $M\theta$  is the empty query, so obviously meta-safe.

**Case 3** :  $c$  is different from clauses (1) and (2).

Then the body of  $c$  is meta-safe, and consequently so is  $M\theta$ .

This proves that  $(M, L)\theta$  is meta-safe. □

**COROLLARY 2.4** *All Prolog queries generated in a computation of a restricted Prolog query and a restricted Prolog program are meta-safe.* □

In Prolog, if the selected atom is a meta-variable, an *error* arises. The above result thus shows that no errors arise in Prolog computations for queries and programs that are obtained by a translation of a general query and a general program.

### 3 COMPUTING WITH GENERAL LOGIC PROGRAMS: LDNF-RESOLUTION

As the next step we define the LDNF-resolution that allows us to compute with general logic programs. The definition of LDNF-resolution given here is derived in a straightforward way from that of the SLDNF-resolution given in Apt and Doets [AD94]. Apart of the fact that we view in this paper a general program as a finite sequence and not as a finite set of general clauses, the differences are that:

- the leftmost selection rule is used,
- *floundering*, so –in this context– an abnormal termination due to selection of a non-ground literal is ignored.

In this way we bring the procedural interpretation of general programs closer to that of the corresponding Prolog programs and make the subsequent comparison possible. Recall from Clark [Cla78] and Lloyd [Llo87] that floundering is a problem that arises only when dealing with the semantic aspects of the SLDNF-resolution, which are irrelevant here.

Before giving the definition of LDNF-resolution, we recall the definitions of *resolvent* and *pseudo-derivation*.

**DEFINITION 3.1** Consider a non-empty general query  $L, M$  and a general clause  $c$ .



- Suppose  $L$  is a positive literal.

Let  $H \leftarrow \mathbf{L}$  be a variant of  $c$  variable disjoint with  $L, \mathbf{M}$  and let  $\theta$  be an mgu of  $L$  and  $H$ . Then  $(\mathbf{L}, \mathbf{M})\theta$  is called a *resolvent* of  $L, \mathbf{M}$  and  $c$  w.r.t.  $L$ , with an mgu  $\theta$ .

We write then  $L, \mathbf{M} \xrightarrow[c]{\theta} (\mathbf{L}, \mathbf{M})\theta$ , and call it a *positive derivation step*. We call  $H \leftarrow \mathbf{L}$  the *input clause* of the derivation step.

- Suppose  $L$  is a negative literal. Then  $\mathbf{M}$  is called a *resolvent* of  $L, \mathbf{M}$  with the identity substitution  $\epsilon$  w.r.t.  $L$ .

We write then  $L, \mathbf{M} \xrightarrow[\emptyset]{\epsilon} \mathbf{M}$ , and call it a *negative derivation step*.

- A general clause  $c$  is called *applicable* to an atom if it has a variant the head of which unifies with the atom.  $\square$

Fix, until the end of this section, a general program  $P$ .

DEFINITION 3.2 A (finite or infinite) sequence  $Q_0 \xrightarrow[c_1]{\theta_1} Q_1 \cdots Q_n \xrightarrow[c_{n+1}]{\theta_{n+1}} Q_{n+1} \cdots$  of derivation steps is called a *pseudo derivation of  $P \cup \{Q_0\}$*  if

- $Q_0, \dots, Q_n, \dots$  are general queries,
- $\theta_1, \dots, \theta_n, \dots$  are substitutions,
- $c_1, \dots, c_n, \dots$  are general clauses of  $P$ , or  $\emptyset$ ,

and for every step involving selection of a positive literal the following condition holds:

**Standardization apart:** the input clause employed is variable disjoint from the initial general query  $Q_0$  and from the substitutions and input clauses used at earlier steps.  $\square$

Intuitively, an LDNF-derivation is a pseudo derivation in which the deletion of every negative literal is justified by means of a subsidiary (finitely failed LDNF-) tree. This brings us to consider special types of trees, called *forests*.

DEFINITION 3.3 A *forest* is a system  $\mathcal{F} = (\mathcal{F}, T, \text{subs})$  where

- $\mathcal{F}$  is a set of trees,
- $T$  is an element of  $\mathcal{F}$  called the *main tree*, and
- $\text{subs}$  is a function assigning to some nodes of trees in  $\mathcal{F}$  a (“subsidiary”) tree from  $\mathcal{F}$ .

By a *path* in  $\mathcal{F}$  we mean a sequence of nodes  $N_0, \dots, N_i, \dots$  such that for all  $i$ ,  $N_{i+1}$  is either an immediate descendant of  $N_i$  in some tree in  $\mathcal{F}$ , or the root of the tree  $\text{subs}(N_i)$ . The *depth* of  $\mathcal{F}$  is the length of the longest path in  $\mathcal{F}$ .  $\square$

Thus a forest is a special directed graph with two types of edges – the “usual” ones stemming from the tree structures, and the ones connecting a node with the root of a subsidiary tree. An LDNF-tree is a special type of forest, built as a limit of certain finite forests: *pre-LDNF trees*.

DEFINITION 3.4 A *pre-LDNF-tree* (relative to  $P$ ) is a forest whose nodes are queries. Leaves can be unmarked, or can be marked as either *success* or *failure*. The class of pre-LDNF-trees is defined inductively:

- For every general query  $Q$ , the forest consisting of the main tree which has the single unmarked node  $Q$  is a pre-LDNF-tree (an *initial* pre-LDNF-tree),
- If  $\mathcal{T}$  is a pre-LDNF-tree, then any *extension* of  $\mathcal{T}$  is a pre-LDNF-tree.

Before defining the notion of an *extension* of a pre-LDNF-tree, we need to define the notion of *successful* and *finitely failed* trees: for  $T \in \mathcal{T}$ ,

- $T$  is called *successful*, if one of its leaves is marked as *success*, and
- $T$  is called *finitely failed*, if it is finite and all its leaves are marked as *failure*.

Now, an *extension* of a pre-LDNF-tree  $\mathcal{T}$  is defined by performing the following actions for every non-empty general query  $Q$  (with leftmost literal  $L$ ) which is an unmarked leaf in some tree  $T \in \mathcal{T}$ :

- Suppose that  $L$  is a positive literal.
  - If  $Q$  has no resolvents w.r.t.  $L$  and a clause from  $P$ :  
Mark  $Q$  as *failure*.
  - If  $Q$  has such resolvents:  
For every clause  $c$  from  $P$  which is applicable to  $L$ , choose one resolvent  $Q'$  of  $Q$  w.r.t.  $L$  and  $c$ , with an mgu  $\theta$ , and add this as an immediate descendant of  $Q$  in  $T$ . Choose the input clauses in such a way that all branches of  $T$  remain pseudo derivations.
- Suppose that  $L$  is a negative literal, say  $\neg A$ .
  - If  $\text{subs}(Q)$  is undefined:  
Add a new tree  $T'$ , consisting of the single node  $A$ , to  $\mathcal{T}$ , and let  $\text{subs}(Q) = T'$ .
  - If  $\text{subs}(Q)$  is defined and successful:  
Mark  $Q$  as *failure*.
  - If  $\text{subs}(Q)$  is defined and finitely failed:  
Add the resolvent  $Q - \{L\}$  of  $Q$  as the only immediate descendant of  $Q$  in  $T$ .

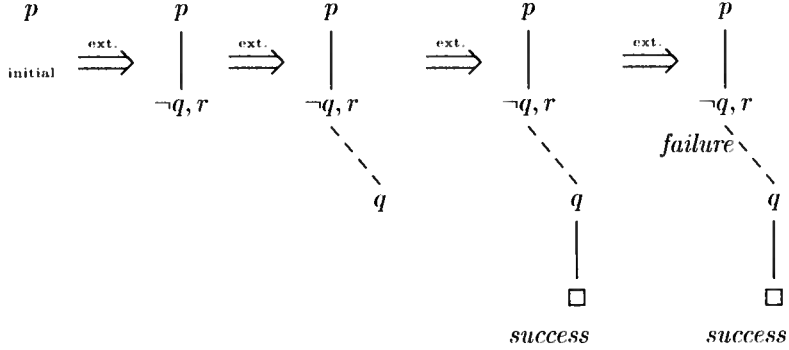


FIGURE 1. Step-by-step construction of an LDNF-tree for the query  $p$  w.r.t. the general program  $p \leftarrow \neg q, r \quad q \leftarrow$ .

Additionally, all empty queries are marked as *success*.  $\square$

Note that, if no tree in  $\mathcal{T}$  has unmarked leaves, then trivially  $\mathcal{T}$  is an extension of itself, and the extension process becomes stationary.

Next, we define LDNF-trees as the limit of sequences of pre-LDNF-trees. Every pre-LDNF-tree is a tree with two types of edges between possibly marked nodes, so the concepts of *inclusion* between such trees and of *limit* of a growing sequence of such trees have a clear meaning.

DEFINITION 3.5

- An *LDNF-tree* is a limit of a sequence  $\mathcal{T}_0, \dots, \mathcal{T}_\alpha, \dots$  such that  $\mathcal{T}_0$  is an initial pre-LDNF-tree, and for all  $i$   $\mathcal{T}_{i+1}$  is an extension of  $\mathcal{T}_i$ .
- An *LDNF-tree for  $Q$*  is an LDNF-tree in which  $Q$  is the root of the main tree.
- A (pre-)LDNF-tree is called *successful* (resp. *finitely failed*) if the main tree is successful (resp. finitely failed).
- An LDNF-tree is called *finite* if no infinite path exists in it (cf. Definition 3.3).  $\square$

In Figure 1, we show how the notions of initial pre-LDNF-trees and extensions of pre-LDNF-trees are used to construct a P-tree.

Finally, we recall the notion of a computed answer substitution.

DEFINITION 3.6 Consider a branch in the main tree of a (pre-)LDNF-tree for  $Q$  which ends with the empty query. Let  $\alpha_1, \dots, \alpha_n$  be the consecutive substitutions along this branch.

Then the restriction  $(\alpha_1 \cdots \alpha_n)|Q$  of the composition  $\alpha_1 \cdots \alpha_n$  to the variables of  $Q$  is called a *computed answer substitution* (*c.a.s.* for short) of  $Q$ .

$\square$

#### 4 COMPUTING WITH PROLOG PROGRAMS: P-RESOLUTION

In this section, we define the computation process used in Prolog to find answers to queries, which we call *P-resolution*. To this end we proceed in two steps.

First, we restrict the LDNF-resolution to logic programs, so general logic programs without negation, by simply disregarding the selection of a negative literal. We call the resulting computation process *LD-resolution*.

Then, we extend the LD-resolution to Prolog programs by allowing the choice of a meta-variable or of a cut atom as a selected atom. In the first case an error is reported, and in the second case the computation tree constructed so far is appropriately pruned.

In Prolog, answers are computed using a left to right depth-first strategy. In particular, Prolog processes the cut atoms in the tree from left to right. On the other hand, LD-resolution is defined in a breadth-first manner: the process of extending a pre-tree consists of extending all unmarked leaves of that tree simultaneously. To solve this problem, we have to refine LD-resolution so that the depth-first strategy is used instead of the breadth-first strategy. At first sight it seems that to this end we have to implement the backtracking mechanism used by Prolog. Fortunately, it is not so. A simpler alternative is to generate at each stage all direct successors of the *leftmost* unmarked leaf only. In this way the backtracking process is taken care of automatically.

Having discussed the modifications of the LD-resolution we now model the computation process of Prolog, by providing a formal definition of P-resolution. The central notion in this definition is that of a *P-tree*. We define them as the limit of a sequence of *pre-P-trees*, which in turn are a subclass of a class of ordered trees called *semi-P-trees*.

**DEFINITION 4.1** A *semi-P-tree* (relative to  $P$ ) is an ordered tree whose nodes contain queries, possibly marked with *success*, *failure*, or *error*.  $\square$

The first step in defining pre-P-trees is to define the effect of the cut operator.

**DEFINITION 4.2** Let  $\mathcal{B}$  be a branch in a semi-P-tree, and let  $Q$  be a node in this branch with a cut atom as the leftmost atom. Then, the *origin* of this cut atom is the first predecessor of  $Q$  in  $\mathcal{B}$  that contains less cut atoms than  $Q$ .  $\square$

To see that this definition properly captures the informal meaning of the origin note that, when following a branch from top to bottom, the cut atoms are introduced and removed in a First-In Last-Out manner.

**DEFINITION 4.3** Let  $\mathcal{T}$  be a semi-P-tree,  $Q$  a query in  $\mathcal{T}$  which has a cut atom as the leftmost atom, and  $Q'$  be the origin of this cut atom. Then, the operator  $cut(\mathcal{T}, Q)$  removes from  $\mathcal{T}$  all the nodes that are descendants of  $Q'$  and lie to the right of  $Q$ .  $\square$

In Figure 2, we illustrate the effect of  $cut(\mathcal{T}, Q)$ .

**DEFINITION 4.4** The class of *pre-P-trees* is defined as follows:

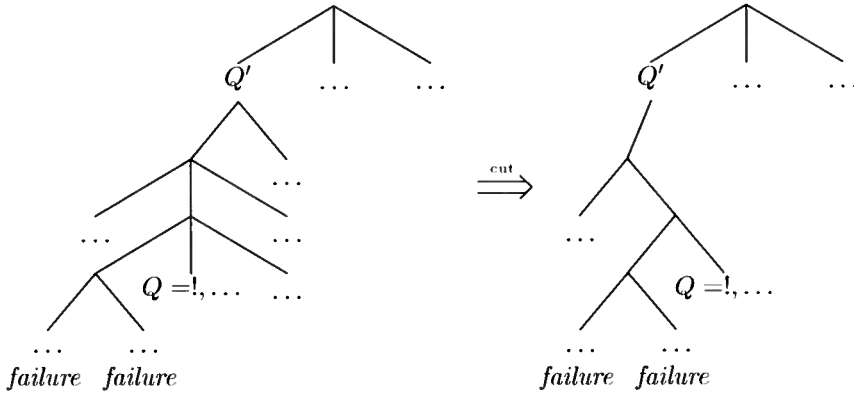


FIGURE 2. The effect of the operator  $cut(T, Q)$

- For every query  $Q$ , the tree consisting of the single unmarked node  $Q$  is a pre-P-tree (an *initial* pre-P-tree).
- If  $T$  is a pre-P-tree, then any *extension* of  $T$  is a pre-P-tree.

An *extension* of a pre-P-tree  $T$  is defined as follows:

Let  $Q$  be the leftmost unmarked leaf in  $T$ . If  $Q$  is the empty query, mark  $Q$  as *successful*. Otherwise, let  $Q$  be of the form  $A, M$ .

- Suppose  $A$  is an ordinary atom (i.e. not a special atom).
  - If  $Q$  has no resolvents w.r.t. a clause from  $P$ :  
Mark  $Q$  as *failure*.
  - If  $Q$  has such resolvents:  
For every clause  $c$  from  $P$  which are applicable to  $A$ , choose one resolvent  $Q'$  of  $Q$  w.r.t.  $c$  and add this as a child of  $Q$  in  $T$ . Choose the input clauses in such a way that all branches of  $T$  remain pseudo derivations. Order these children according to the the order in which their input-clauses appear in  $P$ .
- Suppose  $A$  is a cut atom.  
Apply the operation  $cut(T, Q)$ .  
Provide  $Q$  with a single child  $M$ .
- Suppose  $A$  is a meta-variable.  
Mark  $Q$  as *error*. □

We now define P-trees as the limit of sequences of pre-P-trees. In Figure 3, we show how the notions of initial pre-P-trees and extensions of pre-P-trees can be used to construct a P-tree (the program used in the figure is the translation of

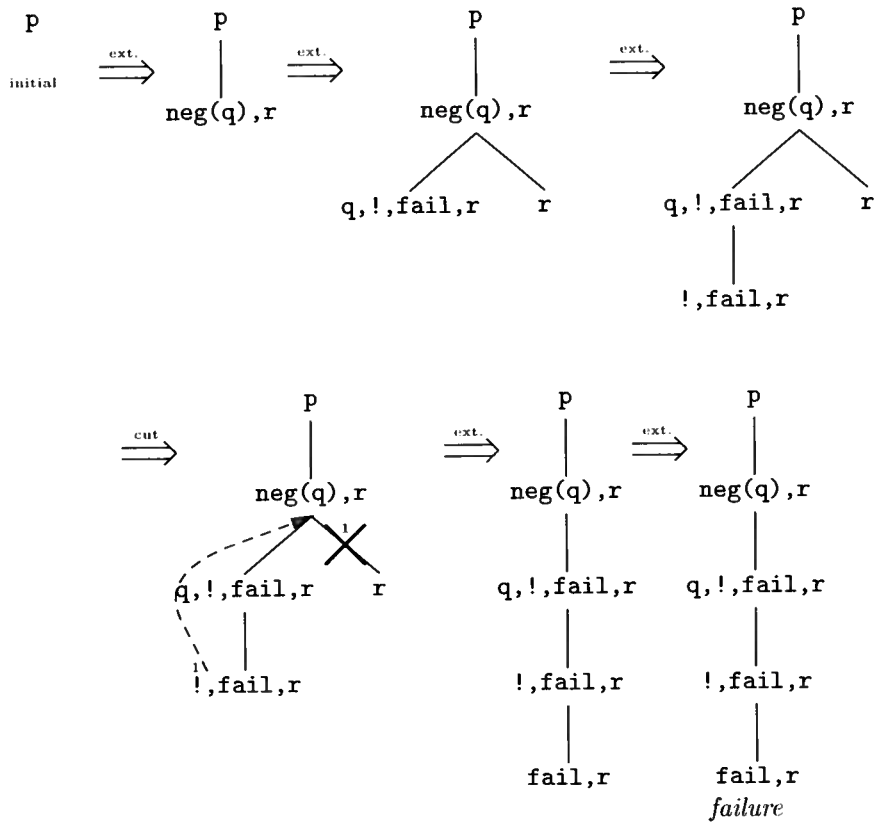


FIGURE 3. Step-by-step construction of a P-tree for the Prolog query  $p$  w.r.t. the Prolog program  $p \leftarrow \text{neg}(q), r. \quad q \leftarrow \dots$



the program used in Figure 1). Note that in this Figure, the result of the ‘cut step’ (that is, the fifth tree) is not itself part of the sequence of extensions; it was added to clarify the use of the cut operator in the construction of P-trees.

To be able to define the limit of a sequence of pre-P-trees, we have to define a notion of an *inclusion* between pre-P-trees, and of the *limit* of a growing sequence of pre-P-trees. For pre-LD-trees and pre-LDNF-trees, these notions were obvious. In the case of pre-P-trees, the pruning that takes place when extending a pre-P-tree, complicates the matters a bit.

**DEFINITION 4.5** Let  $\mathcal{T}$  and  $\mathcal{T}'$  be pre-P-trees.  $\mathcal{T}$  is said to be *included* in  $\mathcal{T}'$  if  $\mathcal{T}'$  can be constructed from  $\mathcal{T}$  by means of one of the following two operations:

1. adding some children to a leaf of  $\mathcal{T}$ .
2. removing a single subtree from  $\mathcal{T}$ , provided its root is not a single child in  $\mathcal{T}$ .

We say that  $\mathcal{T}$  is *properly included* in  $\mathcal{T}'$ , if  $\mathcal{T}$  is included in  $\mathcal{T}'$  and  $\mathcal{T}'$  is not included in  $\mathcal{T}$ . We use  $\subset$  to denote the transitive closure of the relation “ $\mathcal{T}$  is properly included in  $\mathcal{T}'$ ” and define  $\mathcal{T} \subseteq \mathcal{T}'$  as  $(\mathcal{T} \subset \mathcal{T}') \vee (\mathcal{T} = \mathcal{T}')$ .  $\square$

Note that operation (2) never turns an internal node into a leaf.

**LEMMA 4.6** *The relation  $\subset$  is a strict partial order on pre-P-trees.*

**Proof:** We have to prove that the conditions for a strict partial order hold.

1.  $\mathcal{T} \not\subseteq \mathcal{T}$

Suppose by contradiction that  $\mathcal{T} \subset \mathcal{T}$ . Then, there exists a  $\mathcal{T}'$  such that  $\mathcal{T}$  is properly included in  $\mathcal{T}'$ , and  $\mathcal{T}' \subseteq \mathcal{T}$ . There are two cases:

- $\mathcal{T}'$  is constructed by adding children to a leaf of  $\mathcal{T}$ .  
But then, some node  $Q$  that is a leaf in  $\mathcal{T}$ , is an internal node in  $\mathcal{T}'$ . By definition of inclusion, and the fact that  $\mathcal{T}' \subseteq \mathcal{T}$ ,  $Q$  is an internal node in  $\mathcal{T}$ . This is in contradiction with the fact that  $Q$  is a leaf  $\mathcal{T}$ .
- $\mathcal{T}'$  is constructed by pruning a single subtree from  $\mathcal{T}$ .  
By definition of inclusion, the parent of the pruned subtree has at least two children in  $\mathcal{T}$ , and therefore, it has at least one child in  $\mathcal{T}'$ . Moreover, new nodes can only “grow” from leaves. Thus subtrees pruned from  $\mathcal{T}$  can never be “regenerated”, to reconstruct  $\mathcal{T}$  out of  $\mathcal{T}'$ . Therefore,  $\mathcal{T}' \not\subseteq \mathcal{T}$ , which leads to a contradiction.

2.  $\mathcal{T} \subset \mathcal{T}'$  and  $\mathcal{T}' \subset \mathcal{T}''$  imply  $\mathcal{T} \subset \mathcal{T}''$ .

Straightforward by the definition of  $\subset$ .  $\square$

**COROLLARY 4.7** *The relation  $\subseteq$  is a partial order on pre-P-trees.*  $\square$

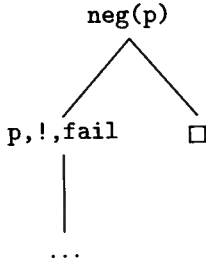


FIGURE 4. A P-tree for the query  $\text{neg}(p)$  w.r.t.  $p \leftarrow p$ .

Clearly, with this notion of inclusion, we have that if  $\mathcal{T}$  extends  $\mathcal{T}'$  in the sense of Definition 4.4, then  $\mathcal{T}' \subseteq \mathcal{T}$ , so we can use this notion of extension to construct monotonously growing chains of pre-P-trees.

DEFINITION 4.8

- A *P-tree* is a limit of a sequence  $\mathcal{T}_0, \dots, \mathcal{T}_i, \dots$  such that  $\mathcal{T}_0$  is an initial pre-P-tree, and for all  $i$ ,  $\mathcal{T}_{i+1}$  is an extension of  $\mathcal{T}_i$ .
- A *P-tree for Q* is a P-tree whose root is the query  $Q$ .
- An P-tree is called *finite* if no infinite branch exists in it. □

Formally, this definition is justified by the fact that every countable partial order with the least element (here the relation  $\subseteq$  on pre-P-trees with the initial pre-P-tree as least element) can be canonically extended to a countable cpo (see e.g. Gierz [GHK<sup>+</sup>80]).

Next, we define the concepts of *successful* and *finitely failed* P-trees.

DEFINITION 4.9

- A P-tree is called *successful* if one of its leaves is marked as *success*.
- A (pre-)P-tree is called *finitely failed*, if it is finite, and all its leaves are marked as *failure*. □

Note that in P-trees, in contrast to LDNF-trees, some leaves can be unmarked. Whenever this is the case, the P-tree will contain exactly one infinite branch to the left of all these unmarked leaves. Such unmarked leaves represent the resolvents the Prolog computation process did not reach, because it got “trapped” in an infinite derivation (the infinite branch). For example, take the program  $p \leftarrow p.$ , and the query  $\text{neg}(p)$ . Its P-tree is shown in Figure 4. This tree contains a branch ending with a leaf containing the empty query. However, this leaf is never reached by the Prolog computation process (and therefore never marked) because there is an infinite branch to the left of it.

Finally, it is clear how to define the notion of a computed answer substitution.

DEFINITION 4.10 Consider a successful derivation in a pre-P-tree for  $Q$ . Let  $\alpha_1, \dots, \alpha_n$  be the consecutive substitutions along this branch.

Then the restriction  $(\alpha_1 \cdots \alpha_n)|Q$  of the composition  $\alpha_1 \cdots \alpha_n$  to the variables of  $Q$  is called a *computed answer substitution* (c.a.s. for short) of  $Q$ .

□

## 5 CORRESPONDENCE BETWEEN LDNF-TREES AND P-TREES

In this section, we prove that there is a close correspondence between (computed answers of) LDNF-trees and P-trees. More precisely, we prove that termination results on general programs w.r.t. LDNF-resolution translate directly into termination of their translated Prolog programs w.r.t. Prolog computation. For this purpose, we start by examining finite LDNF-trees, and their corresponding P-trees.

THEOREM 5.1 *Let  $T_L$  be a finite LDNF-tree for a general query  $Q$ . Then, there exists a finite P-tree  $T_P$  for  $Q$  such that  $T_L$  and  $T_P$  have the same set of computed answers.*

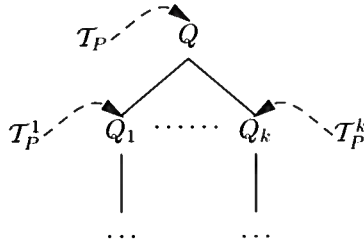
**Proof:** We prove the claim by induction on the depth of LDNF-trees (cf. Definition 3.3). Assume that the claim holds for all LDNF-trees of depth less than  $r$ . We have to prove the claim for LDNF-trees of depth  $r$ .

Let  $T_L$  be an LDNF-tree for  $Q$  of some finite depth  $r$ . In the remainder of this proof, we identify a general query with its translation into a Prolog query. From the context it will always be clear whether we refer to a general query, or a Prolog query. Two cases arise.

- Suppose that  $Q$  is of the form  $A, L$ .

Let  $Q_1, \dots, Q_k$  ( $k \geq 0$ ) be the children of  $Q$  in  $T_L$ . Let, for  $i \in [1..k]$ ,  $T_L^i$  denote the subtree of  $T_L$  starting at  $Q_i$ .

As, for  $i \in [1..k]$ ,  $T_L^i$  is finite and of depth less than  $r$ , by induction hypothesis there exists a P-tree  $T_P^i$  for  $Q_i$  such that  $T_P^i$  contains the same computed answers as  $T_L^i$ . Now consider the semi-P-tree  $T_P$  with root  $Q$ , children  $Q_1, \dots, Q_k$  (ordered according to the order of their input clauses in  $P$ ) and, for  $i \in [1..k]$ ,  $T_P^i$  as the subtree starting at  $Q_i$ , as depicted by the following diagram:



To prove that  $\mathcal{T}_P$  is a P-tree for  $Q$ , it is sufficient to show that all pruning caused by selection of cut atoms is guaranteed to be local to the respective subtrees  $\mathcal{T}_P^i$  (for  $i \in [1..k]$ ). Neither  $Q$ , nor its children  $Q_1, \dots, Q_k$  in  $\mathcal{T}_P$ , contain a cut atom, so no atom in  $\mathcal{T}_P$  has  $Q$  as its origin. It follows from the definition of the cut operator that all pruning is indeed local to the respective subtrees  $\mathcal{T}_P^i$ . Thus  $\mathcal{T}_P$  is a P-tree for  $Q$ . From its construction, it follows that it contains the same computed answers as  $\mathcal{T}_L$ . Moreover, it is finite.

- Suppose that  $Q$  is of the form  $\neg A, \mathbf{L}$ .

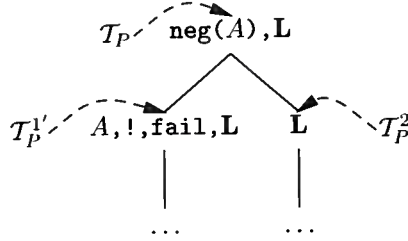
Let  $\mathcal{T}_L^1$  be the subtree of  $\mathcal{T}_L$  starting at the root of  $\text{subs}(Q)$ . As the LDNF-tree  $\mathcal{T}_L^1$  for  $A$  is finite and of depth less than  $r$ , by induction hypothesis there exists a finite P-tree  $\mathcal{T}_P^1$  for  $A$  that has the same computed answers as  $\mathcal{T}_L^1$ . There are two sub-cases.

- Suppose that  $Q$  has a child in  $\mathcal{T}_L$ .

Then,  $\mathcal{T}_L^1$  is finitely failed, and therefore  $\mathcal{T}_P^1$  is finitely failed as well. But then, we can construct a finitely failed P-tree  $\mathcal{T}_P^{1'}$  for  $A, \text{!,fail}, \mathbf{L}$ . In this P-tree, the cut atom introduced at the root will never be reached.

Let  $\mathcal{T}_L^2$  be the subtree of  $\mathcal{T}_L$  starting at the single child  $\mathbf{L}$  of  $Q$ . As the LDNF-tree  $\mathcal{T}_L^2$  for  $\mathbf{L}$  is finite and of depth less than  $r$ , by induction hypothesis there exists a finite P-tree  $\mathcal{T}_P^2$  for  $\mathbf{L}$  that has the same computed answers as  $\mathcal{T}_L^2$ .

Using  $\mathcal{T}_P^{1'}$  and  $\mathcal{T}_P^2$  we can construct a finite P-tree  $\mathcal{T}_P$  for  $Q$  that has the same computed answers as  $\mathcal{T}_L$ . This tree has the following form:



- Suppose that  $Q$  has no children in  $\mathcal{T}_L$ .

Then,  $\mathcal{T}_L^1$  is successful, and therefore  $\mathcal{T}_P^1$  is successful as well. But then we can construct a finitely failed P-tree  $\mathcal{T}_P^{1'}$  for  $A, \text{!,fail}, \mathbf{L}$ , in which the cut atom present in its root is selected at some point.

Let  $\mathcal{T}_P$  be the semi-P-tree such that its root is  $Q$ , and the subtree starting at the single child  $A, \text{!,fail}, \mathbf{L}$  of  $Q$  is  $\mathcal{T}_P^{1'}$ . In this tree, the origin of the cut atom that appears in the single child of  $Q$ , is  $Q$ . This cut atom is the selected atom in some node within  $\mathcal{T}_P^{1'}$ . Thus  $\mathcal{T}_P$  is a P-tree for  $Q$ , because the potential second child of  $Q$ , that

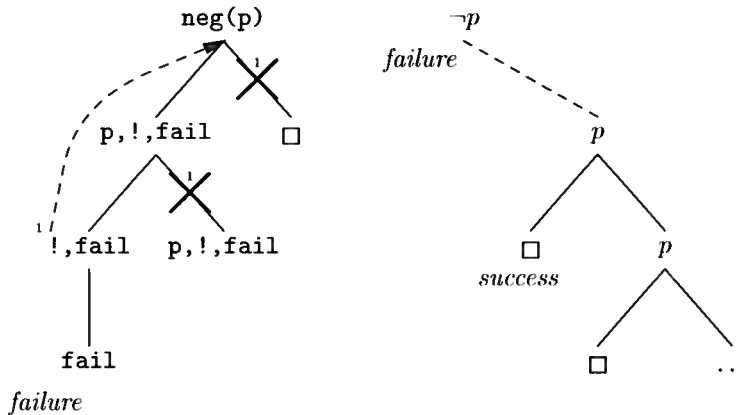


FIGURE 5. A P-tree and an LDNF-tree for  $\text{neg}(p)$

would contain the query  $L$  has been pruned at some stage. Thus  $T_P$  is finitely failed, just as  $T_L$  is.  $\square$

Thus if we have a general query  $Q$  that terminates w.r.t. a general program  $P$ , we know that Prolog computation on that query and that program will terminate, and give the same computed answers as LDNF-resolution.

Now what if we have a finite P-tree for a restricted Prolog query  $Q$  and a restricted Prolog program  $P$ ? Consider the following restricted Prolog program

$$\begin{aligned} p &\leftarrow \\ p &\leftarrow p \end{aligned}$$

and the restricted Prolog query  $\text{neg}(p)$ . The P-tree and LDNF-tree for this query and this program are shown in Figure 5 (note that the pruned branches are not really part of the P-tree for  $\text{neg}(p)$ , but existed at some point during the construction of this P-tree). In this example, the P-tree is finite, because the potentially infinite branch caused by the clause  $p \leftarrow p$  is pruned. However, in the LDNF-tree, this branch has been constructed in full, and therefore this LDNF-tree is infinite.

## 6 APPLICATIONS

Due to the presence of cut in the definition of the predicate  $\text{neg}$  it is difficult to reason in a declarative way about Prolog programs that use negation. In other words, it is not clear how to prove correctness of such programs using their declarative interpretation.

We now show how this is possible using the results of this paper. The key observation is that Theorem 5.1 provides a crucial relationship between the computational behaviour of Prolog programs and their translations into general logic programs.

In the subsequent discussion we assume that the variables in the input clauses and the mgu's are chosen in a fixed way. We can then assume that for every Prolog program  $P$  and Prolog query  $Q$  there exists exactly one P-tree, and similarly for general logic programs, general queries and LDNF-trees.

So consider a restricted Prolog program  $P$  with a restricted query  $Q$  and their translation  $P_L$  and  $Q_L$  onto a general logic program and a general logic query, respectively. To reason about correctness of  $P$  with  $Q$  it is sufficient to reason about  $P_L$  and  $Q_L$ . Indeed, suppose that we proved already that all LDNF-derivations of  $P$  and  $Q$  are finite. Then by Theorem 5.1 the P-tree for  $P_L$  and  $Q_L$  is finite, and  $P_L$  with  $Q_L$  and  $P$  with  $Q$  have the same set of computed answers.

As an example consider the following well-known Prolog program TRANS about which one claims that it computes the transitive closure a binary relation  $e$ :

```

trans(X, Y, E, Avoids) ← member([X, Y], E).
trans(X, Z, E, Avoids) ←
    member([X, Y], E),
    neg(member(Y, Avoids)),
    trans(Y, Z, E, [Y | Avoids]).

member(X, [X | Xs]) ← .
member(X, [_ | Xs]) ← member(X, Xs).

```

In Apt [Apt94] the following facts about its translation  $TRANS_L$  to a general logic program and a binary relation  $e$  were established:

- all LDNF-derivations of  $trans(X, Y, e, [])$  are finite,
- the computed answer substitutions of  $trans(X, Y, e, [])$  determine all pairs of elements which form the transitive closure of  $e$ .

Now, by Theorem 5.1 the same conclusions can be drawn about the original program TRANS.

The fact that above approach to correctness is limited to restricted Prolog programs is in our opinion not serious. In fact, we noticed that practically all “natural” Prolog programs that use negation are restricted.

#### REFERENCES

- [AB87] B. Arbab and D.M. Berry. Operational and denotational semantics of Prolog. *Journal of Logic Programming*, 4(4):309–329, 1987.
- [AB94] K.R. Apt and R. Bol. Logic programming and negation: a survey. *Journal of Logic Programming*, 19-20:9–71, 1994.
- [AD94] K.R. Apt and K. Doets. A new definition of SLDNF-resolution. *Journal of Logic Programming*, 18(2):177–190, 1994.
- [Apt94] K. R. Apt. Program verification and Prolog. In E. Börger, editor, *Specification and Validation methods for Programming languages and systems*. Oxford University Press, 1994. To appear.

- [CKW89] W. Chen, M. Kifer, and D.S. Warren. Hilog: A first-order semantics for higher-order logic programming constructs. In *Proceedings of the North-American Conference on Logic Programming*, Cleveland, Ohio, October 1989.
- [Cla78] K.L. Clark. Negation as failure. In H. Gallaire and G. Minker, editors, *Logic and Data Bases*, pages 293–322. Plenum Press, 1978.
- [Dix93] J. Dix. Semantics of Logic Programs: Their Intuitions and Formal Properties. An Overview. In Andre Fuhrmann and Hans Rott, editors, *Logic, Action and Information. Proceedings of the Konstanz Colloquium in Logic and Information (LogIn '92)*. DeGruyter, 1993.
- [DM88] S.K. Debray and P. Mishra. Denotational and operational semantics for Prolog. *Journal of Logic Programming*, 5(1):61–91, 1988.
- [GHK<sup>+</sup>80] G. Gierz, K.H. Hofmann, K. Keimel, J.D. Lawson, M.W. Mislove, and D.S. Scott. *A Compendium of Continuous Lattices*. Springer-Verlag, 1980.
- [HLS90] P.M. Hill, J.W. Lloyd, and J.C. Shepherdson. Properties of a pruning operator. *Journal of Logic and Computation*, 1(1):99–143, 1990.
- [Jia94] Y. Jiang. Ambivalent logic as the semantic basis for metalogic programming: I. In P. Van Hentenryck, editor, *Proceedings of the International Conference on Logic Programming*, pages 387–401. MIT Press, June 1994.
- [JM84] N.D. Jones and A. Mycroft. Stepwise development of operational and denotational semantics for Prolog. In *International Symposium on Logic Programming*, pages 281–288, 1984.
- [Kal93] M. Kalsbeek. The vanilla meta-interpreter for definite logic programs and ambivalent syntax. Technical Report CT-93-01, Department of Mathematics and Computer Science, University of Amsterdam, The Netherlands, 1993.
- [LB92] A. Lilly and B.R. Bryant. A prescribed cut for Prolog that ensures soundness. *Journal of Logic Programming*, 14(4):287–339, 1992.
- [Llo87] J.W. Lloyd. *Foundations of Logic Programming*. Symbolic Computation – Artificial Intelligence. Springer-Verlag, 1987. Second, extended edition.
- [Mos86] C. Moss. Cut & Paste – defining the impure primitives of Prolog. In E. Shapiro, editor, *Proceedings of the International Conference on Logic Programming*, number 225 in Lecture Notes in Computer Science, pages 686–694. Springer Verlag, 1986.
- [MT92] M. Martelli and C. Tricomi. A new SLDNF-tree. *Information Processing Letters*, 43(2):57–62, 1992.
- [Ric74] B. Richards. A point of reference. *Synthese*, 28:431–445, 1974.



# The Manifold Coordination Language

To Cor Baayen, at the occasion of his retirement

F. Arbab

I. Herman

*CWI*

Email: farhad@cwi.nl, ivan@cwi.nl

Management of the communications among a set of concurrent processes arises in many applications and is a central concern in parallel computing. In this paper we introduce **MANIFOLD**: a *coordination* language whose sole purpose is to describe and manage complex interconnections among independent, concurrent processes. In the underlying paradigm of this language the primary concern is not with *what* functionality the individual processes in a parallel system provide. Instead, the emphasis is on *how* these processes are inter-connected and how their interaction patterns change during the execution life of the system. This paper also includes an overview of our implementation of **MANIFOLD**.

As an example of the application of **MANIFOLD**, we present a series of small manifold programs which describe the skeletons of some adaptive recursive algorithms that are of particular interest in computer graphics. Our concern in this paper is to show the expressibility of **MANIFOLD** and its usefulness in practice. Issues regarding performance and optimization are beyond the scope of this paper.

## 1 INTRODUCTION

Specification and management of the communications among a set of concurrent processes is at the core of many problems of interest to a number of contemporary research trends. The theory of neural networks and the connectionist view of computation emphasize the significance of the concept of *management of connections* versus the local computation abilities of each node. The concept of dataflow programming has a certain resemblance with connectionism, albeit, it is closer to the discrete world of conventional programming than neural networks. Theoretical work on concurrency, e.g., CCS [1] and CSP [2, 3], is primarily concerned with the semantics of communications and interactions of concurrent sequential processes. Communication issues also come up in virtually every other type of computing, and have influenced the design (or at least,

a few constructs) of most programming languages. However, not much effort has been spent on conceptual models and languages whose sole prime focus of attention is on the coordination of interactions among processes.

In their recent paper [4], Gelernter and Carriero elaborate the distinction between *computational models and languages* versus *coordination models and languages*. They correctly observe that relatively little serious attention has been paid in the past to the latter, and that “ensembles” of asynchronous processes (many of which are off-the-shelf programs) running on parallel and distributed platforms will soon become predominant.

**MANIFOLD** is a language whose sole purpose is to manage complex interconnections among independent, concurrent processes. As such, like LINDA [5, 6], it is primarily a coordination language. However, there is no resemblance between LINDA and **MANIFOLD**, nor is there any similarity between the underlying models of these two languages. The details of the **MANIFOLD** model and the syntax and semantics of the **MANIFOLD** language are, of course, beyond the scope of this paper and are described in a separate document [7]. In this paper, we give an overview of the **MANIFOLD** language and its implementation and present the skeleton of some recursive algorithms which are of particular interest in computer graphics. Also, an application of the language in the field of scientific visualization is presented. We summarize only enough of the description of the **MANIFOLD** model and language here, to make the examples and the significant implementation issues presented in this paper understandable.

The rest of this paper is organized as follows. In §2 the main motivations behind the **MANIFOLD** language and its underlying computing model are discussed. In §3 a more detailed description of the language is presented. In §4 we mention some of the application areas where **MANIFOLD** can prove to be a useful tool. In §5, we present the skeleton of a few adaptive recursive algorithms taken from the field of computer graphics. The purpose of these examples is to illustrate the use of some of the features of the **MANIFOLD** language and to demonstrate the general applicability of **MANIFOLD** concepts. The analysis of these programs gives us a good opportunity to show the descriptive power of **MANIFOLD**. In §6, we discuss some of the similarities and major differences between **MANIFOLD** and certain related systems and models for parallel computing. In §7 we mention some of the extensions and enhancements we plan to make to the **MANIFOLD** system in the future. Finally, §8 concludes this paper.

## 2 MOTIVATION

One of the fundamental problems in parallel programming is coordination and control of the communications among the sequential fragments that comprise a parallel program. Programming of parallel systems is often considerably more difficult than (what intuitively seems to be) necessary. It is widely acknowledged that a major obstacle to a more widespread use of massive parallelism is the lack of a coherent model of how parallel systems must be organized and programmed. To complicate the situation, there is an important pragmatic

concern with significant theoretical consequences on models of computation for parallel systems. Many user communities are unwilling and/or cannot afford to ignore their previous investment in existing algorithms and “off-the-shelf” software and migrate to a new and bare environment. This implies that a suitable model for parallel systems must be *open* in the sense that it can accommodate components that have been developed with little or no regards for their inclusion in an environment where they must interact and cooperate with other modules.

Many approaches to parallel programming are based on the same computation models as sequential programming, with added on features to deal with communications and control. This is the case for such concurrent programming languages like Ada [8], Concurrent C [9, 10], Concurrent C++ [11], Occam [12] and many others (the interested reader may consult, e.g., the survey of Bal et al. [13] for more details on these languages).

There is an inherent contradiction in such approaches which shows up in the form of complex semantics for these added on features. The fundamental assumption in sequential programming is that there is only one active entity, *the* processor, and the executing program is in control of this entity, and thus in charge of the application environment. In parallel programming, there are many active entities and a sequential fragment in a parallel application cannot, in general, make the convenient assumption that it can rely on its incrementally updated model of its environment.

To reconcile the “disorderly” dynamism of its environment with the orderly progression of a sequential fragment, “quite a lot of things” need to happen at the explicit points in a sequential fragment when it uses one of the constructs to interact with its environment. Hiding all that needs to happen at such points in a few communication constructs within an essentially sequential language, makes their semantics extremely complex. Inter-mixing the neat consecutive progression of a sequential fragment, focused on a specific function, with updating of its model of its environment and explicit communications with other such fragments, makes the dynamic behavior of the components of a parallel application program written in such languages difficult to understand. This may be tolerable in applications that involve only small scale parallelism, but becomes an extremely difficult problem with massive parallelism.

Contrary to languages that try to hide as much of the “chaos of parallelism” as possible behind a facade of sequential programming, **MANIFOLD** is based on the idea that allowing programmers to see and feel this parallelism is actually beneficial. It is a formidable intellectual experience to realize that if one frees oneself from the confines of the sequential paradigm and accepts that logical processes are “cheap” (that is, they are fast to activate and to communicate with), then a number of practical problems and applications can be described and solved incomparably more easily and more elegantly. In other words, there often *is* a pay-off in using parallel or distributed programming, even if higher speeds are not (necessarily) achieved. Just as a practical example, the basic approach of using multi-processing is very clearly one of the reasons for the un-

deniable technical superiority of the NeWS windowing system over X Windows [14]; also, almost all the applications listed in §4 fall in this category.

The assumption of having cheap logical processes is not only in line with the direction of future hardware development, it is also compatible with the current trend in the evolution of contemporary software systems. The increasingly more frequent use of so-called “light-weight” processes within conventional operating systems<sup>1</sup> is a clear indication (see, for example, the Brown University Thread Package [15], the so-called  $\mu$ System [16], or even the way some of the above cited languages, e.g., AT&T’s Concurrent C, are implemented). More recent operating system designs offer light-weight processes in their kernels (e.g., OSF/1, based on the Mach system [17, 18] of Carnegie Mellon, or SunOS [19]).

Separating communication issues from the functionality of the component modules in a parallel system makes them more independent of their context, and thus more reusable. It also allows delaying decisions about the interconnection patterns of these modules, which may be changed subject to a different set of concerns. This idea is one of the main motivations behind the development of the **MANIFOLD** system.

There are even stronger reasons in distributed programming for delaying the decision about the interconnections and the communication patterns of modules. Some of the basic problems with the parallelism in parallel computing become more acute in real distributed computing, due to the distribution of the application modules over loosely coupled processors, perhaps running under quite different environments in geographically different locations. The implied communications delays and the heterogeneity of the computational environment encompassing an application become more significant concerns than in other types of parallel programming. This mandates, among other things, more flexibility, reusability, and robustness of modules with fewer hard-wired assumptions about their environment.

The tangible payoffs reaped from separating the communications aspect of a multi process application from the functionality of its individual processes include clarity, efficiency, and reusability of modules and the communications specifications. This separation makes the communications control of the cooperating processes in an application more explicit, clear, and understandable at a higher level of abstraction. It also encourages individual processes to make less severe assumptions about their environment. The same communications control component can be used with various processes that perform functions *similar* to each other from a very high level of abstraction. Likewise, the same processes can be used with quite different communications control components.

### 3 THE MANIFOLD LANGUAGE

In this section we give a brief and informal overview of the **MANIFOLD** language. The sole purpose of the **MANIFOLD** language is to describe and manage

---

<sup>1</sup>Some authors prefer the term “pseudo-parallelism” for such or similar forms of parallelism, again, see Bal et al [13].

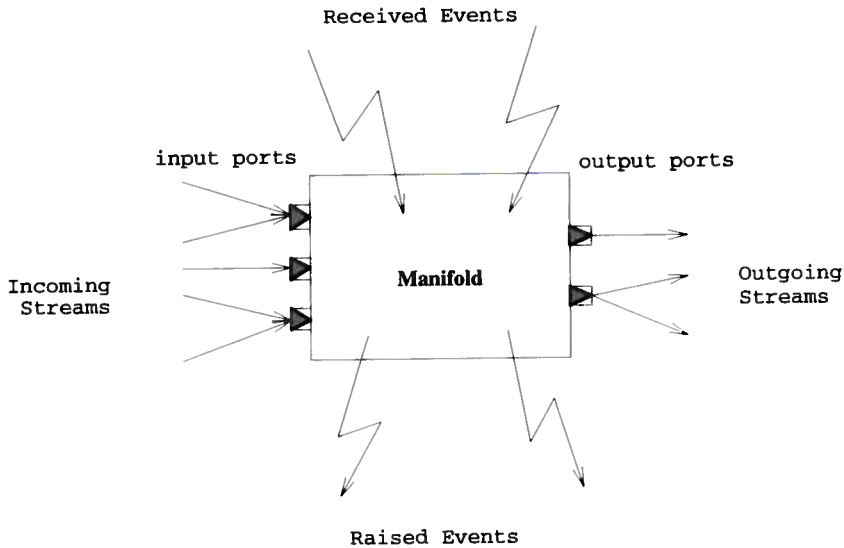


FIGURE 1. The model of a process in Manifold.

complex communications and interconnections among independent, concurrent processes. As stated earlier, a detailed description of the syntax and the semantics of the **MANIFOLD** language and its underlying model is given elsewhere [7]. Other reports contain more examples of the use of the **MANIFOLD** language [20, 21, 22, 23].

The basic components in the **MANIFOLD** model of computation are *processes*, *events*, *ports*, and *streams*. A process is a *black box* with well defined ports of connection through which it exchanges *units* of information with the other processes in its environment. The internal operation of some of these black boxes are indeed written in the **MANIFOLD** language, which makes it possible to open them up, and describe their internal behavior using the **MANIFOLD** model. These processes are called *manifolds*. Other processes may in reality be pieces of hardware, programs written in other programming languages, or human beings. These processes are called *atomic processes* in **MANIFOLD**. In fact, an atomic process is any processing element whose external behavior is all that one is interested in observing at a given level of abstraction. In general, a process in **MANIFOLD** does not, and need not, know the identity of the processes with which it exchanges information. Figure 1 shows an abstract representation of a **MANIFOLD** process.

Ports are regulated openings at the boundaries of processes through which they exchange units of information. The **MANIFOLD** language allows assigning special filters to ports for screening and rebundling of the units of information exchanged through them. These filters are defined in a language of extended regular expressions. Any unit received by a port that does not match its regular



expression is automatically diverted to the **error** port of its manifold and raises a **badunit** event (see later sections for the details of events and their handling in **MANIFOLD**). The regular expressions of ports are an effective means for “type checking” and can be used to assure that the units received by a manifold are “meaningful.”

Interconnections between the ports of processes are made with *streams*. A stream represents a flow of a sequence of units between two ports. Conceptually, the capacity of a stream is infinite. Streams are dynamically constructed between ports of the processes that are to exchange some information. Adding or removing streams does not directly affect the status of a running process. The constructor of a stream (which is a manifold) need not be the sender nor the receiver of the information to be exchanged: any third party manifold process can define a connection between the ports of a producer process and a consumer process. Furthermore, stream definitions in **MANIFOLD** are generally additive. Thus a port can simultaneously be connected to many different ports through different streams (see for example the network in Figure 2). The flows of units of information in streams are automatically replicated and merged at outgoing and incoming port junctions, as necessary. The units of information exchanged through ports and streams, are *passive* pieces of information that are produced and consumed at the two ends of a stream with their relative order preserved. The consumption and production of units via ports by a process is analogous to read and write operations in conventional programming languages. The word “passive” is meant to suggest the similarity between units and the data exchanged through such conventional I/O operations.

Independent of the stream mechanism, there is an event mechanism for information exchange in **MANIFOLD**. Contrary to units in streams, events are *atomic* pieces of information that are *broadcast* by their sources in their environment. In principle, *any* process in an environment can pick up a broadcast event. In practice, usually only a few processes pick up occurrences of each event, because only they are “tuned in” to their sources. Occurrences of the same event from the same source can override each other from the point of view of some observer processes, depending on the difference between the speed of the source and the reaction time of an observer. This provides an automatic *sampling* mechanism for observer processes to pick up information from their environment which is particularly useful in situations where a potentially significant mismatch between the speeds of a producer and a consumer is possible. Events are the primary control mechanism in **MANIFOLD**.

Once an event is raised by a source, it generally continues with its processing, while the event occurrence propagates through the environment independently. Event occurrences are active pieces of information in the sense that in general, they are observed asynchronously and once picked up, they preemptively cause a change of state in the observer. Communication of processes through events is thus inherently asynchronous in **MANIFOLD**.

Each manifold defines a set of events and their sources whose occurrences it is interested to observe; they are called the *observable* set of events and sources,

respectively. It is only the occurrences of observable events from observable sources that are picked up by a manifold. Once an event occurrence is picked up by an observer manifold, it may or may not cause an immediate reaction by the observer. In general, each state in a manifold defines the set of events (and their sources) that are to cause an immediate reaction by the manifold while it is in that state. This set is called the *preemption set* of a manifold state and is a subset of the observable events set of the manifold. Occurrences of all other observable events are *saved* so that they may be dealt with later, in an appropriate state.

Each state in a manifold defines a pattern of connections among the ports of some processes. The corresponding streams implementing these connections are created as soon as a manifold makes a state transition (caused by an event) to a new state, and are deleted as soon as it makes a transition from this state to another one. This is discussed in more detail in §3.2.

### 3.1 Manifold Definition

A manifold definition consists of a *header*, *public declarations*, and a *body*. The header of a manifold definition contains its name and the list of its formal parameters. The public declarations of a manifold are the statements that define its links to its environment. It gives the types of its formal parameters and the names of events and ports through which it communicates with other processes. A manifold body primarily consists of a number of *event handler blocks*, representing its different execution-time states. The body of a manifold may also contain additional declarative statements, defining *private* entities. For an example of a very simple manifold, see Listing 1 which shows the **MANIFOLD** source code for a simple program.<sup>2</sup> More complete manifold programs are also presented, e.g., in §5. Declarative statements may also appear outside of all manifold definitions, typically at the beginning of a source file. These declarations define global entities which are accessible to all manifolds in the same file, provided that they do not redefine them in their own scopes.

Conceptually, each activated instance of a manifold definition – a *manifold* for short – is an independent process with its own virtual processor. A manifold processor is capable of performing a limited set of actions. This includes a set of *primitive actions*, plus the primary action of setting up *pipelines*.

Each event handler block describes a set of actions in the form of a *group* construct. The actions specified in a group are executed in some non-deterministic order. Usually, these actions lead to setting up *pipelines* between various ports of different processes. A *group* is a comma-separated list of members enclosed in a pair of parentheses. In the degenerate case of a singleton group (which contains only one member) the parentheses may be deleted. Members of a group are either primitive actions, pipelines, or groups. The setting up of pipelines

---

<sup>2</sup>In this and other **MANIFOLD** program listings in this paper, the characters “//” denote the beginning of a comment which continues up to the end of the line. Keywords are typeset in bold.



```

// This is the header (there are no arguments):
example()
// These are the public declarations:
// Two ports are visible from the outside of the manifold "example";
// one is an input port and the other is an output one.
// In fact, these ports are the default ones.
port in input.
port out output.
{
  // The body of the manifold begins here.
  //
  // private declarations:
  // three process instances are defined:
  process A is A.type.
  process B is B.type.
  process C is C.type.

  // First block (activated when "example" becomes active)
  // The processes described above are activated on their turn
  // in a "group" construct:
  start: ( activate A, activate B, activate C ); do begin.

  // A direct transfer to this block has been given from "start".
  // Three pipelines in a group are set up:
  begin: ( A → B,output → C,input → output).

  // Event handler for the event "e1"; several pipelines are
  // set up (see Figure 2):
  e1: (B → input,C → A,A → B,output → A,B → C,input → output).

  // Event handler for the event "e2"; a single pipeline
  // is set up (see Figure 3):
  e2: C → B.
}

```

Listing 1. An example for a manifold process.

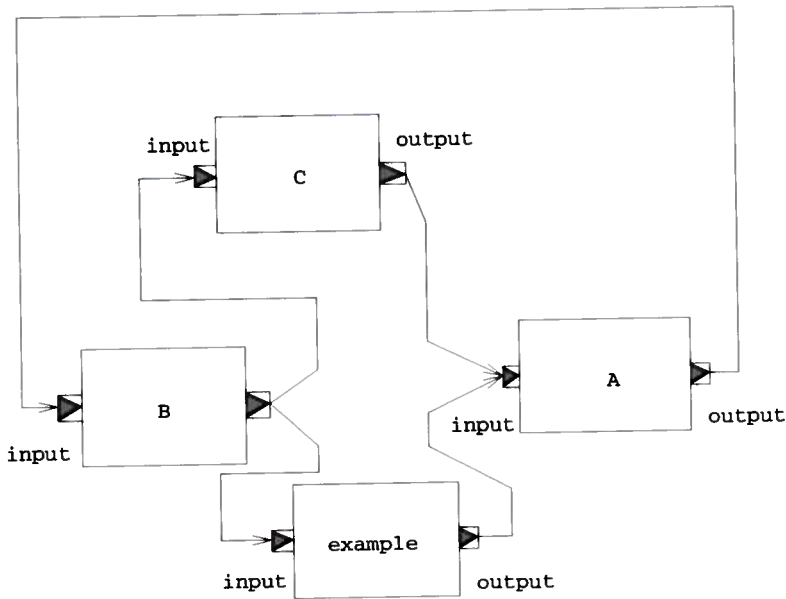


FIGURE 2. Connections set up by the manifold `example` on event `e1`.

within a group is simultaneous and atomic. No units flow through any of the streams inside a group before all of its pipelines are set up. Once set up, all pipelines in a group operate in parallel with each other.

A *primitive action* is typically *activating* or *deactivating* a process, *raising* an event, or a *do* action which causes a transition to another handler block without an event occurrence from outside. A *pipeline* is an expression defining a tandem of streams, represented as a sequence of one or more groups, processes, or ports, separated by right arrows. It defines a set of simultaneous connections among the ports of the specified groups and processes. If the initial (final) name in such a sequence is omitted, the initial (final) connection is made to the current input (output) port. Inside a group, the current input and output ports are the input and output ports of the group. Elsewhere, the current input and output ports are `input` and `output`, i.e., the executing manifold's standard input and output ports. As an example, Figure 2 shows the connections set up by the manifold process `example` on Listing 1, while it is in the handling block for the event `e1` (for the details of event handling see §3.2). Figure 3 shows the connections set up in the handling block for the event `e2`.

In its degenerate form, a pipeline consists of the name of a single port or process. Defining no useful connections, this degenerate form is nevertheless sometimes useful in event handler blocks because it has the effect of defining the named port or process as an observable source of events and a member of the preemption set of its containing block (see §3.4).

An event handler block may also describe sequential execution of a series of

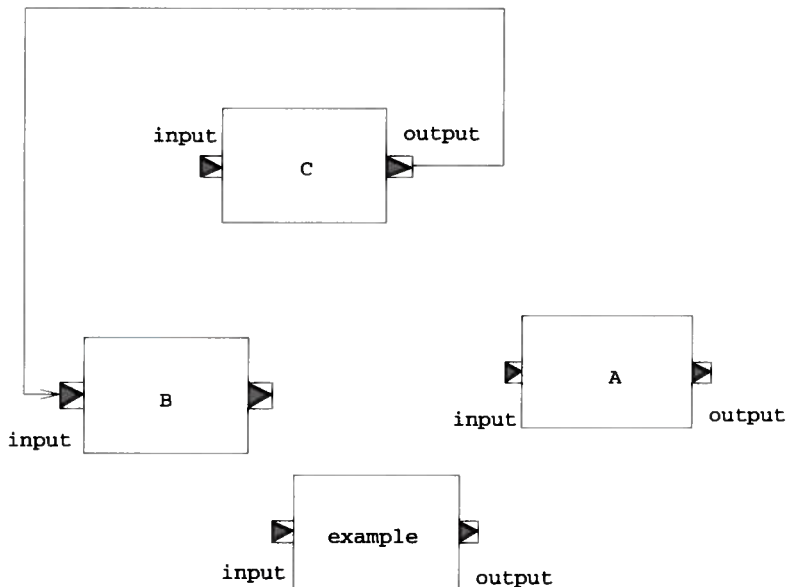


FIGURE 3. Connections set up by the manifold `example` on event `e2`.

(sets of) actions, by specifying a list of pipelines and groups, separated by the semicolon (;) operator<sup>3</sup>. In reaction to a recognized event, a manifold processor finds its appropriate event handler block and executes the list of sequential sets of actions specified therein. Once the manifold processor is through with the sequence in its current block, it terminates.

### 3.2 Event Handling

Event handling in **MANIFOLD** refers to a preemptive change of state in a manifold that observes an event of interest. This is done by its manifold processor which locates a proper event handler for the observed event occurrence. An event handler is a labeled block of actions in a manifold. In addition to the event handling blocks explicitly defined in a manifold, a number of default handlers are also included by the **MANIFOLD** compiler in all manifolds to deal with a set of predefined system events. The manifold processor makes a transition to an appropriate block (which is determined by its current state, the observed event and its source), and starts executing the actions specified in that block. The block is said to *capture* the observed event (occurrence). The name of the event that causes a transfer to a handling block, and the name of its source, are available in each block through the pseudonyms `event_name`

<sup>3</sup>In fact, the semicolon operator is only an infix *manner call* (see §3.5) rather than an independent concept in **MANIFOLD**. However, for our purposes, we can assume it to be the equivalent of the sequential composition operator of a language like Pascal.

and `event_source`, respectively.

The manifold processor finds the appropriate handler block for an observed event  $e$  raised by the source  $s$ , by performing a circular search in the list of block labels of the manifold. The list of block labels contains the labels of all blocks in a manifold in the sequential order of their appearance. The circular search starts with the labels of the current block in the list, scans to the end of the list, continues from the top of the list, and ends with the labels of the block preceding the current block in the list.

The manifold processor in a given manifold is sensitive to (i.e., interested in) only those events for which the manifold has a handler. All other events are to be ignored. Thus, events that do not match any label in this search do not affect the manifold in any way (however, see §3.5 for the case of called manners). Similarly, if the appropriate block found for an event is the keyword `ignore`, the observed event is ignored. Normally, events handled by the current block are also ignored.

The concept of an event in **MANIFOLD** is different than the concepts with the same name in most other systems, notably simulation languages, or CSP [2, 3]. Occurrence of an event in **MANIFOLD** is analogous to a flag that is raised by its source (process or port), *irrespective* of any communication links among processes. The source of an event continues immediately after it raises its flag, independent of any potential observers. This raised flag can potentially be seen by any process in the environment of its source. Indeed, it can be seen by any process to which the source of the event is *visible*. However, there are no guarantees that a raised flag will be observed by anyone, or that if observed, it will make the observer react immediately.

### 3.3 Event Handling Blocks

An event handling block consists of a comma-separated list of one or more block labels followed by a colon (`:`) and a single body. The body of an event handling block is either a group member (i.e., an action, a pipeline, or a group), or a single manner call (see §3.5). If the body of a block is a pipeline, and it starts (ends) with a `→`, the port name `input` (respectively, `output`) is prepended (appended) to the pipeline.

Event handler block labels are patterns designating the set of events captured by their blocks. Blocks can have multiple labels and the same label may appear more than once marking different blocks. Block labels are filters for the events that a manifold will react to. The filtering is done based on the event names and their sources. Event sources in **MANIFOLD** are either ports or processes.

The most specific form of a block label is a dotted pair  $e.s$ , designating event  $e$  from the source (port or process)  $s$ . The wild-card character `*` can be replaced for either  $e$ , or  $s$ , or both, in a block label. The form  $e$  is a short-hand for  $e.*$  and captures event  $e$  coming from any source. The form  $*.s$  captures any event from source  $s$ . Finally, the least specific block label is  $.*$  (or  $*$ , for short) which captures any event coming from any source.

### 3.4 *Visibility of Event Sources*

Every process instance or port defined or used anywhere in a manner (see §3.5) or manifold is an *observable* source of events for that manner or manifold. This simply means that occurrences of events raised by such sources (only) will be picked up by the executing manifold processor, provided that there is a handling block for them. The set of all events from observable sources that match any of the block labels in a manner or manifold is the set of observable events for that manner or manifold. The set of observable events of an executing manifold instance may expand and shrink dynamically due to manner calls and terminations (see §3.5). Depending on the state of a manifold processor (i.e., its current block), occurrences of observable events cause one of two possible actions: preemption of the current block, or saving of the event occurrence.

In each block, a manifold processor can react to only those events that are in the *preemption set* of that block. The **MANIFOLD** language defines the preemption set of a block to contain only those observable events whose sources appear in that block. This means that, while the manifold processor is in a block, except for the manifold itself, no process or port other than the ones named in that block can be the source of events to which it reacts immediately. There are other rules for the visibility of parameters and the operands of certain primitive actions. It is also possible to define certain processes as permanent sources of events that are visible in all blocks. A manifold can always internally raise an event that is visible only to itself via the **do** primitive action.

Once the manifold processor enters a block, it is immune to any of the events handled by that block, except if the event is raised by a **do** action in the block itself. This temporary immunity remains in effect until the manifold processor leaves the block. Other observable event occurrences that are not in the preemption set of the current block are saved.

### 3.5 *Manners*

The state of a manifold is defined in terms of the events it is sensitive to, its visible event sources, and the way in which it reacts to an observed event. The possible states of a manifold are defined in its blocks, which collectively define its behavior. It is often helpful to abstract and parameterize some specific behavior of a manifold in a subroutine-like module, so that it can be invoked in different places within the same or different manifolds. Such modules are called *manners* in **MANIFOLD**.

A *manner* is a construct that is syntactically and semantically very similar to a manifold. Syntactically, the differences between a manner definition and a manifold definition are:

1. The keyword **manner** appears in the header of a manner definition, before its name.
2. Manner definitions cannot have their own port definitions.

Semantically, there are two major differences between a manner and a manifold. First, manners have no ports of their own and therefore cannot be connected to streams. Second, a manner invocation never creates a new processor. A manifold activation always creates a new processor to “execute” the new instance of the manifold. To invoke a manner, however, the invoking processor itself “enters and executes” the manner.

The distinction between manners and manifolds is similar to the distinction between procedures and tasks (or processes) in other distributed programming languages. The term *manner* is indicative of the fact that by its invocation, a manifold processor changes its own context in such a way as to behave in a different manner in response to events.

Manner invocations are dynamically nested. References to all non-local names in a manner are left unresolved until its invocation time. Such references are resolved by following the dynamic chain of manner invocations in a last-in-first-out order, terminating with the environment of the manifold to which the executing processor belongs.

Upon invocation of a manner, the set of observable events of the executing manifold instance expands to the union of its previous value and the set of observable events of the invoked manner. The new members thus added to this set, if any, are deleted from the set upon termination of the invoked manner.

A manner invocation can either terminate normally or it can be preempted. Normal termination of a manner invocation occurs when a **return** primitive action is executed inside the manner. This returns the control back to the calling environment right after the manner call (this is analogous to returning from a subroutine call in conventional programming languages). Preemption occurs when a handling block for a recognized event occurrence cannot be found inside the actual manner body. This initiates a search through the dynamic chain of activations similar to the case of resolving references to non-local names, to find a handler for this event. If no such handler is found, the event occurrence is ignored. If a suitable handler is found, the control returns to its enclosing environment and all manner invocations in between are abandoned.

Manners are simply declarative “subroutines” that allow encapsulation and reuse of event handlers. The search through the dynamic chain of manner calls is the same as dynamic binding of handlers in calling environments, with event occurrences picked up in a called manner. Preemption is nothing but cleanly structured returns by all manner invocations up to the environment of a proper handler.

In principle, dynamic binding can be replaced by the use of (appropriately typed) parameters. Our preference for dynamic binding in manners is motivated by pragmatic considerations. Suppose a piece of information (e.g., how to handle a particular event, or where to return to) must be passed from a calling environment A, to a called environment B, through a number of intermediaries; i.e., B is *not* called directly by A, but rather, A calls some other “subroutine” which calls another one, which calls yet another one, . . . , which eventually calls



B. Passing this information from A to B using parameters means that all intermediaries must know about it and explicitly pass it along, although it has no functional significance for them. Dynamic binding alleviates the need for this explicit passing of irrelevant information and makes the intermediary routines more general, less susceptible to change, and more reusable.

### 3.6 Scope Rules

The scope of a name is the syntactic context wherein that name is known as to denote the same entity. The scope of the names of atomic process specifications, manner definitions, and manifold definitions contained in a source file is the entire source file. The scope of the names defined in the private declarative section (inside the body) of a manifold or manner is the manifold or the manner itself. The scope of the names defined in the declarative statements outside of any manifold or manner definition, is the entire source file.

Ports of a manifold or atomic process are accessible to any process that knows its name and the name of its ports. Ports of a process, together with the events defined in its public declaration section, provide the communication links of a process with other processes running in its environment.

Except in manners, non-local names (i.e., used but not defined in a context), are statically bound to the entities with the same name in their enclosing contexts. It is a compile-time error if such a non-local name remains unresolved. The binding of non-local names (i.e., used but not defined) in manners is dynamic: these names are bound upon activation of a manner to the entities with the same name in the environment of its caller. The chain of manner activations leading to the present activation are traversed all the way up to the environment of a manifold instance, in search of appropriate targets for this binding. Names that remain unresolved at this point are bound to appropriate benign defaults (e.g., `void` described in §5.1.1).

**MANIFOLD** supports separate compilation. This is a very effective mechanism for modularization of large applications. In principle, all names defined and used in a source file are strictly local to that file. Names (of events, manners, manifolds, or atomic processes) that are used in different source files and must indeed designate the same entity at execution time, must be explicitly declared as such using `extern`, `import`, and `export` constructs (see [7]).

## 4 APPLICATIONS

The **MANIFOLD** language has already been used to describe some simple examples, like a parallel bucket sort algorithm, a simplified version of a (graphics) resource management and the like. The interested reader is referred to the reports published elsewhere [20, 21]. These examples were primarily meant to test the **MANIFOLD** concepts themselves. In this section we mention some of the possible application areas for **MANIFOLD** in large-scale and non-trivial parallel systems.



**MANIFOLD** is an effective tool for describing interactions of autonomous active agents that communicate in an environment through address-less messages and global broadcast of events. For example, elaborate user interface design means planning the cooperation of different entities (the human operator being one of them) where the event driven paradigm seems particularly useful. In our view, the central issue in a user interface is the design and implementation of the communication patterns among a set of modules<sup>4</sup>. Some of these modules are generic (application independent) programs for acquisition and presentation of information expressed in forms appealing to humans. Others are, ideally, acquisition/presentation-independent modules that implement various functional components of a specific application. Previous experience with User Interface Management Systems (see, e.g., [24]) has shown that concurrency, event driven control mechanisms, and general interconnection networks are all necessary for effective graphical user interface systems. **MANIFOLD** supports all of that and, in addition, provides a level of dynamism that goes beyond many other user interface design tools. As an example, it has recently been used to successfully reformulate the GKS<sup>5</sup> input model [25]; this work is regarded as a starting point in the development of new concepts for highly flexible, reconfigurable graphics systems suitable for parallel environments.

Separating the specification of the dynamically changing communication patterns among a set of concurrent modules from the modules themselves, seems to lead to better user interface architectures. A similar approach can also be useful in applications of real time computing where dynamic change of interconnection patterns (e.g., between measurement and monitoring devices and actuators) is crucial. For example, complex process control systems must orchestrate the cooperation of various programs, digital and/or analogue hardware, electronic sensors, human operators, etc. Such interactions may be more easily expressed and managed in **MANIFOLD**.

Coordination of the interactions among a set of cooperating autonomous intelligent experts is also relevant in Distributed Artificial Intelligence applications, open systems such as Computer Integrated Manufacturing applications, and the complex control components of systems such as Intelligent Computer Aided Design.

Recently, scientific visualization has raised similar issues as well. The problems here typically involve a combination of massive numerical calculations (sometimes performed on supercomputers) and very advanced graphics. Such functionality can best be achieved through a distributed approach, using segregated software and hardware tools. Tool sets like the Utah Raster Toolkit [26] were already a first step in this direction, although in the case of this toolkit the individual processes can be connected in a pipeline fashion only. More recently, software systems like the apE system of the Ohio Supercomputer

---

<sup>4</sup>In fact, given the previous experiences of the authors, the problems arising in user-interface techniques provided some of the basic motivation to start this project in the first place.

<sup>5</sup>Graphical Kernel System is the ISO Standard for Computer Graphics.

Center [27], the commercially available AVS Visualization Package of Stardent Computer Ltd. [28], the IRIS Explorer system [29] and others, work on the basis of inter-connecting a whole set of different software/hardware components in a more sophisticated communication network. The successes of these packages, and mainly the general ideas behind them, point toward a more general development trend which leads to reconsideration of the software architecture used for graphics packages in general.

For the emerging new technologies and application areas that are expected to result in a tremendous growth in computer graphics in the nineties, a new software base is necessary to accommodate demands for high performance special hardware, dedicated application systems, distributed and parallel computing, scientific visualization, object-oriented methods and multi-media, to name just a few. Some of the major technical concerns in the specification and the development of new graphics systems is *extensibility* and *reconfigurability*. To ensure these features it is feasible to envisage a highly parallel architecture which is based on the concept of cooperating, specialized agents with well defined but reconfigurable communication patterns. An “orchestrator” like **MANIFOLD** can prove to be quite valuable in such applications.

## 5 ADAPTIVE RECURSIVE ALGORITHMS IN MANIFOLD

In this section, a well-known class of algorithms in the field of computer graphics and image processing is described using the **MANIFOLD** formalism. It is *not* the purpose of this section to analyze these methods from a strictly algorithmic point of view, nor do we intend to devise new versions of already existing algorithms. We simply intend to show the descriptive power of **MANIFOLD** using well-established algorithms.

It is beyond the scope of this paper to give all the specific details of each algorithm. The interested reader can consult one of the standard textbooks on computer graphics and/or image processing (e.g., [30] for computer graphics and [31] for image processing) or refer to the literature given in the references (e.g., [32, 33, 34, 35, 36] or others).

### 5.1 Warnock's Algorithm

One of the very well known problems in computer graphics is what is usually referred to as Hidden Surface Removal. The problem is as follows. When a three-dimensional scene, usually modeled using a large number of planar polygons in space, is visualized on a screen, all of its polygons must be projected onto a plane (i.e., the plane of the display screen) from a given viewpoint. Mathematically, this projection is well understood, but there is an additional problem to solve: those polygons, or parts of polygons, that are occluded by another one, as seen from the selected viewpoint, must be eliminated. The removal of these (sub-)polygons is what is called the removal of hidden surfaces.

There are several well-known and widely applied solutions to this problem. One of the earliest is Warnock's algorithm which is described in detail in the

```

TestAndColor() import.
DivideArea()
  port out first_area.
  port out second_area.
  port out third_area.
  port out fourth_area.
  import.

export Warnock()
{
  process test_and_color is TestAndColor.
  process divide_area is DivideArea.
  process v is variable.
  process n is variable.
start:
  ( activate v, activate n
    activate test_and_color,
    input → (→ test_and_color,→ v),
  ).
subdivide:
  ( activate divide_area,
    v → divide_area,
    divide_area.first_area → Warnock(),
    divide_area.second_area → Warnock(),
    divide_area.third_area → Warnock(),
    divide_area.fourth_area → Warnock(),
    n = 4
  );
  do wait_to_die.
terminate:
  save.
wait_to_die:
  void.
terminate:
  n = n - 1;
  if( n == 0, do end, do wait_to_die ).
done:
  do end.
end:
  deactivate parent.
}

```

Listing 2. Manifold Program for Warnock's Algorithm.

literature, e.g., in [30]. A short description of this algorithm is as follows.

This algorithm is based on a recursive area-subdivision of the computer screen. At each stage in the recursive subdivision process, the projection of each polygon has one of four relationships to the area of interest (which is, at the beginning, the full screen of the display):

1. *surrounding polygons* completely contain the area of interest;
2. *intersecting polygons* intersect the area;
3. *contained polygons* are completely inside the area;
4. *disjoint polygons* are completely outside the area.

Based on these tests, there are certain cases where the exact color(s) for rendering the area of interest can be determined very easily. Obvious cases include when all polygons are disjoint from the area (and hence the background color can be used), when there is only one polygon which either intersects the area or is contained in it, or when there is one and only one polygon which completely surrounds the area. There are also some less obvious but still easily decidable cases which the original version of the algorithm takes into account.

There are, however, cases where there is no easy way to color the area. In these cases, Warnock's algorithm subdivides the area into four equal sub-areas to simplify the problem and then the same method is applied recursively for each of the four sub-areas. The recursion stops when the dimension of the sub-area has reached the size of one pixel on the screen; some additional calculations are then done to determine the color of this single pixel.

### 5.1.1 A Manifold Program for Warnock's Algorithms

Before commenting further on the algorithm, let us see how its skeleton can be described using **MANIFOLD**. The complete listing of the program appears as Listing 2.

The program uses two (atomic) processes which implement its truly algorithm specific and numerically oriented details. These atomic processes are "imported", which means that they are external to the present **MANIFOLD** source file and will be made available at link-time. **TestAndColor** is supposed to receive the description of an area on its standard input (as far as **MANIFOLD** is concerned, this description is just an abstract unit to be forwarded; we refer to it as "area handle" in what follows). It then performs the test on all polygons in the scene, following the scheme described in the previous section. The result of this step is either:

- the area can be filled without ambiguities, in which case **TestAndColor** raises the event **done**, fills the area with the calculated color(s) and terminates; or

- the area cannot be filled without ambiguities, in which case `TestAndColor` raises the event `subdivide` and terminates.

The atomic process `DivideArea` receives an area handle on its standard input; it has, apart from the standard ports, four publicly declared output ports, onto which it places the four area handles after it performs a subdivision. Once these units are produced, `DivideArea` terminates.

It is the manifold process `Warnock` that embodies the skeleton of Warnock's algorithm. It is important to understand the details of this program to gain a real insight into the descriptive power of `MANIFOLD`; this is why a more detailed description of this process is given in what follows.

In the declaration part of `Warnock`, two instances of the atomic processes described above are declared. This means that the manifold `Warnock` now has a reference for these processes and can, therefore, involve them into several parallel pipelines, if necessary. The additional two declarations concern two "utility" processes (part of the standard environment of the `MANIFOLD` system) which are able to store some units and, if the type of the units permit, to perform some elementary arithmetic on them.

The start state of `Warnock` activates the two variable processes and the local instance of `TestAndColor`. A pipeline is then set up, which involves a group as well. This pipeline describes the following relationships:

- a unit (*i.e.*, an area handle) arriving on the input of `Warnock` is redirected to the local instance of `TestAndColor`, and
- a copy of the same unit is "stored" in the variable `v`.

The manifold is suspended in this block and must receive an external event to change its state. According to our specifications, these external events may be either `subdivide` or `done`, depending on the result of the test performed on the local area. (Note that although many instances of `TestAndColor` may be active and raise the events `subdivide` and/or `done`, the only instance of `TestAndColor` visible to an instance of `Warnock` is its locally declared one. This is why the other events raised by other instances cause no confusion.)

The state labeled `subdivide` is obviously the essential part of the manifold `Warnock`. The corresponding block contains, in fact, two statements, joined by the connective ";", which can be thought of as a delimiter for sequential execution. In the first statement, the local instance of the atomic process `DivideArea` is activated and, also, four *independent* instances of the manifold `Warnock` are implicitly created and activated (using a process specification name in a statement, instead of declaring an instance in the declaration section, means the implicit creation and activation of an instance of that process). The pipelines defined in the group are fairly straight-forward:

- the content of the variable `v` is transferred to the area divider, and

- the four handles for the generated sub-areas are forwarded, respectively, to the four (recursive) instances of **Warnock** <sup>6</sup>.

This series of pipelines are the ones which realize the recursive step.

The rest of the manifold **Warnock** makes sure that the processes are terminated properly. A separate **variable** (**n**) is used to store the (constant) value of 4. The top-level instance of **Warnock** waits for all of its “children” to deactivate before it deactivates itself. This is done by the combination of the states labeled **wait\_to\_die** and **terminate**. The basic idea is that each instance of the **Warnock** manifold sends a deactivation request to its parent before its own deactivation (see the state labeled **end**). This deactivation request is turned by the **MANIFOLD** system into a system event called **terminate** on the receiver’s side; the particularity of this event is that it can always be caught in a manifold, irrespective of the visibility of its originator. This is exactly what the manifold **Warnock** does: it catches the event and checks against its counter to see if all of its children processes are deactivated before it terminates itself. The **if** statement used for this purpose is, in fact, a manner, with the obvious meaning and is part of the “standard” **MANIFOLD** environment.

Note that there are two blocks in **Warnock** with the same label **terminate**. The reason is to avoid a race condition which can happen in the block for **subdivide**. Indeed, it is perfectly possible that **divide\_area** is still busy calculating, e.g., the fourth sub-area while the **Warnock** instance for, say, the first sub-area already terminates. Obviously, **Warnock** must *not* (yet) change state but it must not ignore the event either (otherwise a non-termination will occur). By putting a separate block for **terminate** with the statement **save** we make sure that the event is neither lost nor preempts the state **subdivide**.

If no subdivision is necessary, **Warnock** makes a state transition to the block labeled **done**, which does an immediate state transition again. This, finally, leads to the termination of the manifold. Strictly speaking, it is not necessary to have a separate intermediary state in this case (a block may have multiple labels). However, when our example is extended further in the next sections, having a separate state will prove to be beneficial.

## 5.2 Analysis of the Program

Warnock’s algorithm is an example of the *image space algorithms* in computer graphics. These algorithms are primarily concerned with images and compute the attributes of each pixel on the screen. Resolution of the relationships among objects in a scene becomes a secondary concern. On the other hand, *object space algorithms* are concerned with the properties of and relationships among the objects in a scene and compute an image only after these relationships

---

<sup>6</sup>The use of the term *recursive* is perhaps somewhat misleading here. Contrary to its common connotations in other programming languages, there is no implied “wait for return or death of your child” process in **MANIFOLD**. This means that a parent process can terminate (and have its resources deallocated) as soon as it spins off its (recursively created) children, if there is no functional requirement for it to wait for their results.



are determined. Warnock's algorithm is not very much in use today. Indeed, if the hidden surface removal is to be performed in image space, availability of powerful hardware makes other methods (primarily, the so called Z-buffer method) more attractive. Whether or not this preference will persist in the future is a matter of debate and its details are far beyond the scope of this paper.

Nevertheless, Warnock's algorithms is still of interest, because it is a very simple example of a general principle which seems to be extremely popular both in computer graphics and in image processing. This principle is what we might call *recursive subdivision*. The idea is the extremely simple, albeit very powerful, concept of divide and conquer: if a problem cannot be solved at a given level, the underlying model is somehow divided and the same algorithm is used recursively on the results of the division. If the subdivision of the problem is chosen appropriately, the problem becomes more easily solvable for each of the results of the subdivision. Interestingly, with a properly chosen subdivision scheme, such algorithms are sometimes readily adaptable for parallel hardware.

Although, obviously, the principle of recursive subdivision is not restricted to computer graphics, its popularity within the computer graphics community seems to be related to the special nature of the field. Indeed, the geometric nature of the underlying problems often gives very clear clues for how to perform the subdivisions and how to control its recursion in an optimal way. Thus, the application of recursive subdivision is very natural in working with synthetic or digital images. Apart from Warnock's algorithm for removal of hidden surfaces, similar or more elaborate approaches can be used in calculating and/or displaying spline curves or surfaces [33], perform calculations on CSG<sup>7</sup> objects using quadtrees [32], digital filtering of images, global histogramming of digital images [37], parallelizing such time consuming rendering procedures as ray tracing [35] especially on CSG objects, performing the calculations necessary to visualize volumes [38], etc.

What is the role of **MANIFOLD** in this respect? Looking at the program on Listing 2, it is clear that **MANIFOLD** has a real expressive power in describing the skeleton of a recursive subdivision algorithm. Note that the atomic processes used by the program are defined in a fairly abstract way; any atomic process, abiding to these specifications, can be "plugged in" the same **MANIFOLD** program to serve a different application. Although most of the algorithms listed above require a more sophisticated version of the algorithm (and we will elaborate on these improvements in the following sections), we believe the listing commented in detail in §5.1.1 makes the essential point: that using **MANIFOLD** it is possible to describe in a very concise and declarative form, the primary communication skeleton of a certain class of systems or algorithms without bothering with their computational details.

These examples also reveal another general and more important characteristic: most of the algorithms cited above were, originally, *not* meant for parallel

---

<sup>7</sup>Constructive Solid Geometry



hardware. Instead, the recursive subdivision approach made the problems at hand just (more) easily solvable and manageable; it was the expressive power of “parallelism” and not performance gains per se, that was important here. It is almost a “by-product” that some of these algorithms are good candidates for true parallelism. We use the term “some” because it is not even certain that all these algorithms run much more efficiently on a true, massively parallel hardware, than on a conventional sequential machine. There may be a trade-off between the obvious gains of parallelism and other considerations (e.g., bulk data access).

Nevertheless, **MANIFOLD** is useful for expressing the communications and control structure of these algorithms, even if the actual implementation of a **MANIFOLD** system may run only on a conventional single-processor computer supporting simulated parallelism only (as in the case of our first experimental implementation based on Concurrent C++). This seems to be a clear case of a more general principle: it may be extremely beneficial to use mental models which use concurrency, communication, and coordination, as natural paradigms to grasp the essence of a problem and/or of an algorithm. Concurrency need not be considered a “necessary curse,” as perceived by a large number of practitioners. On the contrary, it is often very helpful in conceptual simplification of the problem at hand. Gelernter and Carriero ([4]) stress that:

... in principle you can use the *same* coordination language that you rely on for parallel applications programming when you develop distributed systems. You can use the same model in building ... a file system.

We agree both with this statement, and with their implied position that the same language can also be used to describe systems and problems at large, that will not necessarily end up running in a parallel or distributed environment. We believe that as a coordination language, **MANIFOLD** is useful towards these ends.

### 5.3 *Improvements to the Program*

In this section we present enhancements to the **MANIFOLD** program described in §5.1 and evolve a better framework for expressing different versions of the adaptive recursive algorithms mentioned above. The improvement to the program is done in two steps. First, the restriction of a fixed number of subdivisions is relaxed. Second, we allow the possibility of backward control in the recursive processes; i.e., allow a parent to wait for and use the results produced by its children.

#### 5.3.1 *Variable Number of Subdivisions*

The program in §5.1 has an obvious restriction that may make it inappropriate for general use in other applications. This program has a “hardwired” subdivision feature: each area must be subdivided into exactly four sub-areas.

Although this is natural in the case of Warnock's algorithm, and it is trivial to change the number four, imposing any fixed number by itself is a constraint that hinders more general usability of this program for other applications. In particular, a more general class of recursive subdivision algorithms use an adaptive subdivision scheme wherein the number of subdivisions at each level of recursion, as well as the subdivision boundaries, may depend on the data and thus cannot be predetermined.

In this section, we present an improvement to the **MANIFOLD** program of §5.1 that allows the number of subdivisions to be determined dynamically at each level. To put our revised **MANIFOLD** program in the right perspective, we remark that a later version of Warnock's algorithm, called the Weiler-Atherton algorithm (see [30]), subdivides the screen along polygon boundaries, rather than along the two mid-lines of the screen. Clearly, the Weiler-Atherton algorithm requires a variable number of subdivisions.

The revised **MANIFOLD** program now consists of two parts: the one in Listing 3 and the one in Listing 4. The first part is, in fact, a somewhat simplified version of the program in Listing 2. We have changed the specification of the **DivideArea** process: what we require now is that when **DivideArea** receives an area handle, it produces a series of area handles (one for each sub-area) on its standard output and then terminates.

The recursive step is now hidden into a separate manifold process, called **Distribute**. This program appears in Listing 4 and will be explained later. As far as the manifold **Warnock**<sup>8</sup> is concerned, **Distribute** receives the area handles for this level's sub-areas on its standard input and, somehow, takes care of the recursion. A separate pipeline is set up in the block labeled **subdivide** to send these handles to a local instance of **Distribute**. Note that now it is **Distribute** that is responsible for proper termination; consequently, the counter **n** has disappeared from **Warnock**.

As a commentary on **MANIFOLD** programming, note the difference between the two pipelines:

$$v \rightarrow \text{divide\_area}, \text{divide\_area} \rightarrow \text{distribute}$$

that appear as separate group members in the state **subdivide**, and the somewhat similar single pipeline:

$$v \rightarrow \text{divide\_area} \rightarrow \text{distribute}$$

that may be mistaken as their equivalent. While the two alternatives work the same as long as the flow of units are concerned, they indeed behave quite differently on termination. In **MANIFOLD**, a pipeline breaks up as soon as any one of its processes terminates or raises a special event **break**. In case of our single pipeline, this can happen as soon as the process **v** has delivered its value,

---

<sup>8</sup>By now "Warnock" is a misnomer for this program and "Weiler-Atherton" is probably a better name. However, we prefer to keep the name "Warnock" to preserve the similarity with the previous **MANIFOLD** program, for pedagogical reasons.

```

TestAndColor() import.
DivideArea() import.
Distribute() import.

Warnock()
{
    process test_and_color is TestAndColor.
    process v is variable.
    process divide_area is DivideArea.
    process distribute is Distribute.

    start:
        ( activate v,
          activate test_and_color,
          input  $\rightarrow$  ( $\rightarrow$  test_and_color, $\rightarrow$  v),
        ).
    subdivide:
        ( activate divide_area,
          activate distribute,
          v  $\rightarrow$  divide_area,
          divide_area  $\rightarrow$  distribute
        );
    do end.
    done:
    do end.
    end:
    deactivate parent.
}

```

Listing 3. Program with variable area subdivision; part I.

```

Distribute()
{
    port      in internal.
    process n  is variable.

    start:
        ( activate n, n = 0 ); do main_cycle.
    main_cycle:
        getunit(input) → internal;
        do next_area.
    next_area:
        ( n = n + 1, getunit(internal) → Warnock );
        do main_cycle.
    terminate:
        save.
    disconnected.input: wait_for_death:
        void.
    terminate:
        n = n - 1;
        if( n == 0, do end, do wait_for_death ).
    end: .
}

```

Listing 4. Program with variable area subdivision; part II.

which can result in the breakup of the connection between `divide_area` and `distribute`, if they are all in the same pipeline. Having them in two separate pipelines in a group, as in the state `subdivide` in Listing 3, ensures that such premature breakups will not happen. (In `MANIFOLD`, a group terminates when all of its members are broken up.)

A number of constructs used in the original Warnock program (Listing 2) now appear in `Distribute` (see Listing 4). Using the counter `n` to count the number of activated child processes, as well as handling of their deactivations, are exactly the same as before. The primary difference is, of course, in the handling of a variable number of incoming units.

The `Distribute` manifold uses the built-in pseudo-process<sup>9</sup> `getunit` which acts as follows:

- it is suspended on a port of the caller, as long as there is no unit available for delivery on the port;
- when a unit is or becomes available, this unit is sent out onto the output port of `getunit` and the pseudo-process terminates (*i.e.*, the pipelines in

---

<sup>9</sup>By *pseudo-process* we mean one of the primitive actions of `MANIFOLD` that behave like a real process in a pipeline, although they are not truly separate processes.

which it is involved are broken);

- if there is no unit available for delivery on the port *and* there is no external process connected to that port, `getunit` is not only suspended, but it also raises the `disconnected` event (with the selected port as the source of the event).

The `Distribute` manifold takes advantage of these features of `getunit`. In the block labeled `main_cycle` (which, except for activation of the counter is the effective starting block of `Distribute`), a pipeline is set up using `getunit` with its output connected to another (externally non-visible) port of `Distribute`. The role of this pipeline is twofold:

1. When a unit arrives (actually, an area handle from the `DivideArea` process, although `Distribute` does not know the origin of the unit), it is picked and put into the `internal` port. Next, an internal state transition is made which results in the activation of a new instance of `Warnock`.
2. When there is no unit in the buffer of the `input` port of `Distribute`, *and* this port is no longer connected to any other port (which means that the connecting `DivideArea` process has terminated), `getunit` raises a `disconnected` event (which results in the preemption of the current state).

The rest is relatively clear: the unit stored in the `internal` port is picked by another instance of `getunit`, which passes it to an (implicitly activated) instance of `Warnock`, and the manifold returns to its waiting state in `main_cycle`.

It may not be immediately obvious why we use a separate state (`next_area`) to activate a new instance of `Warnock`. Indeed, merging the two states `main_cycle` and `next_area` is possible and also alleviates the need for the port `internal`, since we can use the pipeline

$$\text{getunit}(\text{input}) \rightarrow \text{Warnock}$$

in the block labeled `main_cycle`. However, the advantage of having two separate states instead of one is that we avoid an unnecessary activation of yet another instance of `Warnock` in each recursion. Using two distinct states, we can be sure that `Warnock` is activated if and only if there *is* another area handle in the `internal` port of `Distribute`.

### 5.3.2 Handling Return Values

The algorithms that can use the `MANIFOLD` programs in §5.1.1 and §5.3.1 are constrained by another limitation. Once the recursive branches of the algorithm start off, they do not communicate with their parents any more (or, to be precise, they have no communication expressed by the `MANIFOLD` program). This is fine (indeed, desirable) with the original Warnock's algorithm: the sub-areas of a screen can be filled independently of one another, and a parent has

```

Permanent(inp,outp)
  port out inp.
  port in outp.
{
  start:
    inp → outp.
}

```

```

Permanent(middle,second)
  process middle.
  process second.
{
  start:
    input → middle → second.
}

```

Listing 5. Programs to set up permanent pipelines.

no reason to stay alive and take up resources once its children are started. However, this is obviously inappropriate in a number of other applications.

Once again, a slight improvement on Warnock’s algorithm serves as a good motivating example. In §5.1.1 we assumed that the recursion stops when the size of an area reaches the size of a pixel. Strictly speaking, this assumption is true, but it results in aliasing problems (*i.e.*, the appearance of “staircase” polygon edges and unpleasant color transitions). One of the anti-aliasing methods which can be easily used with Warnock’s algorithm requires the recursion to go on at least one more step, to the level of sub-pixels. The color properties computed at sub-pixel levels are then returned to the pixel level routines, which in turn average them out to calculate the color of their pixels.

To use **MANIFOLD** for such an algorithm implies that (at least between the pixel and sub-pixel levels) each recursive branch must compute and return a value to its parent, and each parent must wait for the returned result of all of its children before it can complete its function and terminate. In this section, we modify our **MANIFOLD** programs to accommodate returned values.

Listings 6 and 7 show the new version of our **MANIFOLD** program; they correspond to the Listings 3 and 4, respectively. As in the previous section, we only highlight the differences between the old and the new versions in this section.

The specification of the atomic process **TestAndColor** is now slightly different. Representing the “bottom” of the recursion, this atomic process is also required to return a value to be forwarded to the upper level (e.g., the color value, in the anti-aliasing example). Additionally, a new process, called **Merge**, is defined: this process receives “values” on its standard input port and

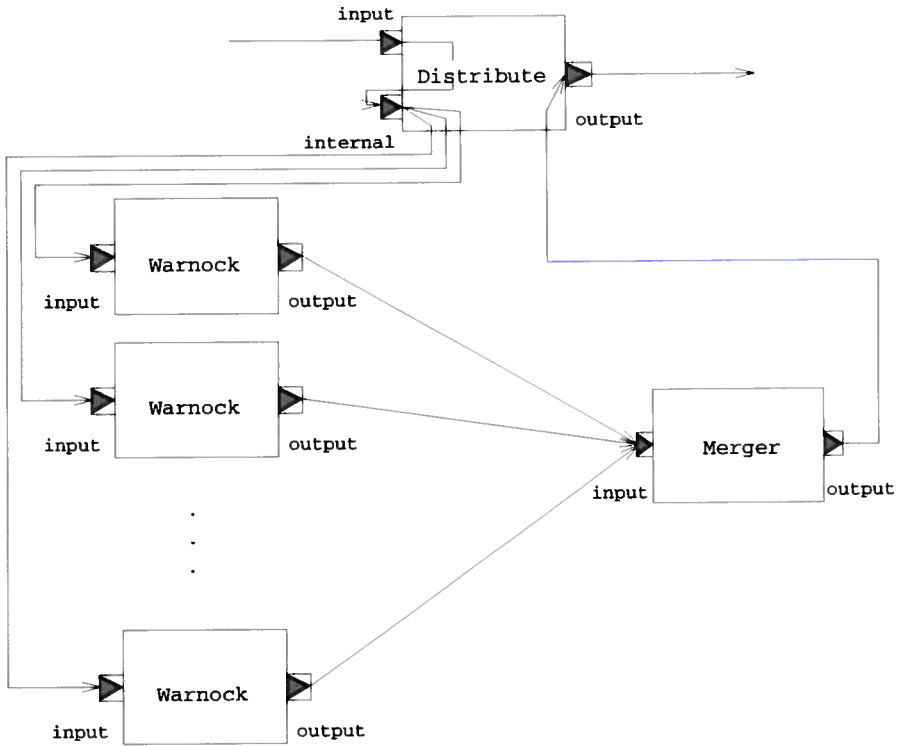


FIGURE 4. A pictorial representation of the manifold **Distribute**.

“merges” them into one value delivered on its output port (in our anti-aliasing example, this process calculates the average of color values it receives)<sup>10</sup>. What **Merge** does is to read an unknown number of units from its standard input, compute their “merged” result (e.g., their average), write it out to its standard output, and terminate. It detects the equivalent of an end-of-file on its standard input (if it is in fact an atomic process), or reacts to a **disconnected** event (if it is another manifold), to realize that it has received all input units it is expected to process.

With these definitions in mind, the differences between the new and the old version of **Warnock** are not too difficult to understand. In the **start** block, the pipeline contains an additional item, which stores the output of **test\_and\_color** in a local port. Also, the new version of **Distribute** is expected to have an output, too, which is redirected to the output port of **Warnock**. Finally, the state labeled **done** is no longer only a state transition; it first reads the value produced previously by the bottom of the recursion and

<sup>10</sup>Note that in Listing 6, the declaration of **Merge** does *not* specify whether it is an atomic process or yet another manifold. It simply states that its declaration is contained in a separate **MANIFOLD** source file, and will be available at link time.



transfers it to the output port. Apart from these differences, the new version of the **Warnock** manifold has an identical behavior to the previous one.

The new version of **Distribute** uses two small manifolds of Listing 5 which are usually part of the standard **MANIFOLD** environment. The meaning of these two manifolds is clear: they set up pipelines which remain unbroken as long as members of the pipeline are active. Remember that, according to the specification of **MANIFOLD**, if a manifold leaves a state, all pipelines set up in that state are broken before leaving. The use of the **Permanent** manifolds is to avoid this breakup.

**Distribute** now sets up a slightly more complicated network of connections. Figure 4 is a pictorial representation of these connections. In the startup state of **Distribute**, a permanent connection (using the first version of **Permanent** in Listing 5) is set up from the output port of **merge** (an instance of **Merge**) to the output port of the running instance of **Distribute**. Note that this is a perfectly legitimate setup: ports of a process instance (e.g., **merge**) can be connected in pipelines even before the process is activated. Additionally, another pseudo-process, **guard**, is activated. The role of this pseudo-process is to raise an event (named in its argument) if a unit appears on its designated port.

The pipelines set up in the state **next\_area** are slightly different: the connection between each new instance of **Warnock** and **merge** is set up using **Permanent**, to prevent its breakup in case of a state transition. This is where the second version of **Permanent** is used (note that the different signatures of the two **Permanent** manifolds disambiguates the choice).

The two events **disconnected.input** and **wait\_for\_death** are now handled by two distinct states. The state labeled **wait\_for\_death** is the same as before: it is used to wait to receive the right number of **terminate** events before dying. The new state for **disconnected.input** activates **merge** and then makes a transition to **wait\_for\_death**.

There is a subtlety about **merge** that needs more explanation here. Our specification of **Merge** states that it receives an unknown number of input units, and detects the equivalent of an end-of-file to know they have been exhausted. Thus, we must make sure that at least all connections between **merge** and its suppliers are established before it is activated. This is why we connect all instances of **Warnock** to **merge** before arriving at **disconnected.input** where we activate it.

Before terminating, **Distribute** must not only wait for all of its local instances of **Warnock** to terminate, but it must also make sure that the output value of **merge** has actually arrived and is transferred out of its output port. This is done by the event **output\_arrived** which is raised by **guard**. Note the use of the **save** action for this event; its role is the same as for the event **terminate**, as explained earlier.

## 6 RELATED WORK

The general concerns which led to the design of **MANIFOLD** are not new. The CODE system [39, 40] provides a means to define dependency graphs on sequential programs. The programs can be written in a general purpose programming language like Fortran or Ada. The translator of the CODE system translates dependency graph specifications into the underlying parallel computation structures. In the case of Ada, for example, these are the language constructs for rendezvous. In the case of languages like Fortran or C, some suitable language extensions are necessary. Just as in traditional dataflow models, the dependency graph in the CODE system is static.

The **MANIFOLD** streams that interconnect individual processes into a network of cooperating concurrent active agents are somewhat similar to links in dataflow networks. However, there are several important differences between **MANIFOLD** and dataflow systems. First, dataflow systems are usually fine-grained (see for example Veen [41] or Herath et. al [42] for an overview of the traditional dataflow models). The **MANIFOLD** model, on the other hand, is essentially oblivious to the granularity level of the parallelism, although the **MANIFOLD** system is mainly intended for coarser-grained parallelism than in the case of traditional dataflow. Thus, in contrast to most dataflow systems where each node in the network performs roughly the equivalent of an assembly level instruction, the computational power of a node in a **MANIFOLD** network is much higher: it is the equivalent of an arbitrary process. In this respect, there is a stronger resemblance between **MANIFOLD** and such higher level dataflow environments like the so called Task Level Dataflow Language (TDFL) of Suhler et al. [43].

Second, the dataflow-like control through the flow of information in the network of streams is not the only control mechanism in **MANIFOLD**. Orthogonal to the mechanism of streams, **MANIFOLD** contains an event driven paradigm. State transitions caused by a manifold's observing occurrences of events in its environment, dynamically change the network of a running program. This seems to provide a very useful complement to the dataflow-like control mechanism inherent in **MANIFOLD** streams.

Third, dataflow programs usually have no means of reorganizing their network at run time. Conceptually, the abstract dataflow machine is fed with a given network only once at initialization time, prior to the program execution. This network must then represent the connections graph of the program throughout its execution life. This lack of dynamism together with the fine granularity of the parallelism cause serious problems when dataflow is used in realistic applications. As an example, one of the authors of this paper participated in one of the very rare practical projects where dataflow programming was used in a computer graphics application [44]. This experience shows that the time required for the effective programming of the dataflow hardware (almost 1 year in this case) was not commensurate with the rather simple functionality of the implemented graphics algorithms.

The previously mentioned TDFL model [43] changes the traditional dataflow

model by adding the possibility to use high level sequential programs as computational nodes, and also a means for dynamic modification of the connections graph of a running program. However, the equivalent of the event driven control mechanism of **MANIFOLD** does not exist in TDFL. Furthermore, the programming language available for defining individual manifolds seems to be incomparably richer than the possibilities offered in TDFL.

Following a very different mental path, the authors of LINDA [5, 6] were also clearly concerned with coordination of communications and the reusability of existing software. LINDA uses a so called generative communication model, based on a *tuple space*. The tuple space of LINDA is a centrally managed space which contains all pieces of information that processes want to communicate. A process in LINDA is a black box. The tuple space exists outside of these black boxes which, effectively, do the real computing. LINDA processes can be written in any language. The semantics of the tuples is independent of the underlying programming language used. As such, LINDA supports reusability of existing software as components in a parallel system, much like **MANIFOLD**.

Instead of designing a separate language for defining processes, the authors of LINDA have chosen to provide language extensions for a number of different existing programming languages. This is necessary in LINDA because seemingly, its model of communication (i.e., its tuple space and the operations defined for it) is not intended to express computation of a general nature by itself. The LINDA language extensions on one hand place certain communication concerns inside of the “black box” processes. On the other hand, there is no way for a process in LINDA to influence other processes in its environment directly. Communication is restricted to the information contained in the tuples, voluntarily placed into and picked up from the tuple space. We believe a mechanism for direct influence (but not necessarily direct control), such as the event driven control in **MANIFOLD** is desirable in parallel programming.

One of the best known paradigms for organizing a set of sequential processes into a parallel system is the Communicating Sequential Processes model formalized by Hoare [2, 3] which served also as a basis for the development of the language Occam [12]. Clearly not a programming language by itself, CSP is a very general model which has been used as the foundation of many parallel systems. Sequential processes in CSP are abstract entities that can communicate with each other via pipes and events as well. CSP is a powerful model for describing the behavior of concurrent systems. However, it lacks some useful properties for constructing real systems. For example, there is no way in CSP to dynamically change the communications patterns of a running parallel system, unless such changes are hard-coded inside the communicating processes. The communications between a process and its environment are an integral part of its semantics in CSP. Occam inherits both of these characteristics from CSP. In contrast, **MANIFOLD** clearly separates the functionality of a process from the concerns about its communication with its environment, placing the latter entirely outside of the process itself. The responsibility for establishing and managing the interactions among processes in a parallel system is completely

taken over by manifolds. A manifold orchestrates the interactions among a set of processes (some of which may be other manifolds) without their knowledge.

Another significant difference between CSP (and Occam) and **MANIFOLD** is that all communication in CSP is synchronous, whereas everything (including events) in **MANIFOLD** are asynchronous. Furthermore, the data-flow-like means of communication and its associated control mechanisms are deemed especially important in **MANIFOLD**, for which it has first class support through special language constructs.

An important distinction between **MANIFOLD** and many other systems (e.g., Occam) is that they generally fix the number of processes, the topology of the communication network, and the potential connectivity of each individual process at compile time. **MANIFOLD** processes, on the other hand, do not know who they are connected to, can be created dynamically, and can be dynamically connected/disconnected to/from other processes while they are running.

An ISO standard for open systems interconnection is the language LOTOS (Language Of Temporal Ordering Specification)[45, 46, 47]. It is a formal description technique based on the temporal ordering of observable behavior of concurrent processes. The LOTOS language is based on a concurrency model of parallelism described by Milner, called CCS (see [1]). (CCS is similar in its flavor to CSP, although there are significant differences between them.) The atomic form of interaction in LOTOS is through events which, as in CSP, synchronize their participating processes. The behavior of a process in LOTOS is described in *behavior expressions* that are composed of simpler behaviors using sequential and choice operators. LOTOS includes many other language constructs, e.g., to support abstract data types. Nevertheless, its view of parallelism is essentially the same as CSP.

As mentioned in §2, the complexity of using languages like Ada, Occam, and Concurrent C++ can become overwhelming in highly parallel applications that require dynamically changing communication patterns. The **MANIFOLD** environment offers an abstraction of the necessary communication facilities which can then be built on top of a distributed programming language like Concurrent C++, or Ada.

## 7 DIRECTIONS FOR FURTHER WORK

More experience is needed with a fully operational **MANIFOLD** system to evaluate its potentials and the adequacy of its constructs in real, practical applications. Nevertheless, it is already clear that certain changes and extensions to the **MANIFOLD** language can have a positive impact on its use in large and complex systems. Several such improvements are currently in our list, of which we mention only a few major ones here.

For instance, the notion of *derived manifolds* may be a useful extension to the language. This concept leads to a hierarchy of manifold definitions with inheritance, analogous to the class hierarchies in object oriented languages. Language support for such syntactic conveniences seem to be quite useful in



large software developments.

An issue that we have encountered a few times in our examples is a need for *directed events*. Strictly speaking, the concept of event in the **MANIFOLD** model is, of course, contrary to the notion of *directed events*, because **MANIFOLD** events are broadcast and can be picked up by any process in the environment. We do not yet know how important the need for *directed events* is, because we have been able to do without them so far. Nevertheless, the effect of *directed events* can be supported at the language level in **MANIFOLD** by introducing proper constructs to explicitly control the observability of event sources and/or the preemption sets of manifolds. Observability and preemption sets are both defined implicitly in the current **MANIFOLD** language: they are derived by the compiler from the source code. Symmetric to the way in which a third party process can define streams between two other processes in the current **MANIFOLD** language, new language constructs can allow processes to define and modify observability and/or preemption sets.

## 8 CONCLUSIONS

This paper is an overview of the **MANIFOLD** system and sketches the highlights of its implementation. More experience is still necessary to thoroughly evaluate the practical usefulness of **MANIFOLD**. However, our experience so far indicates that **MANIFOLD** is well suited for describing complex systems of cooperating parallel processes.

**MANIFOLD** uses the concepts of modern programming languages to describe and manage connections among a set of independent processes. The unique blend of event driven and data driven styles of programming, together with the dynamic connection graph of streams seem to provide a promising paradigm for parallel programming. The emphasis of **MANIFOLD** is on orchestration of the interactions among a set of autonomous *expert* agents, each providing a well-defined segregated piece of functionality, into an integrated parallel system for accomplishing a larger task. The declarative nature of the **MANIFOLD** language and the **MANIFOLD** model's separation of communication and coordination from functionality and coordination, both significantly contribute to simplify programming of large, complex parallel systems.

In the **MANIFOLD** model, each process is responsible to *protect* itself from its environment, if necessary. This shift of responsibility from the producer side to the consumer of information seems to be a crucial necessity in open systems, and contributes to reusability of modules in general. This model imposes only a "loose" connection between an individual process and its environment: the producer of a piece of information is not concerned with who its consumer is. In contrast to systems wherein most, if not all, information exchange takes place through targeted send operations within the producer processes, processes in **MANIFOLD** are not "hard-wired" to other processes in their environment. The lack of such strong assumptions about their operating environment makes **MANIFOLD** processes more reusable.

The recursive algorithms as well as the example related to the IRIS Explorer system, described in **MANIFOLD**, are only small-scale albeit important practical examples for the usage of **MANIFOLD**. However, **MANIFOLD** can be used to implement more complex interactions, e.g., in a user interface toolkit, as well. For example, in a separate paper, [25], we describe an implementation of the GKS logical input device in **MANIFOLD**.

In our view, massive parallel systems and the current trend in computer technology toward *computing farms* open new horizons for large applications and present new challenges for software technology. Classical views of parallelism in programming languages that are based on extensions of the sequential programming paradigm are ill-suited to meet this challenge. We also believe that it is counter-productive to base programming paradigms for computing farms and massively parallel systems solely on strictly synchronous communication. Many of the ideas underlying the **MANIFOLD** system, if not the present **MANIFOLD** language itself, seem promising towards this goal.

#### ACKNOWLEDGMENT

We are thankful for the direct and indirect contributions of all members of the **MANIFOLD** group at CWI. In particular, Paul ten Hagen inspired some of the original concerns and motivation for **MANIFOLD**. Kees Blom helped to refine the formal syntax for the **MANIFOLD** language and produced its first compiler. Eric Rutten is developing the formal semantics of **MANIFOLD**. Dirk Soede's exercises in **MANIFOLD**, his ongoing work with Anco Smit on a visual interface to **MANIFOLD**, and especially Freek Burger's programming work on the **MANIFOLD** run-time system are also much appreciated.

Last, but not least, we thank the comments of our paper's anonymous referees. We took the liberty of paraphrasing some of the comments made by one referee in our revised conclusion. The same referee also encouraged us to change our original example of a window manager. Motivated by his suggestion, we worked out the present set of examples, which we believe show the concepts and relevance of **MANIFOLD** much better.

#### REFERENCES

1. R. Milner, *Communication and Concurrency*. Prentice Hall International Series in Computer Science, New Jersey: Prentice Hall, 1989.
2. C. Hoare, "Communicating sequential processes," *Communications of the ACM*, vol. 21, August 1978.
3. C. Hoare, *Communicating Sequential Processes*. Prentice Hall International Series in Computer Science, New Jersey: Prentice-Hall, 1985.
4. D. Gelernter and N. Carriero, "Coordination languages and their significance," *Communication of the ACM*, vol. 35, pp. 97–107, February 1992.
5. N. Carriero and D. Gelernter, "LINDA in context," *Communications of the ACM*, vol. 32, pp. 444–458, 1989.

6. W. Leler, "LINDA meets UNIX," *IEEE Computer*, vol. 23, pp. 43–54, February 1990.
7. F. Arbab, "Specification of manifold," Tech. Rep. to appear, Centrum voor Wiskunde en Informatica, Amsterdam, 1992.
8. United States Department of Defense, *Reference Manual for the Ada Programming Language*, November 1980.
9. N. Gehani and W. Roome, "Concurrent C," *Software — Practice and Experience*, vol. 16, pp. 821–844, 1986.
10. N. Gehani and W. Roome, *The Concurrent C Programming Language*. Summit NJ: Silicon Press, 1989.
11. N. Gehani and W. Roome, "Concurrent C++: Concurrent programming with class(es)," *Software — Practice and Experience*, vol. 18, pp. 1157–1177, 1988.
12. INMOS Ltd., *OCCAM 2, Reference Manual*. Series in Computer Science, London — Sydney — Toronto — New Delhi — Tokyo: Prentice-Hall, 1988.
13. H. Bal, J. Steiner, and A. Tanenbaum, "Programming languages for distributed computing systems," *ACM Computing Surveys*, vol. 21, pp. 261–322, 1989.
14. W. Roberts, M. Slater, K. Drake, A. Simmins, A. Davidson, and P. Williams, "First impression of NeWS," *Computer Graphics Forum*, vol. 7, pp. 39–58, 1988.
15. T. Doeppner Jr., "A threads tutorial," Tech. Rep. CS-87-06, Brown University, 1988.
16. P. Buhr and R. Strooboscher, "The  $\mu$ System: Providing light-weight concurrency on shared-memory multiprocessor computers running UNIX," *Software — Practice and Experience*, vol. 20, pp. 929–964, 1990.
17. M. Accetta, R. Baron, W. Bolosky, D. Golub, R. Rashid, A. Tevanian, and M. Young, "Mach: A new kernel foundation for UNIX development," in *Proceedings of the Summer Usenix Conference*, (Atlanta, GA), July 1986.
18. D. Black, "Scheduling support for concurrency and parallelism in the Mach operating system," *IEEE Computer*, vol. 23, pp. 35–43, May 1990.
19. SUN Microsystems, *SunOS Manuals, Lightweight Processes*, revision A ed., 1990.
20. F. Arbab and I. Herman, "Examples in Manifold," Tech. Rep. CS-R9066, Centrum voor Wiskunde en Informatica, Amsterdam, 1990.
21. F. Arbab and I. Herman, "Manifold: A language for specification of inter-process communication," in *Proceedings of the EurOpen Autumn Conference* (A. Finlay, ed.), (Budapest), pp. 127–144, September 1991.
22. F. Arbab, I. Herman, and P. Spilling, "Interaction management of a window manager in Manifold," in *Computing and Information ICCI'92* (W. Koczkodaj, P. Lauer, and A. Toptsis, eds.), (Toronto), IEEE Press, June 1992.
23. I. Herman and F. Arbab, "More examples in examples in Manifold," Tech. Rep. CS-R9214, Centrum voor Wiskunde en Informatica, Amsterdam, 1992.
24. H. Schouten and P. ten Hagen, "Dialogue cell resource model and basic



- dialogue cells,” *Computer Graphics Forum*, vol. 7, no. 3, pp. 311–322, 1988.
25. D. Soede, F. Arbab, I. Herman, and P. ten Hagen, “The GKS input model in manifold,” *Computer Graphics Forum*, vol. 10, pp. 209–224, September 1991.
  26. J. Peterson, R. Bogart, and S. Thomas, “The Utah Raster Toolkit,” in *Proceedings of the Usenix Workshop on Graphics*, (Monterey, California), 1986.
  27. S. Dyer, “A dataflow toolkit for visualization,” *IEEE Computer Graphics & Applications*, vol. 10, July 1990.
  28. C. Upson, “Scientific visualization environments for the computational sciences,” in *Proceedings of the 34<sup>th</sup> IEEE Computer Society International Conference*, (San Francisco), March 1989.
  29. Silicon Graphics, Inc., “IRIS Explorer user’s guide,” tech. rep., Silicon Graphics, Inc., Mountain View, California, 1991.
  30. J. Foley, A. van Dam, S. Feiner, and J. Hughes, *Computer Graphics — Principles and Practice*. Reading: Addison–Wesley, 1990.
  31. R. Gonzalez and P. Wintz, *Digital Image Processing*. Reading, Massachusetts: Addison–Wesley, 1983.
  32. W. Bronsvort, F. Jansen, and F. Post, “Design and display of solid models,” in *Advances in Computer Graphics VI* (G. Garcia and I. Herman, eds.), Heidelberg: EurographicSeminar Series, Springer Verlag, 1991.
  33. R. Bartels, J. Beatty, and B. Barsky, *An Introduction to Splines for Use in Computer Graphics & Geometric Modelling*. Los Altos, California: Morgan Kaufmann Publishers, Inc., 1987.
  34. M. Cohen and J. Painter, “State of the art in image synthesis,” in *Advances in Computer Graphics VI* (G. Garcia and I. Herman, eds.), Heidelberg: EurographicSeminar Series, Springer Verlag, 1991.
  35. S. Green, *Parallel Processing for Computer Graphics*. Research Monographs in Parallel and Distributed Computing, London: Pitman, 1991.
  36. F. Crow, “Parallel computing for graphics,” in *Advances in Computer Graphics VI* (G. Garcia and I. Herman, eds.), Heidelberg: EurographicSeminar Series, Springer Verlag, 1991.
  37. H. Siegel, J. Armstrong, and D. Watson, “Mapping computer-vision related tasks onto reconfigurable parallel processing systems,” *IEEE Computer*, vol. 25, pp. 54–64, February 1992.
  38. D. Laur and P. Hanrahan, “Hierarchical splatting: Progressive refinement algorithm for volume rendering,” *Computer Graphics (SIGGRAPH’91)*, vol. 25, pp. 285–288, July 1991.
  39. J. Browne, M. Azam, and S. Sobek, “CODE: A unified approach to parallel programming,” *IEEE Software*, pp. 10–18, July 1989.
  40. J. Browne, T. Lee, and J. Werth, “Experimental evaluation of a reusability-oriented parallel programming environment,” *IEEE Transaction on Software Engineering*, vol. 16, pp. 111–120, 1990.
  41. A. Veen, “Dataflow machine architecture,” *ACM Computing Surveys*, vol. 18, pp. 365–396, 1986.

42. J. Herath, N. Saiko, and T. Yuba, "Dataflow computing models, languages and machines for intelligence computations," *IEEE Transactions on Software Engineering*, vol. 14, pp. 1805–1828, 1988.
43. P. Suhler, J. Bitwas, K. Korner, and J. Browne, "TDFL: A task-level dataflow language," *Journal of Parallel and Distributed Computing*, vol. 9, pp. 103–115, 1990.
44. P. ten Hagen, I. Herman, and J. de Vries, "A dataflow graphics workstation," *Computers and Graphics*, vol. 14, pp. 83–93, 1990.
45. International Organization for Standardization, Geneva, *Information Processing Systems — Open Systems Interconnections — LOTOS (Formal Description Technique based on the temporal ordering of observational behaviour) ISO/DIS 8807*, March 1988.
46. T. Bolognesi and E. Brinksma, "Introduction to the ISO specification language LOTOS," *Computer Networks and ISDN Systems*, vol. 14, pp. 25–59, 1986.
47. E. Brinksma, "A tutorial in LOTOS," in *Protocol Specification, Testing, and Verification V* (M. Diaz, ed.), pp. 171–194, Amsterdam: North-Holland, 1986.

```

TestAndColor() import.
DivideArea() import.
Merge() import.
Distribute() import.

Warnock()
{
    process test_and_color is TestAndColor.
    process v is variable.
    process divide_area is DivideArea.
    process distribute is Distribute.
    port in internal.

    start:
        ( activate v,
          activate test_and_color,
          input  $\rightarrow$  ( $\rightarrow$  test_and_color  $\rightarrow$  ,  $\rightarrow$  v)  $\rightarrow$  internal,
        ).
    subdivide:
        ( activate divide_area,
          activate distribute,
          v  $\rightarrow$  divide_area,
          divide_area  $\rightarrow$  distribute,
          distribute  $\rightarrow$  output
        );
    do end.
    done:
        getunit(internal)  $\rightarrow$  output;
    do end.
    end:
        deactivate parent.
}

```

Listing 6. Program with return values I.

```

Distribute()
{
    port          in internal.
    process n     is variable.
    process merge is Merger.

start:
    ( activate n,
      Permanent(merge.output,self.output),
      guard(self.output,output_arrived),
      n = 0
    );
    do main_cycle.
main_cycle:
    getunit(input) → internal;
    do next_area.
next_area:
    (n = n + 1, getunit(internal) → Permanent(Warnock,merge));
    do main_cycle.
terminate:
    save.
disconnected.input:
    ( activate merge, do wait_for_death ).
wait_for_death:
    void.
terminate:
    n = n - 1;
    if( n == 0, do end, do wait_for_death ).
output_arrived:
    save.
end:
    void.
output_arrived: .
}

```

Listing 7. Program with return values II.



# The randomness assumption in word frequency statistics

R. Harald Baayen

*Max Planck Institute for Psycholinguistics, Nijmegen*

## 1 INTRODUCTION

The mathematical and computational tools available for the study of word frequency distributions have become increasingly powerful since Zipf published his seminal studies some 60 years ago (Zipf 1935, 1949). The first frequency counts were obtained manually, either by going through a text and filing new words and updating the frequencies of words already encountered on slips of paper, or by going through (manually compiled) concordances. The first statistician to study word frequency distributions, G. U. Yule, obtained the data for his book on “The statistical study of literary vocabulary” (Yule, 1944) in this way. The first frequency dictionary of Dutch, “De meest voorkomende woorden en woordcombinaties in het Nederlandsch”, was similarly compiled manually by De la Court in 1937.

The first frequency list of Dutch obtained by means of a computer was compiled at the Mathematical Centre in 1965 by van Berckel, Brandt Corstius, Mokken, and van Wijngaarden. By 1967, Kučera and Francis had compiled a corpus of one million wordforms for English, and had published frequency counts and analyses in their famous “Computational Analysis of present-day American English” (Kučera and Francis, 1967). This prompted the construction of a slightly smaller corpus (727000 wordforms) of similar design for Dutch by the ‘Werkgroep Frequentie-Onderzoek Nederlands’, leading to the publication of “Woordfrequenties in geschreven en gesproken Nederlands” (Uit den Boogaart, 1975) and “Spreektaal. Woordfrequenties in Gesproken Nederlands” (de Jong, 1979). The most recent frequency information for Dutch is available in the CELEX lexical database (Burnage 1990), which can be queried on-line in the Netherlands, and of which a version on CD-ROM is also available (Baayen, Piepenbrock and van Rijn, 1993). The frequency counts in the CELEX database, which also contains information on spelling, phonology, morphological structure and syntactic features, are based on a corpus of 42 million wordforms compiled by the Institute for Dutch Lexicology in Leiden.

The transition from printed frequency lists based on relatively small corpora to on-line lexical databases based on corpora of tens of millions of words is accompanied by an ever increasing body of texts available in electronic form. Some collections of texts are made accessible via sophisticated software that enables users to search for words or word collocations. Typically, the matches found are presented with some preceding and following context.



For Dutch, the Institute for Dutch Lexicology (INL) has recently made a corpus of 5 million wordforms available for such on-line queries. Similarly, a Dutch newspaper, 'de Volkskrant', is now available on CD-ROM. The software facilitating access to 'De Volkskrant' and to the INL on-line corpus has as a serious drawback that the user is denied access to the texts themselves. Access to the full text, however, is especially critical for the question addressed in this study, namely the randomness assumption underlying all presently available statistical models for word frequency distributions.

Word frequency models build on the fundamental assumption that word tokens occur randomly in texts. It is clear that for natural language this assumption is too strong. The syntax of natural languages imposes severe constraints on where words can occur. For instance, following the Dutch determiner *de*, adjectives and nouns, but not verbs, are allowed (*de lamp*, *de felle lamp*, \**de schijnt*). Similarly, semantic constraints and principles of discourse organization may severely limit the way in which words occur in texts. This raises the question to what extent the predictions of theoretical models can be relied on, especially since it is known that the interpolated vocabulary size tends to seriously overestimate the observed vocabulary size (Brunet 1978, Hubert and Labbe 1988, Labbe and Hubert 1993). The aim of this paper is to trace the source of this overestimation, and to evaluate its consequences for the application of word frequency models in lexical statistics.

To do so, we need access to complete texts in electronic form. Fortunately, collections of raw electronic texts without limiting software-guided access are available by anonymous ftp. The Oxford Text Archive, at [black.ox.ac.uk](http://black.ox.ac.uk), the Gutenberg Project at [mrcnext.cso.uiuc.edu](http://mrcnext.cso.uiuc.edu), and the Online Book Initiative at [obi.std.com](http://obi.std.com) have brought together large numbers of electronic texts, most of which are in English, ranging from election speeches by Clinton to electronic *Star Trek* novels, and from Milton's 'Paradise Lost' to the Book of Mormon. From the Project Gutenberg, I obtained an electronic copy of *Alice in Wonderland*, by Lewis Carroll, and a copy of *Moby Dick*, by Herman Melville.<sup>1</sup> The Online Book Initiative has recently made available the first complete text of a Dutch novel to come to my attention, *Max Havelaar* by Multatuli, which I have also included in my analyses.

My discussion is structured as follows. In section 2, I introduce some basic expressions for the expectation and variance of the vocabulary size  $V_N$  as a function of the number of word occurrences  $N$  in the sample, and of the frequency spectrum, the number of different word types  $V_N(m)$  with frequency  $m$ , again as a function of  $N$ . In section 3, the randomness assumption is tested by studying the development of the vocabulary in the three texts mentioned above. For each of these texts, it is shown that the observed and expected values diverge significantly for a large range of values of  $N$ . The goal of section 4 is to trace the source of this misfit, which may arise due to syntactic and

---

<sup>1</sup>The header of the electronic version of Melville's *Moby Dick* requires that I mention that this version was prepared by E. F. Tray at the University of Colorado, Boulder, on the basis of the Hendricks House Edition.

semantic constraints operating at the sentence level, to lexical specialization (Brunet 1978, Hubert and Labbe 1988), or to the discourse organization of the text. I will show that it is the way in which discourse is developed over time that gives rise to the misfit. The consequences of these findings are discussed in section 5.

## 2 WORD FREQUENCY MODELS

A text can be viewed as an ordered sequence of occurrences (or tokens) of words

$$(w_1, w_2, w_3, \dots, w_N).$$

Usually, the number  $V$  of distinct words, the so-called word types, in the observed vocabulary

$$(A_1, A_2, A_3, \dots, A_V)$$

is much smaller than the sample size  $N$ , due to the repeated occurrence of many word types. Let  $f_N(A_i)$  denote the frequency with which word type  $A_i$  occurs in a sample of size  $N$ . Expressions for the numbers of different word types occurring for arbitrary sample sizes, as well as expressions for the numbers of different word types occurring with some specified frequency at a given sample size have been available since Good (1953), Kalinin (1965), and Good and Toulmin (1976) (see Chitashvili and Baayen (1993) for a review of word frequency models). In this section I introduce the expressions required for studying in what way the randomness assumption is violated in written texts.

Let  $f_N(A_i) = m$  denote the event that word type  $A_i$  occurs with frequency  $m$  in a sample of  $N$  tokens. The expected total number of such word types,  $E[V_N(m)]$ , is given by

$$\begin{aligned} E[V_N(m)] &= E\left[\sum_i I_{[f_N(A_i)=m]}\right] \\ &= \sum_i \binom{N}{m} p(A_i)^m (1 - p(A_i))^{N-m}. \end{aligned} \quad (1)$$

Note that the assumption that  $f_N(A_i)$  is  $\text{bin}(N, p(A_i))$  distributed implies that the tokens of  $A_i$  occur randomly in the text. The expected overall number of different types in the sample, irrespective of their frequency, follows immediately:

$$\begin{aligned} E[V_N] &= E\left[\sum_{m \geq 1} V_N(m)\right] \\ &= \sum_{m \geq 1} \sum_i \binom{N}{m} p(A_i)^m (1 - p(A_i))^{N-m} \\ &= \sum_i (1 - (1 - p(A_i))^N). \end{aligned} \quad (2)$$

For large  $N$  and small  $p$ , binomial probabilities can be approximated by Poisson probabilities, leading to the simplified expressions

$$\begin{aligned} \mathbb{E}[V_N(m)] &= \sum_i \frac{(\lambda(A_i)N)^m}{m!} e^{-\lambda(A_i)N} \\ \mathbb{E}[V_N] &= \sum_i (1 - e^{-\lambda(A_i)N}). \end{aligned} \quad (3)$$

Conditional on a given frequency spectrum ( $V_N(m), m = 1, 2, \dots$ ), the vocabulary size  $\mathbb{E}[V_M]$  for sample size  $M < N$  equals

$$\begin{aligned} \mathbb{E}[V_M] &= \sum_{i=1}^{V_N} (1 - e^{-\lambda(A_i)M}) \\ &= \sum_{i=1}^{V_N} (1 - e^{-\frac{f_N(A_i)}{N}M}) \\ &= V_N - \sum_{m=1} V_N(m) e^{-\frac{M}{N}m}. \end{aligned} \quad (4)$$

Note that (4) suggests that, under randomness, and conditional on the words appearing in the first  $N$  tokens,  $f_M(A_i)$  can alternatively be viewed as a binomially distributed random variable with parameters  $M/N$  and  $f_N(A_i)$ .

The Poisson approximation is especially useful for obtaining expressions for covariances:

$$\begin{aligned} \text{COV}(V_N(m), V_N(k)) &= \\ &= \text{COV}\left(\sum_{i=1}^S \mathbb{I}_{[f_N(A_i)=m]}, \sum_{j=1}^S \mathbb{I}_{[f_N(A_j)=k]}\right) \\ &= \sum_i \sum_j \mathbb{E}[\{\mathbb{I}_{[f_N(A_i)=m]}\} \cap \{\mathbb{I}_{[f_N(A_j)=k]}\}] \\ &\quad - \sum_i \sum_j \mathbb{E}[\mathbb{I}_{[f_N(A_i)=m]}] \mathbb{E}[\mathbb{I}_{[f_N(A_j)=k]}] \\ &= \sum_{i \neq j} \sum \frac{(\lambda(A_i)N)^m}{m!} \frac{(\lambda(A_j)N)^k}{k!} e^{-\lambda(A_i)N - \lambda(A_j)N} + \\ &\quad \delta_{mk} \mathbb{E}[V_N(m)] - \mathbb{E}[V_N(m)] \mathbb{E}[V_N(k)] \\ &= \sum_i \sum_j \frac{(\lambda(A_i)N)^m}{m!} \frac{(\lambda(A_j)N)^k}{k!} e^{-\lambda(A_i)N - \lambda(A_j)N} - \\ &\quad \sum_i \frac{(\lambda(A_i)2N)^{m+k}}{(m+k)!} \binom{m+k}{m} \frac{1}{2^{m+k}} e^{-2\lambda(A_i)N} + \\ &\quad \delta_{mk} \mathbb{E}[V_N(m)] - \mathbb{E}[V_N(m)] \mathbb{E}[V_N(k)] \end{aligned}$$

$$= \delta_{mk} E[V_N(m)] - \binom{m+k}{m} \frac{1}{2^{m+k}} E[V_{2N}(m+k)]. \quad (5)$$

Let  $S$  denote the number of different word types in the population from which a given text is sampled. Since  $E[V_N] = S - E[V_N(0)]$ ,

$$\text{VAR}(V_N) = \text{VAR}(V_N(0)) = E[V_{2N}] - E[V_N]. \quad (6)$$

### 3 TESTING THE RANDOMNESS ASSUMPTION

The left hand panels of Figure 1 show the characteristic divergence between the observed and expected vocabulary size measured at 40 equally spaced intervals for Lewis Carroll's *Alice in Wonderland* (top), Herman Melville's *Moby Dick* (middle), and Multatuli's *Max Havelaar* (bottom). The type definition used here is a very simple one in which distinct strings represent different types. No morphological preprocessing has been applied. Hence *house* and *houses* are counted as two different types. The expected vocabulary size  $E[V_M]$  was obtained using (4), for each novel conditioning on the frequency spectrum of the complete text.

Note that for all three novels the difference between the expected and observed vocabulary size tends to be substantial for a large range of values of  $M$  ( $M < N$ ). In the case of *Alice in Wonderland*, the expected vocabulary size exceeds the observed vocabulary size for the full range of values of  $M$ . For *Moby Dick* and *Max Havelaar*, this divergence is reversed for large  $M$ , where the expected vocabulary size is smaller than the observed vocabulary size. For the first 20 measurement points, (6) can be used to estimate the variance of  $V_N$ , so that standardized scores  $Z = (V_N - E[V_N])/\sqrt{\text{VAR}[V_N]}$  can be obtained. Measurement points for which  $|Z| > 1.96$  have been highlighted. For the three novels studied here, all Z-scores obtained, except for one text size in Multatuli's *Max Havelaar*, are smaller than  $-1.96$ , suggesting informally that the divergence between the observed and expected growth curves is significant for at least the first half of the text.

### 4 TRACING THE SOURCE OF THE MISFIT

We have seen that the predictions derived from the basic model for word frequency distributions, essentially a simple urn model (without replacement), diverge substantially from the empirical intermediate vocabulary sizes. Instead of rejecting the model as unfit for the study of actual language data, it is useful to study the source of the misfit in some more detail, as this may shed some light on the conditions under which the model might remain valid.

There are three possible sources for the divergence between the empirical and expected vocabulary growth curves. Syntactic and semantic constraints at the level of the sentence are in conflict with the randomness assumption. These constraints might give rise to the observed misfit. Alternatively, it has been claimed that lexical specialization is at issue here. If the use of specialized words is restricted to particular text fragments, as it often appears to be, the

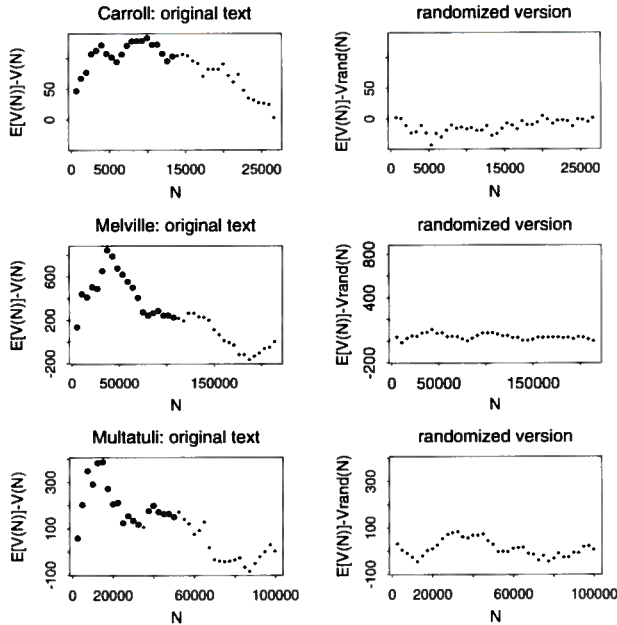


FIGURE 1. The size of the divergence between the empirical and expected vocabulary size  $E[V_M] - V_M$  for 40 equally spaced measurement points for L. Carroll's *Alice in Wonderland*, H. Melville's *Moby Dick*, and Multatuli's *Max Havelaar* (left column), and the size of this difference for a version of the novel in which the order of the sentences but not the order of the words in the sentences was randomized (right column). Significant differences have been highlighted.

uneven, clustered occurrence of the tokens of these types may underlie the misfit. Finally, it might be the case that the discourse organization of the text induces a non-random development of the vocabulary. I will explore these possibilities in turn.

#### 4.1 Syntactic and semantic constraints

In order to trace the possible role of syntactic and semantic constraints, I made artificial versions of the three novels in which the order of the sentences was randomized, while keeping the order of the words in the sentences unchanged. Table 1 summarizes for each text the number of tokens  $N$ , the number of types  $V$ , the number of sentences  $s$  and the mean sentence length  $msl$ . The mean

novel	$N$	$V$	$s$	$m_{sl}$
Carroll	26611	2695	2323	11.45
Melville	213756	16741	16307	13.11
Multatuli	99819	11126	6791	14.69

TABLE 1. Number of tokens  $N$ , number of types  $V$ , number of sentences  $s$  and mean sentence length  $m_{sl}$  for Lewis Carroll’s *Alice in Wonderland*, Herman Melville’s *Moby Dick*, and Multatuli’s *Max Havelaar*.

sentence length ranges between 11 and 15 words per sentence. Given these far from trivially small mean sentence lengths, syntactic and semantic constraints at the sentence level cannot but be operative. If their presence induces the misfit between the observed and expected vocabulary size, the randomized versions of the novels should show a similar pattern as found in the left hand panels of Figure 1.

The right hand panels of Figure 1 plot the results obtained. For all novels, the divergence between the observed and expected vocabulary sizes is substantially reduced. For all measurement points, the Z-score did not reach significance ( $|Z| < 1.96$ ). Moreover, the direction of the difference appears to vary randomly, yielding largely negative scores for *Alice in Wonderland*, generally positive scores for *Moby Dick*, and both negative and positive scores for *Max Havelaar*. These results show that syntactic and semantic constraints at the sentence level can be ruled out as factors responsible for the lack of goodness-of-fit.

#### 4.2 Lexical specialization

It has been argued that lexical specialization is to be held responsible for this lack of goodness-of-fit (Brunet 1978, Labbe and Hubert, 1993). The argument is based on the observation that the curve of  $V_N$  often reflects differences between texts when texts of different authors, or even different texts of the same author are studied jointly. To illustrate this simple observation, I concatenated Carroll’s *Alice in Wonderland*, Baum’s *The Wizard of Oz*, a collection of election speeches by Clinton, and Barrie’s *Peter Pan*.<sup>2</sup> The observed and predicted vocabulary growth curves are shown in Figure 2. A marked discontinuity in the growth curve can be observed at the second vertical line, where the officialese of Clinton’s election speeches succeeds *Alice in Wonderland* and *The Wonderful Wizard of Oz*. The specialized, concentrated use of officialese in the third partition of this artificial text gives rise to both substantial quantitative as well as qualitative differences between the observed and expected growth curves.

In this example, it is evident that the texts have not been sampled from the same population. Different authors will generally tend to use different sets of words. In addition, present-day officialese can hardly be compared with books

<sup>2</sup>The last three texts were obtained from the Project Gutenberg, the Online Book Initiative, and the Oxford Text Archive, respectively.

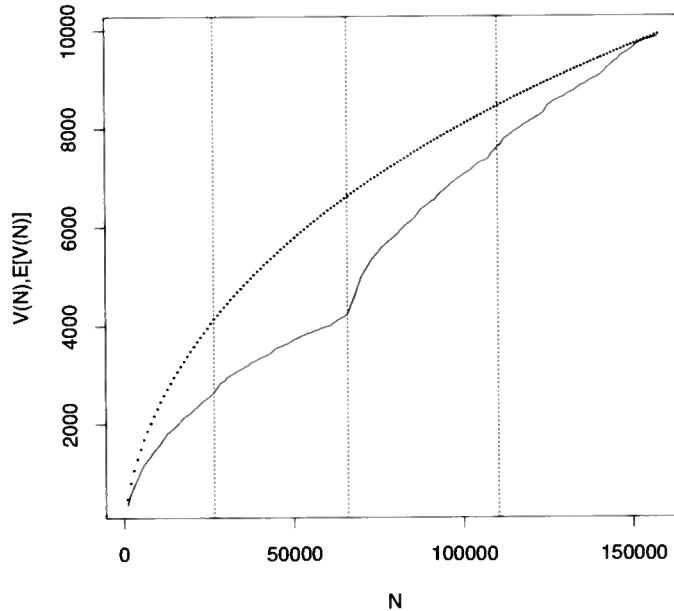


FIGURE 2. Empirical (solid line) and expected (dotted line) growth curves for the concatenated texts of L. Carroll's *Alice in Wonderland*, L. F. Baum's *The Wonderful Wizard of Oz*, election speeches by B. Clinton, and J. M. Barrie's *Peter Pan*, for 160 measurement points. Dotted vertical lines indicate the transition points between texts.

written for children more than 70 years ago. The substantial misfit comes as no surprise. Within a single novel, the effects of lexical specialization will not be as extreme. At first sight, there are two ways in which lexical specialization might violate the randomness assumption. It might be that lexical specialization is characteristic of certain parts of the text, but not for others, as in the above artificial example. Alternatively, lexical specialization, although uniformly distributed in the text, might as such give rise to the misfit between the observed and expected vocabulary size. If lexical specialization leads to local concentration of the tokens of a specialized type, this local concentration might imply that within the relevant text slice tokens that would otherwise have been free to represent additional non-specialized types are now allocated to one specialized type. For text slices with specialized words, this would result in a lower expected value for the vocabulary size.



This idea has been formalized by Hubert and Labbe (1988) and Labbe and Hubert (1993), who present the following modification of (4):

$$E[V_M] = p \frac{M}{N} V_N + (1-p) \left\{ V_N - \sum_m (V_N(m) e^{-\frac{M}{N} m}) \right\}. \quad (7)$$

To obtain (7), assume that all tokens of a specialized word occur jointly in a particular fragment of the text. Also assume that a proportion  $p$  of all  $V_N$  types in the text enjoy specialized use, and that this specialization affects the same proportion  $p$  of the  $V_N(m)$  types for all  $m$ . Finally, assume that the chunks of tokens of the specialized word types ( $S_i, i = 1, 2, \dots, pV_N$ ) appear randomly distributed over the text. If so,

$$\begin{aligned} E[V_M] &= E \left[ \sum_{i=1}^{pV_N} I_{[f_M(S_i) > 0]} + \sum_{i=1}^{(1-p)V_N} I_{[f_M(A_i) > 0]} \right] \\ &= \sum_{i=1}^{pV_N} \frac{M}{N} + \sum_{i=1}^{(1-p)V_N} (1-p) V_N (1 - e^{-\frac{M}{N} f_N(A_i)}) \\ &= \frac{M}{N} p V_N + (1-p) V_N - \sum_m (1-p) V_N(m) e^{-\frac{M}{N} m}. \end{aligned} \quad (8)$$

For  $K$  measurement points ( $M_k, k = 1, 2, \dots, K, M_k < N$ ), Labbe and Hubert (1993) determine  $p$  by minimizing the chi-squared statistic

$$\sum_{k=1}^K \frac{(V_{M,k} - E[V_{M,k}])^2}{E[V_{M,k}]}, \quad (9)$$

conveniently ignoring that the variance of  $E[V_{M,k}]$  increases with  $M$ . In this way, much improved and often excellent fits can be obtained. For instance, for *Alice in Wonderland*, the optimal value of  $p$  for  $K = 40$  equals 0.16, and the fit obtained is a perfect smoothed curve through the observed values of  $V_{M,k}$  ( $\chi^2_{(39)} = 3.58, p > 0.05$ ). These results would suggest that lexical specialization as such violates the randomness assumption and gives rise to the discrepancy between the observed and expected vocabulary growth curves. Unfortunately, some of the assumptions underlying (7) are questionable.

First, for the majority of texts, the number of so-called hapax legomena,  $V_N(1)$ , accounts for roughly half the number of types  $V_N$ . Hapaxes, by virtue of occurring once only, cannot enjoy specialized use, if the operationalization of lexical specialization in terms of the bundled occurrence of all the tokens of a given type in a particular segment of the text is not to be trivialized. Second, if text slices in which specialized words occur are characterized by a deficit in the number of types, there should also be text slices with a surplus of types — the successive increments in the vocabulary size sum up to  $V_N$  for both the expected and the observed counts. If the text slices with a surplus of types also occur randomly in the text, it may well be that the effects of

lexical specialization are counterbalanced by effects of lexical richness. If so, no discrepancy between theory and observation should arise. Third, observe that in Figure 1  $E[V_M] - V_M$  tends to be negative for large  $M$  in the novels by Melville and Multatuli. Application of (7) and (9) shows that for *Moby Dick* the optimal choice for Labbe and Hubert's parameter  $p$ , 0.12, does not yield an acceptable fit ( $\chi^2_{(39)} = 162.79, p < 0.001$ ), and the same holds for *Max Havelaar*,  $\chi^2_{(39)} = 92.58, p < 0.001$  for the Labbe and Hubert parameter  $p = 0.10$ . If the modification of (4) proposed by Labbe and Hubert (1993) is has any validity at all, this validity is restricted to texts with the developmental profile of *Alice in Wonderland* only. Texts with skewed profiles such as observed for *Moby Dick* and *Max Havelaar* cannot be analyzed in this way. We may conclude that if lexical specialization is to lead to violation of the randomness assumption, specialized types should not be randomly distributed in the text.

#### 4.3 Discourse Structure

To test for possible effects of lexical specialization as a function of the discourse structure of the text, we need a formal definition of lexical specialization. Given the intuitive idea that lexical specialization implies a significant concentration in the occurrences of a word, we can define lexical specialization in terms of underdispersion. If a text is divided into  $K$  text slices, the dispersion  $d_i$  of word  $A_i$  is defined as the number of text slices in which this word occurs. If a word's dispersion is smaller than expected under chance conditions, it is underdispersed. To test whether  $A_i$  is significantly underdispersed, the test statistic

$$Z_i = \frac{d_i - E[d_i]}{\sqrt{\text{VAR}[d_i]}} \quad (10)$$

can be used. Since we have no reason to suppose that overdispersion occurs, we may assume that  $A_i$  is significantly underdispersed at the 5% level when  $Z_i < -1.645$ .

Expressions for  $E[d_i]$  and  $\text{VAR}[d_i]$  can be obtained using occupancy theory (Johnson and Kotz 1977: 113-114). Let  $X$  denote the number of text slices unoccupied by a token of word  $A_i$  with frequency  $f_N(A_i)$ . On partitioning a text into  $K$  slices, we can express  $X$  as the sum of the individual unoccupied slices:

$$X = \sum_{k=1}^K X_k, \quad (11)$$

with

$$X_k = \begin{cases} 0 & \text{if } A_i \text{ appears in the } k^{\text{th}} \text{ text slice,} \\ 1 & \text{otherwise.} \end{cases} \quad (12)$$

The number of text slices occupied by at least one token of  $A_i$  equals  $d_i = K - X$ . Since  $\text{Pr}(X_k = 1) = (1 - p_k)^{f_N(A_i)}$ , with  $p_k$  the probability of assigning

a word token to the  $k^{\text{th}}$  text slice, we find that

$$\begin{aligned}
\mathbb{E}[d_i] &= \mathbb{E}\left[K - \sum_{k=1}^K X_k\right] \\
&= K - \sum_{k=1}^K \mathbb{E}[X_k] \\
&= K - \sum_{k=1}^K (1 - p_k)^{f_N(A_i)}. \tag{13}
\end{aligned}$$

When a given word token is equally likely to be assigned to any of the text slices, (13) reduces to

$$\mathbb{E}[d_i] = K\left(1 - \left(1 - \frac{1}{K}\right)^{f_N(A_i)}\right). \tag{14}$$

After allotting  $N$  word tokens to  $K$  equiprobable text slices, each text slice will contain on average  $N/K$  word tokens. This allows us to use (14) to estimate the expected dispersion of all types  $A_i$  for the 40 equally large text slices of *Alice in Wonderland*, *Moby Dick*, and *Max Havelaar*.

The variance of  $d_i$  is obtained as follows.

$$\begin{aligned}
\text{VAR}[X] &= \text{VAR}\left[\sum_{k=1}^K X_k\right] \\
&= \sum_k \text{VAR}[X_k] + 2 \sum_{n < m} \text{COV}(X_n X_m) \\
&= \sum_k (\mathbb{E}[X_k^2] - (\mathbb{E}[X_k])^2) \\
&\quad + 2 \sum_{n < m} (\mathbb{E}[X_n X_m] - \mathbb{E}[X_n]\mathbb{E}[X_m]). \tag{15}
\end{aligned}$$

As  $X_k^2$  is nonzero only when  $X_k = 1$ ,  $\mathbb{E}[X_k^2] = \mathbb{E}[X_k]$ . Similarly, we have that  $\mathbb{E}[X_n X_m] = 1$  iff  $X_n = X_m = 1$ , and hence

$$\begin{aligned}
\mathbb{E}[X_n X_m] &= \Pr(X_n X_m = 1) \\
&= \Pr(\{X_n = 1\} \cap \{X_m = 1\}) \\
&= (1 - p_n - p_m)^{f_N(A_i)}. \tag{16}
\end{aligned}$$

This leads directly to

$$\begin{aligned}
\text{VAR}[X] &= \sum_{k=1}^K (1 - p_k)^{f_N(A_i)} (1 - (1 - p_k)^{f_N(A_i)}) \\
&\quad + 2 \sum_{n < m} (1 - p_n - p_m)^{f_N(A_i)} - (1 - p_n)^{f_i} (1 - p_m)^{f_N(A_i)}. \tag{17}
\end{aligned}$$

For equally sized text slices,  $\text{VAR}[d_i] = \text{VAR}[K - X] = \text{VAR}[X]$  is simplified to

$$\begin{aligned} \text{VAR}[d_i] = & K\left(1 - \frac{1}{K}\right)^{f_N(A_i)} + K(K-1)\left(1 - \frac{2}{K}\right)^{f_N(A_i)} \\ & - K^2\left(1 - \frac{1}{K}\right)^{2f_N(A_i)}. \end{aligned} \quad (18)$$

For each of the 40 text slices of *Alice in Wonderland*, *Moby Dick*, and *Max Havelaar*, I calculated the number of significantly underdispersed words. To study the relation between the growth of the vocabulary and the amount of underdispersion, it is useful to compare, for each successive text slice, the influx of new types with the influx of new underdispersed types. In order to compare observed with empirical values, it is convenient to introduce two difference functions. Let  $D_V(k)$  denote the difference between the expected and observed number of new types in text slice  $k$ ,

$$D_V(k) = (\text{E}[V_{M,k}] - \text{E}[V_{M,k-1}]) - (V_{M,k} - V_{M,k-1}), \quad (19)$$

and let

$$D_U(k) = (U_{M,k} - U_{M,k-1}) - (\text{E}[U_{M,k}] - \text{E}[U_{M,k-1}]), \quad (20)$$

with  $U_{M,k}$  the number of underdispersed types in the  $k^{\text{th}}$  text slice, denote the difference between the observed and expected numbers of new underdispersed types in text slice  $k$ . Figure 3 plots  $D_V(k)$  (small dots) and  $D_U(k)$  (large dots) and the corresponding smoothed curves using running medians (Tukey, 1977) for our three texts. In each case, we find that the two curves tend to be each other's mirror images. Especially for the first 7 measurement points,  $D_V(k)$  tends to be large and  $D_U(k)$  small. In other words, in the initial parts of these novels, both new types and significantly underdispersed types are scarce. In later parts of the novels, there is a tendency for the expected increase in vocabulary to slightly underestimate the empirical increase, and it is here that the empirical numbers of underdispersed words are slightly higher than expected.

This pattern of results suggests that lexical specialization, defined in terms of significant underdispersion, is not randomly distributed in the text, and that it is the scarcity of significant underdispersion in the initial segments of the text, combined with a deficit in type richness, that gives rise to the divergence between the observed and expected vocabulary growth curves. In hindsight, it is obvious that lexical specialization and vocabulary richness go hand in hand. When a particular topic is discussed in detail, key words for that topic will be used intensively. These key words are the significantly underdispersed words of this study (see Baayen, 1994, for detailed discussion). At the same time, additional vocabulary is called upon, without which the many facets of the topic that make it worth mentioning could not be discussed.

What we find, then, is that the organization of texts at the discourse level is at issue. In the initial sections of the text, the reader is introduced gently to the fictive world of the novel. Here, large numbers of specialized words, both

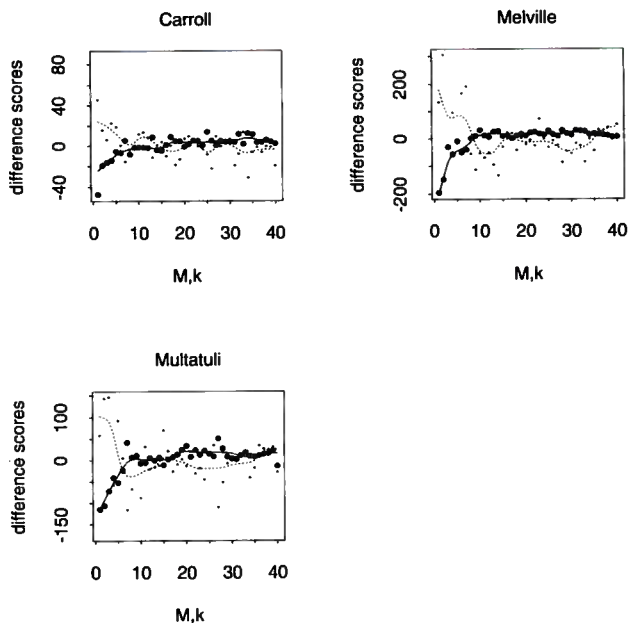


FIGURE 3. Difference scores  $D_V(k)$  (dotted line, small dots) and  $D_U(k)$  (solid line, large dots) for L. Carroll's *Alice in Wonderland*, H. Melville's *Moby Dick*, and Multatuli's *Max Havelaar*.

the hard-worked underdispersed words, as well as the specialized low-frequency words their use brings along, are avoided. Once the general topic domain has been established, specialized vocabulary is put to use to elaborate more specific topics in full.

## 5 DISCUSSION

I have shown that the lack of goodness-of-fit of any probabilistic model for word frequency distributions of texts that assumes that words occur randomly in texts is due to the way in which texts are structured on the discourse level. Syntactic and semantic constraints operating on the sentence level, as well as lexical specialization by itself, do not play a significant role.

This finding has important consequences for the statistical analysis of word frequency distributions, as it shows that the theoretical predictions of the urn model will be accurate either if the textual materials studied do not have the

discourse structure observed for the novels studied here, or if this discourse structure is irrelevant to the question at hand. The first possibility arises in studies where corpora are investigated. Corpora such as *Uit den Boogaart* (1975) and Kučera and Francis (1965) are collections of randomly sampled short text fragments of approximately the same length. No discourse organization will be present in the sequence of such fragments. Hence the model will accurately predict the observed vocabulary size for  $M < N$ .

The second possibility arises when the model is used to obtain estimates of population parameters that are relatively independent of discourse organization. For instance, in studies of vocabulary richness (Good and Toulmin 1976, Efron and Thisted 1976, Sichel 1986), the number of different word types in an author's vocabulary is estimated on the basis of one or more of his texts. If the number of different word types an author chooses to use to discuss a particular topic domain is independent of the way in which he structures the text to facilitate comprehension for the reader, then the rhetorical structure of the text becomes irrelevant when one's aim is to estimate the size of the vocabulary the author had at his disposal for discussing this topic domain, including the words he knew but did not use.

Summing up, the finding that the randomness assumption is violated at the level of discourse structure implies that word frequency models for which this assumption is crucial can nevertheless be reliably applied in corpus-based studies and in studies of lexical richness.

## 6 EPILOGUE

Having come to the end of my discussion of the randomness assumption in word frequency statistics, I would like to add a few words on the occasion of Cor Baayen's retirement as scientific director of the Centre for Mathematics and Computer Science.

As mentioned in the introduction, the first computerized frequency list of Dutch was compiled in 1965 at the Mathematical Centre, the name of the Centre for Mathematics and Computer Science at that time. The director of the institute, Aad van Wijngaarden, one of the pioneers of computer science in the Netherlands, had a keen interest in language in general, and in lexicology and etymology in particular. Not surprisingly, the first study of the Dutch language in which the computer was used as a tool for obtaining word frequency counts and for carrying out morphological analyses to appear in print was a *Mathematical Centre Tract* (van Berckel et al., 1965).

Van Wijngaarden's successor as director of the Mathematical Centre was Cor Baayen. While sharing the same interest in historical linguistics and etymology, Cor was well aware of the importance of methods of formal logic as tools for the analysis of problems of ambiguity and scope that arise at the level of the syntax and semantics of natural language, and he has stimulated research in the interdisciplinary domain of language, logic and computer science throughout his directorship.

Looking back, it is clear that the study of natural language in the Netherlands has profited from the erudition and breadth of vision of the Mathematical Centre's last scientific directors. It is to be hoped that the future CWI will be able to demonstrate a similar breadth of vision, stimulating the use of new mathematical techniques not only in the sciences and in engineering, but also in the humanities.

#### REFERENCES

- [Baaed] R. H. Baayen. The effects of lexical specialization on the growth curve of the vocabulary. (submitted), September 1994 submitted.
- [BPvR93] R. H. Baayen, R. Piepenbrock, and H. van Rijn. *The CELEX lexical database (CD-ROM)*. Linguistic Data Consortium, University of Pennsylvania, Philadelphia, PA, 1993.
- [Bru78] E. Brunet. *Le Vocabulaire de Jean Giraudoux*, volume 1 of *TLQ*. Slatkine, Genève, 1978.
- [Bur88] Burnage. *CELEX; A guide for users*. Centre for Lexical Information, Nijmegen, 1988.
- [CB93] R. J. Chitashvili and R. H. Baayen. Word frequency distributions. In G. Altmann and L. Hřebíček, editors, *Quantitative Text Analysis*, pages 54–135. Wissenschaftlicher Verlag Trier, Trier, 1993.
- [DIC37] J. F. H. A. De la Court. *De meest voorkomende woorden en woordcombinaties in het Nederlandsch*. Volkslectuur, Batavia, 1937.
- [ET76] B. Efron and R. Thisted. Estimating the number of unseen species: How many words did shakespeare know? *Biometrika*, 63:435–447, 1976.
- [Goo53] I. J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40:237–264, 1953.
- [GT56] I. J. Good and G. H. Toulmin. The number of new species and the increase in population coverage, when a sample is increased. *Biometrika*, 43:45–63, 1956.
- [HL88] P. Hubert and D. Labbe. Un modèle de partition du vocabulaire. In D. Labbe, P. Thoirou, and D. Serant, editors, *Etudes sur la richesse et les structures lexicales*, pages 93–114. Slatkine-Champion, Paris, 1988.
- [JK77] N. L. Johnson and S. Kotz. *Urn Models and Their Application. An Approach to Modern Discrete Probability Theory*. John Wiley & Sons, New York, 1977.
- [Jon79] E. D. de Jong. *Spreektaal. Woordfrequenties in Gesproken Nederlands*. Oosthoek, Scheltema en Holkema, Utrecht, 1979.



- [Kal65] V. M. Kalinin. Functionals related to the poisson distribution, and statistical structure of a text. In J. V. Fimmik, editor, *Articles on Mathematical Statistics and the Theory of Probability*, pages 202–220, Providence, Rhode Island, 1965. Steklov Institute of Mathematics 79, American Mathematical Society.
- [KF67] H. Kučera and W. N. Francis. *Computational Analysis of Present-Day American English*. Brown University Press, Providence, RI, 1967.
- [LH93] D. Labbe and P. Hubert. La richesse du vocabulaire (vocabulary richness). Centre de Recherche sur le Politique, l'Administration et le Territoire, October 1993.
- [Sic86] H. S. Sichel. Word frequency distributions and type-token characteristics. *Mathematical Scientist*, 11:45–72, 1986.
- [Tuk77] J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, Reading, Mass., 1977.
- [UdB75] P. C. Uit den Boogaart, editor. *Woordfrequenties in Gesproken en Geschreven Nederlands*. Oosthoek, Scheltema & Holkema, Utrecht, 1975.
- [vBBCMvW65] J. A. Th. M. van Berckel, H. Brandt Corstius, R. J. Mokken, and A. van Wijngaarden. *Formal Properties of Newspaper Dutch*. Mathematisch Centrum, Amsterdam, 1965.
- [Yul44] G. U. Yule. *The Statistical Study of Literary Vocabulary*. Cambridge University Press, Cambridge, 1944.
- [Zip35] G. K. Zipf. *The Psycho-Biology of Language*. Houghton Mifflin, Boston, 1935.
- [Zip49] G. K. Zipf. *Human Behavior and the Principle of the Least Effort. An Introduction to Human Ecology*. Hafner, New York, 1949.

# Discriminating coded lambda terms

*Dedicated in friendship to Cor Baayen  
on the occasion of his retirement*

Henk Barendregt

*Computing Science Institute  
Catholic University Nijmegen*

*Department of Software Technology  
CWI, Amsterdam*

A *coding* for a (type-free) lambda term  $M$  is a lambda term  $\ulcorner M \urcorner$  in normal form such that  $M$  (and its parts) can be reconstructed from  $\ulcorner M \urcorner$  in a lambda definable way. Kleene[1936] defined a coding  $\ulcorner M \urcorner^K$  and a *self-interpreter*  $\mathbf{E}^K \in \Lambda^\circ$  such that

$$\forall M \in \Lambda^\circ \quad \mathbf{E}^K \ulcorner M \urcorner^K = M. \quad (1)$$

In this style one can construct a *discriminator*  $\Delta^K \in \Lambda^\circ$  such that

$$\forall M, N \in \Lambda \quad \Delta^K \ulcorner M \urcorner^K \ulcorner N \urcorner^K = \begin{cases} \mathbf{true} & (\equiv \lambda xy.x) \text{ if } M \equiv N; \\ \mathbf{false} & (\equiv \lambda xy.y) \text{ else.} \end{cases} \quad (2)$$

The terms  $\mathbf{E}^K$  and  $\Delta^K$  are complicated. They depend on the lambda definability of functions on the integers dealing with coded syntactic properties. Inspired by a construction of P. de Bruin (see Barendregt [1991]) Mogensen [1992] constructed a different coding  $\ulcorner M \urcorner$  and an efficient self-interpreter  $\mathbf{E} \in \Lambda^\circ$  such that

$$\forall M \in \Lambda \quad \mathbf{E} \ulcorner M \urcorner = M. \quad (3)$$

This construction does not use an encoding of syntax as numbers but directly as lambda terms. This results in a much less complex  $\mathbf{E}$ . Mogensen's construction was simplified even further in Böhm et al. [1994]. In this paper we construct a simple discriminator  $\Delta \in \Lambda^\circ$  such that

$$\forall M, N \in \Lambda^\circ \quad \Delta \ulcorner M \urcorner \ulcorner N \urcorner = \begin{cases} \mathbf{true} & \text{if } M \equiv_\alpha N; \\ \mathbf{false} & \text{else.} \end{cases} \quad (4)$$

Note that in (1) and (4) the statement is only about closed lambda terms, while that in (2) and (3) is about all lambda terms. It will become clear why this is so.

## 1. INTRODUCTION

The most important notations for the type-free lambda calculus will be given here. Background can be found in Barendregt [1984].

1.1. DEFINITION. Variables and terms of the lambda calculus are defined by the following abstract syntax.

$$\begin{aligned} \text{var} &= a \mid \text{var}' \\ \text{term} &= \text{var} \mid \text{term term} \mid \lambda \text{var term} \end{aligned}$$

NOTATION. (i)  $M, N, \dots, P, Q, \dots$  range over  $\lambda$ -terms. The letters  $x, y, z, \dots$  range over variables. Note that the variables are  $\{a, a', a'', \dots, a^{(n)}, \dots\}$ .

(ii)  $\Lambda$  is the set of lambda terms.  $\text{FV}(M)$  is the set of free variables of  $M$ . The set of closed terms is  $\Lambda^\circ = \{M \in \Lambda \mid \text{FV}(M) = \emptyset\}$ .

(iii) The relation  $\equiv$  denotes syntactic equality; the relation  $\equiv_\alpha$  denotes syntactic equality up to a change of names of the bound variables. For example

$$\lambda x.x \equiv_\alpha \lambda y.y \not\equiv \lambda x.x.$$

(iv) The relation  $=$  denotes  $\beta$ -convertibility, axiomatized by

$$(\lambda x.M)N = M[x := N].$$

Here  $[x := N]$  denotes substitution of  $N$  in the free occurrences of  $x$ . E.g.

$$(x(\lambda x.x))[x := a] \equiv a(\lambda x.x).$$

(v)  $\mathbb{N}$  is the set of natural numbers. For  $n \in \mathbb{N}$  the terms  $\mathbf{c}_n \equiv \lambda f x.f^n x$ , where  $f^0 x \equiv x$  and  $f^{n+1} x \equiv f(f^n x)$ , denote the so called Church numerals. Note that the  $\mathbf{c}_n$  are distinct normal forms; hence

$$\mathbf{c}_n = \mathbf{c}_m \Rightarrow n = m$$

by the Church-Rosser theorem.

A lambda term can be seen as an executable: the redexes want to be evaluated. In this sense a normal form is not executable anymore. For a lambda term  $M$  its *code*  $\ulcorner M \urcorner$  is a normal form such that  $M$  is reconstructible from  $\ulcorner M \urcorner$ . Kleene [1936] defined a code  $\ulcorner M \urcorner^k$  essentially as follows.

1.2. DEFINITION. (i) By induction on the structure of  $M$  we define  $\#M$ .

$$\begin{aligned} \#(a^{(n)}) &= \langle 0, n \rangle; \\ \#(PQ) &= \langle 1, \langle \#(P), \#(Q) \rangle \rangle; \\ \#(\lambda x.P) &= \langle 2, \langle \#(x), \#(P) \rangle \rangle. \end{aligned}$$

Here  $\langle -, - \rangle$  denotes a recursive pairing function on  $\mathbb{N}$  with the recursive projections  $(-)_0, (-)_1$ :

$$\langle n_0, n_1 \rangle_i = n_i.$$

(ii) The map  $\ulcorner - \urcorner^K : \Lambda \rightarrow \Lambda$  is defined by

$$\ulcorner M \urcorner^K = \mathbf{c}_{\#M}.$$

Note that for all  $M \in \Lambda$  the term  $\ulcorner M \urcorner^K$  is in normal form. Moreover,

$$\ulcorner M \urcorner^K = \ulcorner N \urcorner^K \Rightarrow M \equiv N.$$

1.3. PROPOSITION. *There is no lambda term  $Q$  such that for all  $M \in \Lambda^{(o)}$  one has*

$$QM = \ulcorner M \urcorner^K.$$

PROOF. Suppose  $Q$  exists. Then for  $\mathbf{I} \equiv \lambda x.x$  one has

$$Q(\mathbf{I}) = \ulcorner \mathbf{I} \urcorner^K = \mathbf{c}_{\#\mathbf{I}} = \mathbf{c}_{\langle 1, \langle \#\mathbf{I}, \#\mathbf{I} \rangle \rangle}.$$

But also

$$Q(\mathbf{I}) = Q\mathbf{I} = \ulcorner \mathbf{I} \urcorner^K = \mathbf{c}_{\#\mathbf{I}} = \mathbf{c}_{\langle 2, \langle \#(x), \#(x) \rangle \rangle}.$$

Hence  $\langle 1, \langle \#\mathbf{I}, \#\mathbf{I} \rangle \rangle = \langle 2, \langle \#(x), \#(x) \rangle \rangle$ , a contradiction. ■

In spite of this fact that the ‘quote’  $Q$  does not exist, the inverse ‘evaluation’  $\mathbf{E}$  can be constructed.

1.4. THEOREM (Kleene [1936]). *There exists an  $\mathbf{E}^K \in \Lambda^\circ$  such that for all  $M \in \Lambda^\circ$  one has*

$$\mathbf{E}^K \ulcorner M \urcorner^K = M.$$

PROOF. See Kleene [1936] or Barendregt [1984], theorem 8.1.16. ■

The self-interpreter  $\mathbf{E}$  can work only for closed terms  $M$  (or terms having at most a fixed finite set of free variables). The reason is that if

$$\mathbf{E}^K \ulcorner M \urcorner^K = M,$$

then

$$\text{FV}(M) \subseteq \text{FV}(\mathbf{E}^K \ulcorner M \urcorner^K) = \text{FV}(\mathbf{E}^K).$$

Therefore if  $\mathbf{E}^K$  is closed, then the  $M$  have to be closed as well. This causes one difficulty in the construction of  $\mathbf{E}^K$ . The closed terms do not form a context-free language. Kleene solved this problem by constructing  $\mathbf{E}$  first for the set of combinatory terms  $\mathcal{C}^\circ$  built from the basis  $\{\mathbf{K}, \mathbf{S}\}$  using application only; then the real self-interpreter can be obtained by translations between  $\Lambda^\circ$  and  $\mathcal{C}^\circ$ .

A different construction of a self-interpreter was given by a former student of mine, using ideas from denotational semantics.

1.5. THEOREM (P. de Bruin). *There exists an  $\mathbf{E}_0 \in \Lambda^\circ$  such that for all  $M \in \Lambda$  and all  $F \in \Lambda$  one has*

$$\mathbf{E}_0 \ulcorner M \urcorner F = M[x_1, \dots, x_n := F \ulcorner x_1 \urcorner, \dots, F \ulcorner x_n \urcorner] \quad (5)$$

(simultaneous substitution), where  $\{x_1, \dots, x_n\} = \text{FV}(M)$ .

PROOF. By the representability of computable functions and the fixedpoint theorem there is a term  $\mathbf{E}_0 \in \Lambda^\circ$  such that

$$\begin{aligned}\mathbf{E}_0^\ulcorner x^\urcorner^K F &= F^\ulcorner x^\urcorner^K; \\ \mathbf{E}_0^\ulcorner PQ^\urcorner^K F &= F(\mathbf{E}_0^\ulcorner P^\urcorner^K F)(\mathbf{E}_0^\ulcorner Q^\urcorner^K F); \\ \mathbf{E}_0^\ulcorner \lambda x.P^\urcorner^K F &= \lambda x.(\mathbf{E}_0^\ulcorner P^\urcorner^K F_{[\ulcorner x^\urcorner \mapsto x]}),\end{aligned}$$

where  $F_{[\ulcorner x^\urcorner \mapsto x]} = F'_x$  with

$$\begin{aligned}F'_x x^\ulcorner &= x; \\ F'_x y^\ulcorner &= F^\ulcorner y^\urcorner, \text{ if } y \neq x.\end{aligned}$$

Note that  $F'_x$  can be written as  $G F x$ , with  $G$  closed. By induction on the structure of  $M \in \Lambda$  one can show that the statement holds. ■

1.6. COROLLARY. *There exists an  $\mathbf{E}^{dB} \in \Lambda^\circ$  such that for all  $M \in \Lambda^\circ$  one has*

$$\mathbf{E}^\ulcorner M^\urcorner^K = M.$$

PROOF (P. de Bruin). We can take

$$\mathbf{E}^{dB} \equiv \lambda m. \mathbf{E}_0 m \mathbf{l}.$$

Indeed, for closed terms  $M$  it follows from (5) that

$$\mathbf{E}^{dB \ulcorner M^\urcorner} = \mathbf{E}_0^\ulcorner M^\urcorner \mathbf{l} = M. \blacksquare$$

## 2. REPRESENTING DATA TYPES

After seeing the method of P. de Bruin, Mogensen [1992] gave an improved version of it by representing data types directly (i.e. not using the natural numbers) in lambda calculus as done in e.g. Böhm and Berarducci [1985]. This approach was improved later by Böhm et al. [1994] by constructing a new representation of data types into type-free lambda calculus. This new representation will be treated in a slightly modified form in this section.

2.1. DEFINITION. Write

$$\begin{aligned}\langle M_1, \dots, M_n \rangle &= \lambda z. z M_1 \dots M_n; \\ \mathbf{U}_i^n &= \lambda x_1 \dots x_n. x_i; \\ \mathbf{true} &= \mathbf{U}_1^2; \\ \mathbf{false} &= \mathbf{U}_2^2.\end{aligned}$$

Note that

$$\begin{aligned}\langle M_1, \dots, M_n \rangle \mathbf{U}_i^n &= M_i; \\ \mathbf{true} PQ &= P; \\ \mathbf{false} PQ &= Q.\end{aligned}$$

In particular we have  $\langle M \rangle = \lambda z. z M$  and  $\langle \rangle = \lambda x. x = \mathbf{l}$ . Now we define the notion of lists inspired by the language LISP, McCarthy et al. [1961].

2.2. DEFINITION. (i) Write

$$\begin{aligned} \text{nil} &= \langle \rangle; \\ \text{cons} &= \lambda xy. \langle x, y \rangle; \\ \text{car} &= \langle \mathbf{U}_1^2 \rangle; \\ \text{cdr} &= \langle \mathbf{U}_2^2 \rangle; \\ \text{null?} &= \langle \mathbf{U}_3^3, \mathbf{U}_1^2, \text{false}, \text{true} \rangle. \end{aligned}$$

(ii) Define

$$\begin{aligned} [] &= \langle \rangle; \\ [M_1, \dots, M_{n+1}] &= \text{cons } M_1 [M_2, \dots, M_{n+1}]. \end{aligned}$$

So for example

$$[M_1, M_2, M_3] = \langle M_1, \langle M_2, \langle M_3, \langle \rangle \rangle \rangle \rangle.$$

(In Barendregt [1984] this term is written as  $[M_1, M_2, M_3, \mathbf{I}]$ . At the time of writing that book we did not yet see the usefulness of terminating a list with a special constructor.) Note that

$$\begin{aligned} \text{car}(\text{cons } PQ) &= P; \\ \text{cdr}(\text{cons } PQ) &= Q; \\ \text{null? nil} &= \text{true}; \\ \text{null?}(\text{cons } PQ) &= \text{false}. \end{aligned}$$

2.3. PROPOSITION. *There exists lambda definable functions  $( )_i$  such that for  $1 \leq i \leq n$  one has*

$$([M_1, \dots, M_n])_i = M_i.$$

PROOF. Take

$$\begin{aligned} (l)_1 &= \text{car } l; \\ (l)_{i+1} &= (\text{cdr } l)_i. \blacksquare \end{aligned}$$

2.4. DEFINITION. An (*algebraic*) *signature*  $s$  consists of a number  $n \in \mathbb{N}$  (thought of as the list of symbols  $[f_1, \dots, f_n]$ ) together with a list of numbers  $[s_1, \dots, s_n]$  (thought of as the *arity* of the respective  $f_i$ 's). We write  $s = [s_1, \dots, s_n]$ .

For example a field has signature  $s = [2, 2, 1, 1, 0, 0]$  (thought of as the arities of the functionsymbols  $[+, \times, -, ^{-1}, 0, 1]$ ; so  $f_1 = +, f_2 = \times$  etcetera).

2.5. DEFINITION. If  $s$  is a signature then  $\text{term}_s$ , the set of *terms of signature*  $s$ , is defined as follows.

$$\begin{aligned} x \in \text{var} &\Rightarrow x \in \text{term}_s; \\ t_1, \dots, t_{s_i} \in \text{term}_s &\Rightarrow f_i(t_1, \dots, t_{s_i}) \in \text{term}_s. \end{aligned}$$

For example in the signature of fields the term  $f_1(f_1(x, f_3(f_2(y, f_4(z)))), f_6)$  is usually written as  $x - yz^{-1} + 1$ .

2.6. DEFINITION. Let  $s = [s_1, \dots, s_n]$  be a signature.

- (i) A *lambda interpretation* of  $s$  is a list of ‘constructors’  $C_1, \dots, C_n \in \Lambda$ .
- (ii) Let  $C_1, \dots, C_n$  be a lambda interpretation of  $s$ . Then we define a map

$$T : \mathbf{term}_s \rightarrow \Lambda$$

as follows.

$$\begin{aligned} T_x &= x; \\ T_{f_i(t_1, \dots, t_{s_i})} &= C_i[T_{t_1}, \dots, T_{t_{s_i}}], \end{aligned}$$

where  $[T_{t_1}, \dots, T_{t_{s_i}}]$  is the list operation on lambda terms defined in 2.2.

Example. The signature of *binary trees* is  $[0, 2]$ . The term  $t = f_2(f_2(f_1, f_1), f_1)$  denotes a simple tree and  $t' = f_2(f_1, f_2(f_1, f_1))$  its mirror image. Can we find a lambda interpretation for this signature in such a way that mirroring becomes lambda definable, i.e. for some  $F \in \Lambda^\circ$  one has  $FT_t = T_{t'}$ ? The following result, due to Böhm et al. [1994], will affirm this. We present the result in a modified form that will be useful for §4.

2.7. THEOREM. *For every algebraic signature  $s = [s_1, \dots, s_n]$  there exists a lambda interpretation  $C_1, \dots, C_n$  such that the following hold.*

- (i)  $\forall A_1 \dots A_n \exists F$

$$FT_{f_i(t_1, \dots, t_{s_i})} = A_i[T_{t_1}, \dots, T_{t_{s_i}}]F, \quad 1 \leq i \leq n. \quad (6)$$

- (ii) *The  $C_1, \dots, C_n$  only depend on  $n$ , not on the  $[s_1, \dots, s_n]$ . In (6) we can take  $F \equiv \langle\langle A_1, \dots, A_n \rangle\rangle$ .*

PROOF. Define  $C_{f_i} \equiv \lambda l e. e \mathbf{U}_i^n l \langle e \rangle$ .

- (i) Given  $A_1, \dots, A_n$ , we try whether  $F \equiv \langle\langle A_1, \dots, A_n \rangle\rangle$  works. Indeed,

$$\begin{aligned} FT_{f_i(t_1, \dots, t_{s_i})} &= \langle\langle A_1, \dots, A_n \rangle\rangle (C_{f_i}[T_{t_1}, \dots, T_{t_{s_i}}]) \\ &= C_{f_i}[T_{t_1}, \dots, T_{t_{s_i}}] \langle\langle A_1, \dots, A_n \rangle\rangle \\ &= \langle\langle A_1, \dots, A_n \rangle\rangle \mathbf{U}_i^n [T_{t_1}, \dots, T_{t_{s_i}}] \langle\langle A_1, \dots, A_n \rangle\rangle \\ &= A_i[T_{t_1}, \dots, T_{t_{s_i}}]F. \end{aligned}$$

- (ii) By the construction. ■

2.8. COROLLARY. *Let  $s = [s_1, \dots, s_n]$  be an algebraic signature. Let  $C_1, \dots, C_n$  be the lambda interpretation of  $s$  constructed in theorem 2.7. Then for all  $B_1 \dots B_n$  there exists an  $F$  such that*

$$F(C_{f_i}[x_1, \dots, x_{s_i}]) = B_i x_1 \dots x_{s_i} F, \quad 1 \leq i \leq n. \quad (7)$$



PROOF. Let  $B_1, \dots, B_n$  be given. Define  $A_i = \lambda l. B_i(l)_1 \dots (l)_{s_i}$ . Then

$$A_i[x_1, \dots, x_{s_i}] = B_i x_1, \dots, x_{s_i}.$$

Then the  $F$  for the  $A_i$  found in theorem 2.7 is the  $F$  satisfying (7). ■

Now we can program the function that ‘mirrors’ trees. In the signature  $[0, 2]$  for binary trees let

$$\begin{aligned} \mathbf{leaf} &= T_{f_1} &= C_{f_1} \langle \rangle; \\ \mathbf{tree} &= \lambda ab. T_{f_2(a,b)} &= \lambda ab. C_{f_2} \langle a, b \rangle. \end{aligned}$$

By corollary 2.8 there exists an  $F$  such that

$$\begin{aligned} F\mathbf{leaf} &= \mathbf{leaf}; \\ F(\mathbf{tree} \ a \ b) &= \mathbf{tree}(fb)(fa). \end{aligned}$$

This  $F$  has the mirror effect. E.g.  $F(f_2(f_2(f_1, f_1), f_1)) = f_2(f_1, f_2(f_1, f_1))$ .

### 3. A SIMPLE SELF-INTERPRETER

In Mogensen [1992] a simple coding and self-interpreter for lambda terms is defined, using the fact that data types (term algebras of a signature  $s$ ) have a lambda interpretation. The method was simplified by Böhm et al. [1994] by making use of their lambda representation of algebraic signatures given in §2.

3.1. DEFINITION. Let  $s$  be the signature  $[1, 2, 1]$ . Define

$$\begin{aligned} \mathbf{const} &= C_{f_1} \equiv \lambda e. e\mathbf{U}_1^3 l \langle e \rangle; \\ \mathbf{app} &= C_{f_2} \equiv \lambda e. e\mathbf{U}_2^3 l \langle e \rangle; \\ \mathbf{abs} &= C_{f_3} \equiv \lambda e. e\mathbf{U}_3^3 l \langle e \rangle. \end{aligned}$$

3.2. DEFINITION. For  $M \in \Lambda$  define  $\ulcorner M \urcorner$  as follows.

$$\begin{aligned} \ulcorner x \urcorner &= \mathbf{const} [x]; \\ \ulcorner PQ \urcorner &= \mathbf{app} [\ulcorner P \urcorner, \ulcorner Q \urcorner]; \\ \ulcorner \lambda x. P \urcorner &= \mathbf{abs} [\lambda x. \ulcorner P \urcorner]. \end{aligned}$$

Note that  $\text{FV}(\ulcorner M \urcorner) = \text{FV}(M)$ .

3.3. THEOREM (Mogensen [1992]). *There exists an  $\mathbf{E} \in \Lambda^\circ$  such that*

$$\forall M \in \Lambda \ \mathbf{E} \ulcorner M \urcorner = M.$$

PROOF (Böhm et al. [1994]). By corollary 2.8 there exists a term  $\mathbf{E} \in \Lambda^\circ$  such that

$$\begin{aligned} \mathbf{E}(\mathbf{const} [p]) &= p; \\ \mathbf{E}(\mathbf{app} [p, q]) &= (\mathbf{E}p)(\mathbf{E}q); \\ \mathbf{E}(\mathbf{abs} [p]) &= \lambda x. \mathbf{E}(px). \end{aligned}$$

Then

$$\begin{aligned}\mathbf{E}^\Gamma x^\neg &= x; \\ \mathbf{E}^\Gamma PQ^\neg &= (\mathbf{E}^\Gamma P^\neg)(\mathbf{E}^\Gamma Q^\neg); \\ \mathbf{E}^\Gamma \lambda x.P^\neg &= \lambda x.\mathbf{E}^\Gamma P^\neg.\end{aligned}$$

Now the results follows by induction on the structure of  $M$ . ■

Using the constructions in §2 the self-interpreter becomes

$$\mathbf{E} \equiv \langle\langle \lambda l f.(l)_1, \lambda l f.f(l)_1(f(l)_2), \lambda l f x.f((l)_1 x) \rangle\rangle.$$

The construction in Böhm et al. [1994] is simpler. They take

$$\begin{aligned}\mathbf{const} &= \lambda x e.e \mathbf{U}_1^3 x e; \\ \mathbf{app} &= \lambda x y e.e \mathbf{U}_2^3 x y e; \\ \mathbf{abs} &= \lambda x e.e \mathbf{U}_3^3 x e.\end{aligned}$$

The resulting self-interpreter then becomes  $\mathbf{E}^B = \langle\langle \mathbf{K}, \mathbf{S}, \mathbf{C} \rangle\rangle$ . Here  $\mathbf{K} \equiv \lambda x y.x$ ,  $\mathbf{S} \equiv \lambda x y z.xz(yz)$  and  $\mathbf{C} \equiv \lambda x y z.x(z y)$ . For reasons of uniformity we have given the definition of **const**, **app** and **abs** as in 3.1. This will be useful in §4.

#### 4. A SIMPLE DISCRIMINATOR

In this section we will construct a simple term discriminating between coded closed lambda term. The discrimination is even modulo  $\alpha$ -conversion. For open terms discrimination is possible only for the coding  $\Gamma \neg^K$  of Kleene.

4.1. LEMMA. (i) *There exists a term  $\delta_{\mathbf{N}} \in \Lambda^\circ$  such that*

$$\delta_{\mathbf{N}} \mathbf{c}_n \mathbf{c}_m = \begin{cases} \mathbf{true} & \text{if } n = m; \\ \mathbf{false} & \text{else.} \end{cases}$$

(ii) *There exists a term  $\mathbf{and} \in \Lambda^\circ$  such that*

$$\begin{aligned}\mathbf{and} \mathbf{true} \mathbf{true} &= \mathbf{true}; \\ \mathbf{and} \mathbf{true} \mathbf{false} &= \mathbf{false}; \\ \mathbf{and} \mathbf{false} \mathbf{true} &= \mathbf{false}; \\ \mathbf{and} \mathbf{false} \mathbf{false} &= \mathbf{false}.\end{aligned}$$

PROOF. (i) By the representability of the recursive functions.

(ii) Take  $\mathbf{and} = \lambda a b.a \mathbf{true} b$ . ■

4.2. PROPOSITION. *There exists a term  $\delta \in \Lambda^\circ$  such that (writing  $\delta_n$  for  $\delta_{\mathbf{c}_n}$ ) one has*

$$\begin{aligned}
\delta_n \ulcorner x \urcorner \urcorner x' \urcorner &= \delta_{\mathbb{N}} xy; \\
\delta_n \ulcorner x \urcorner \urcorner P' Q' \urcorner &= \mathbf{false}; \\
\delta_n \ulcorner x \urcorner \urcorner \lambda x'. P' \urcorner &= \mathbf{false}; \\
\\
\delta_n \ulcorner P Q \urcorner \urcorner x' \urcorner &= \mathbf{false}; \\
\delta_n \ulcorner P Q \urcorner \urcorner P' Q' \urcorner &= \mathbf{and}(\delta_n \ulcorner P \urcorner \urcorner P' \urcorner)(\delta_n \ulcorner Q \urcorner \urcorner Q' \urcorner); \\
\delta_n \ulcorner P Q \urcorner \urcorner \lambda x'. P' \urcorner &= \mathbf{false}; \\
\\
\delta_n \ulcorner \lambda x. P \urcorner \urcorner x' \urcorner &= \mathbf{false}; \\
\delta_n \ulcorner \lambda x. P \urcorner \urcorner P' Q' \urcorner &= \mathbf{false}; \\
\delta_n \ulcorner \lambda x. P \urcorner \urcorner \lambda x'. P' \urcorner &= \delta_{n+1}(\ulcorner P \urcorner[x := \mathbf{c}_n])(\ulcorner P' \urcorner[x' := \mathbf{c}_n]).
\end{aligned}$$

PROOF. We introduce the following ad hoc notation.

(i) Let  $A_1, \dots, A_n \in \Lambda$ . Then we write

$$\lambda \vec{x}! [A_1, \dots, A_n] \equiv \langle \langle \lambda \vec{x}. A_1, \dots, \lambda \vec{x}. A_n \rangle \rangle.$$

(ii) If  $B_i \equiv [A_{i1}, \dots, A_{in}]$ , then we write

$$\lambda \vec{x}! [B_1, \dots, B_n] \equiv \langle \langle \lambda \vec{x}! B_1, \dots, \lambda \vec{x}! B_n \rangle \rangle.$$

(iii) Let for  $1 \leq i \leq n, 1 \leq j \leq n$  be given  $A_{ij} \in \Lambda$ . Then

$$\begin{aligned}
[A_{ij}] &\equiv [[A_{11}, \dots, A_{1n}], \\
&\quad [A_{21}, \dots, A_{2n}], \\
&\quad \dots \\
&\quad [A_{n1}, \dots, A_{nn}]].
\end{aligned}$$

If  $n = 3$  we may write  $[A_{ij}]$  as

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}.$$

Now define  $\delta \equiv \lambda n t t'.$

$$\left( \lambda t d! \lambda t' d' n!! \left[ \begin{array}{ccc} \delta_{\mathbb{N}}(t)_1(t')_1 & \mathbf{false} & \mathbf{false} \\ \mathbf{false} & \mathbf{and}(d(t)_1(t')_1 n)(d(t)_2(t')_2 n) & \mathbf{false} \\ \mathbf{false} & \mathbf{false} & d(tn)(t'n)(\mathbf{S}^+ n) \end{array} \right] \right) t t' n,$$

where  $\mathbf{S}^+$  lambda defines the successor function. This  $\delta$  satisfies the specification. ■

4.3. PROPOSITION. *For all  $M, M' \in \Lambda$  such that  $\mathbf{FV}(MM') \subseteq \{x_1, \dots, x_n\}$  and for substitutions  $*$  =  $[x_1 := \mathbf{c}_{k_1}] \dots [x_n := \mathbf{c}_{k_n}]$  with  $k_i \neq k_j$  (for  $1 \leq i < j \leq n$ ) one has for  $p > k_i$  (for all  $1 \leq i \leq n$ ) that*

$$\delta_p \ulcorner M \urcorner \urcorner M' \urcorner * = \begin{cases} \mathbf{true} & \text{if } M \equiv_\alpha M'; \\ \mathbf{false} & \text{else.} \end{cases}$$

PROOF. By induction on the structure of  $M$ , in each case making distinctions according to the structure of  $M'$ . We treat four instructive cases.

Case  $M \equiv x, M' \equiv x'$ . Then

$$\delta_p \ulcorner M \urcorner M' \urcorner * = \delta_{\mathbb{N}} \mathbf{c}_{k_i} \mathbf{c}_{k_{i'}},$$

where  $x \equiv x_i, x' \equiv x_{i'}$ . This is true or false depending on whether  $x \equiv x'$  (so  $i = i'$ ) or  $x \not\equiv x'$  (so  $i \neq i'$ ).

Case  $M \equiv x, M' \equiv P'Q'$ . Then

$$\delta_p \ulcorner M \urcorner M' \urcorner * = \mathbf{false}.$$

Case  $M \equiv PQ, M' \equiv P'Q'$ . Then

$$\begin{aligned} \delta_p \ulcorner M \urcorner M' \urcorner * &= \mathbf{and}(\delta_p \ulcorner P \urcorner P' \urcorner *)(\delta_p \ulcorner Q \urcorner Q' \urcorner *) \\ &=_{IH} \mathbf{and}(\mathbf{true} / \mathbf{false})(\mathbf{true} / \mathbf{false}) \\ &= \mathbf{true} / \mathbf{false}, \end{aligned}$$

as it should ( $= \mathbf{true}$  only if  $PQ \equiv P'Q'$  i.e. if both  $P \equiv P'$  and  $Q \equiv Q'$ ).

Case  $M \equiv \lambda x.P, M' \equiv \lambda x'.P'$ . Then

$$\begin{aligned} \delta_p \ulcorner M \urcorner M' \urcorner * &= \delta_{p+1}(\ulcorner P \urcorner [x := \mathbf{c}_p])(\ulcorner P' \urcorner [x' := \mathbf{c}_p]) * \\ &= \delta_{p+1} \ulcorner P \urcorner P' [x' := x] \urcorner [x := \mathbf{c}_p] * \\ &= \delta_{p+1} \ulcorner P \urcorner P' [x' := x] \urcorner *', \end{aligned}$$

with  $*' = *[x := \mathbf{c}_p]$  being an admissible substitution. So

$$\delta_p \ulcorner M \urcorner M' \urcorner * =_{IH} \begin{cases} \mathbf{true} & \text{if } P \equiv_{\alpha} P' [x' := x]; \\ \mathbf{false} & \text{else.} \end{cases}$$

Now  $M \equiv_{\alpha} M'$  iff  $\lambda x.P \equiv_{\alpha} \lambda x'.P'$  ( $\equiv_{\alpha} \lambda x.P' [x' := x]$ ) iff  $P \equiv_{\alpha} P' [x' := x]$ . Hence we are done. ■

4.4. COROLLARY. Write  $\Delta \equiv \delta_0$ . Then for all  $M, M' \in \Lambda^{\circ}$  one has

$$\Delta \ulcorner M \urcorner M' \urcorner = \begin{cases} \mathbf{true} & \text{if } M \equiv_{\alpha} M'; \\ \mathbf{false} & \text{else.} \end{cases}$$

PROOF. Immediate from the proposition. ■

Note that this corollary cannot hold for arbitrary  $M, M' \in \Lambda$ . For example, it is impossible to discriminate  $\ulcorner x \urcorner$  and  $\ulcorner x' \urcorner$ . Indeed take  $x \not\equiv x'$  and make a substitution:

$$\Delta \ulcorner x \urcorner \urcorner x' \urcorner = \mathbf{false} \Rightarrow \Delta \ulcorner x \urcorner \urcorner x \urcorner = \mathbf{false},$$

a contradiction.

## REFERENCES

BARENDREGT, H.P.

[1984] *The Lambda Calculus, its Syntax and Semantics*, revised edition, Studies in Logic and the Foundations of Mathematics, North-Holland, Amsterdam.

[1991] Theoretical pearls: Self-interpretation in lambda calculus, *J. Funct. Programming*, **1**(2), 229–233.

BÖHM, C., and A. BERARDUCCI

[1985] Automatic synthesis of typed  $\lambda$ -programs on term algebras, *Theor. Comput. Sci.* **39**, 135–154.

BÖHM, C., A. PIPERNO and S. GUERRINI

[1994]  $\lambda$ -definitions of function(al)s by normal forms, in: *ESOP '94*, (ed. D. Sannella), LNCS 788, Springer, Berlin, 135–149

CHURCH, A.

[1941] *The calculi of lambda conversion*, Princeton University Press, Princeton. Reprinted by Kraus reprint corporation, New York, 1965.

KLEENE, S.C.

[1936]  $\lambda$ -definability and recursiveness, *Duke Math. J.* **2**, 340–353.

MCCARTHY, J., P.W. ADAMS, D.J. EDWARDS, T.P. HART and M.I. LEVIN

[1962] *LISP 1.5 Programmer's manual*, MIT Press.

MOGENSEN, T.Æ.

[1992] Efficient self-interpretation in lambda calculus, *J. Funct. Programming*, **2**(3), 345–364.



# New Trends in Applied Mathematics

*Dedicated to Professor Cor Baayen*

A. Bensoussan

*University Paris-Dauphine and INRIA*

Applied mathematics has become an extremely important and useful discipline in the context of development of powerful computers. On the one hand, mathematics (in a broad sense) is the most efficient approach to model reality, especially complex reality. Moreover, it provides the best possibilities of reasoning. With cheap powerful computers, mathematics becomes implementable and unavoidable in designing, producing, deciding ...

On the other hand, mathematics has evolved considerably to extend its applicability to real problems. This is why applied mathematics is so alive and fast progressing. Needless to say, the connection between applied mathematics and information technology is an extremely fruitful approach to new ideas and a basic source of research topics. This is a line to which Professor Cor Baayen has always dedicated his efforts. He has greatly contributed to closing the gap between mathematics and computer science. To give an exhaustive presentation of all directions of applied mathematics in a short talk is of course out of reach, and beyond the possibilities of one speaker. So the purpose of this lecture is more to outline some significant features, among many others.

## 1 SCIENTIFIC COMPUTING

The traditional applications of mathematics arise in Physics, Mechanics, .... Powerful computing means and supercomputers have permitted :

- to study completely new areas of physical sciences.
- to consider new numerical techniques
- to investigate new approaches.

### *1.1 New Areas of physical sciences*

It would be particularly unrealistic to be exhaustive here. Nevertheless, among important developments in several fields, we emphasize the *Numerical Simulation of Reactive flow*. It applies indeed to *combustion, aeronomy, partially*



ionized plasmas, aerodynamics, gas dynamic lasers, astrophysics, general multiphase and magneto-hydrodynamic flows, . . . .

The model takes into account the *coupling* between *fluid dynamics* and *chemical reactions*, and thus opens the door to a large family of complex problems.

The traditional model of an *homogeneous, viscous, incompressible flow* with *no chemical reactions* and *no external forces* consists of Navier Stokes equations :

$$\begin{aligned} \rho_0 \left( \frac{\partial u}{\partial t} + (u \cdot D)u \right) - \mu \Delta u + Dp &= 0 \\ \operatorname{div} u &= 0 \end{aligned}$$

If the fluid has a constant *specific heat*  $c$  and there are no external heat sources, then the temperature of the fluid is the solution of :

$$\rho_0 c \left( \frac{\partial T}{\partial t} + u \cdot DT \right) - \lambda \Delta T = 2 \operatorname{tr} \varepsilon^2(u)$$

where  $\varepsilon = \frac{1}{2}(Du + (Du)^T)$  is the *velocity tensor*. The internal energy density is  $cT$ .

In general, all variables are coupled and appear as the solution of a complex system of P.D.E.

The main unknown are the mass density  $\rho$ , the velocity of the flow  $u$ , the number densities  $n^i$  of the individual chemical species and the total energy density  $E$ .

The system of equations is the following :

$$\begin{aligned} \frac{\partial \rho}{\partial t} + D \cdot (\rho u) &= 0 \\ \frac{\partial (\rho u)}{\partial t} + D \cdot (\rho u u) + D \cdot \sigma &= \sum_i \rho^i a^i \\ \frac{\partial n^i}{\partial t} + D \cdot (n^i (u + u^i)) &= Q^i - L^i n^i \\ \frac{\partial E}{\partial t} + D \cdot (Eu) + D \cdot (u \cdot \sigma) + D \cdot (q + q_r) &= \sum_i (u + u^i) \cdot m^i a^i \end{aligned}$$

where  $\sigma$  is the *pressure tensor*,  $q$  the *heat flux*,  $q_r$  the *radiative heat flux*,  $a^i$  represent external forces, and  $Q^i, L^i$  represent the chemical production rates and losses of species  $i$ ,  $u^i$  is the diffusion velocity of species  $i$ . They are highly nonlinear expressions of the unknowns, including the temperature  $T$ .

In view of the complexity, a *modular* approach is useful. Each physical process is calculated accurately and calibrated separately .

The physical properties should be incorporated in the numerical algorithms and a mathematical analysis of the behaviour of the algorithms should be performed. For more details, see [18].

### 1.2 Numerical methods

We shall illustrate the general idea of *decoupling* the difficulties in the case of Navier Stokes equations:

$$\begin{aligned} \frac{\partial u}{\partial t} + (u \cdot D)u - \mu \Delta u + Dp &= f \\ \operatorname{div} u &= 0 \\ u(x, 0) &= u_o(x) & \operatorname{div} u_0 &= 0 \\ u &= g \text{ on } \Gamma & \int_{\Gamma} \nu \cdot g \, d\Gamma &= 0 \end{aligned}$$

The two main difficulties are non linearities and incompressibility condition. *Operator splitting* will realize the decoupling.

Let  $\theta$  be a parameter in  $(0, \frac{1}{2})$  and  $\alpha, \beta$  with  $\alpha + \beta = 1$ .

Knowing  $u^n$ , we compute  $\{u^{n+\theta}, p^{n+\theta}\}$ ,  $u^{n+1-\theta}$  and  $\{u^{n+1}, p^{n+1}\}$  by the iteration :

$$\begin{aligned} \frac{u^{n+\theta} - u^n}{\theta \Delta t} - \alpha \mu \Delta u^{n+\theta} + Dp^{n+\theta} &= \beta \mu \Delta u^n \\ -(u^{n+\theta} \cdot D)u^n + f^{n+\theta} & \\ \operatorname{div} u^{n+\theta} &= 0 \\ u^{n+\theta} &= g^{n+\theta} \quad \text{on } \Gamma \end{aligned} \tag{1}$$

$$\begin{aligned} \frac{u^{n+1-\theta} - u^{n+\theta}}{(1-2\theta)\Delta t} - \beta \mu \Delta u^{n+1-\theta} + (u^{n+1-\theta} \cdot D)u^{n+1-\theta} &= \alpha \mu \Delta u^{n+\theta} \\ -Dp^{n+\theta} + f^{n+1-\theta} & \\ u^{n+1-\theta} &= g^{n+1-\theta} \quad \text{on } \Gamma \end{aligned} \tag{2}$$

$$\begin{aligned} \frac{u^{n+1} - u^{n+1-\theta}}{\theta \Delta t} - \alpha \mu \Delta u^{n+1} + Dp^{n+1} &= \beta \mu \Delta u^{n+1-\theta} \\ -(u^{n+1-\theta} \cdot D)u^{n+1-\theta} + f^{n+1} & \\ \operatorname{div} u^{n+1} &= 0 \\ u^{n+1} &= g^{n+1} \quad \text{on } \Gamma \end{aligned} \tag{3}$$

(2) is nonlinear and solved by a *least square technique*, and *conjugate gradient* minimization. (1) and (3) are linear and can be reformulated as variational problems for the pressure  $p$ .

Various possibilities of finite element approximation, multigrid methods and domain decomposition can then be used at the discretization stage.

Efficient software packages result in the combination of all these techniques. For more details, see [10].

### 1.3 New approaches

We present two new directions :

### 1.3.1 Wavelets

An alternative to Fourier analysis has been developed in recent years, with applications to signal and image processing, sound analysis and numerical analysis. It has foundations in quantum field theory, statistical mechanics and pure mathematics (geometry of Banach spaces). This is the *Wavelet analysis*.

It combines advantages of the Haar system and of the trigonometrical system. The Haar system is defined by :

$$\psi(x) = \begin{cases} 1, & 0 \leq x < \frac{1}{2} \\ -1 & \frac{1}{2} \leq x < 1 \\ 0 & \text{otherwise} \end{cases}$$

$$\psi_{m,n}(x) = 2^{-\frac{m}{2}} \psi(2^{-m}x - n), \quad m, n \in \mathbb{Z}.$$

The  $\psi_{m,n}$  form an orthonormal basis of  $L^2(\mathbb{R})$ , (and even  $L^p$ ) but not for Sobolev spaces (unlike trigonometric series for periodic Sobolev spaces). On the other hand, the  $\psi_{m,n}$  have good localization properties unlike trigonometric functions (the reverse being true for their Fourier transforms).

A wavelet system is defined by a function  $\psi(x)$  and

$$\psi_{m,n}(x) = 2^{-\frac{m}{2}} \psi(2^{-m}x - n)$$

with the property

$$L^2(\mathbb{R}) = \bigoplus_{m \in \mathbb{Z}} W_m$$

$$W_m = \overline{\text{span} \{ \psi_{m,n} \}}, \quad \text{orthogonal spaces}$$

$\{ \psi_{m,n}, n \in \mathbb{Z} \}$  is an orthonormal basis for  $W_m$ .

Y. Meyer has constructed a wavelet system with  $\psi, C^\infty$  with rapide decay (faster than any power). Later one has constructed a wavelet system with  $\psi, C^k$  with exponential decay, and finally I. Daubechies has shown the existence of wavelet systems with compact support and arbitrary regularity. They will be very useful for all kinds of applications.

They are obtained from sequences  $h_n$ , with compact support, satisfying *additional assumptions* by the following procedure :

$$\phi(x) = \lim_{k \rightarrow \infty} \eta_k(x)$$

with

$$\eta_k(x) = \sqrt{2} \sum_n h_n \eta_{k-1}(2x - n)$$

$$\eta_0 = \mathbb{1}_{[-\frac{1}{2}, \frac{1}{2}[}$$

then

$$\psi(x) = \sqrt{2} \sum_n (-1)^n h_{-n+1} \phi(2x - n)$$

The *most* compact support corresponds to the two possible choices :

$$h_0 = \frac{1 \mp \sqrt{3}}{4\sqrt{2}}, \quad h_1 = \frac{3 \mp \sqrt{3}}{4\sqrt{2}}, \quad h_2 = \frac{3 \pm \sqrt{3}}{4\sqrt{2}}, \quad h_3 = \frac{1 \pm \sqrt{3}}{4\sqrt{2}},$$

For more details, see [15], [9].

### 1.3.2 Cellular Automata

The availability of massively parallel computers, has motivated the use of cellular automata on large lattices for obtaining solutions to P.D.E., in particular the incompressible Navier Stokes equations. A lot of work is necessary to justify this approach.

We describe here a model due to B.M. BOGHOSIAN, C.D. LEVERMORE,[5]. See also U. FRISCH, B. HASSLACHER, Y. POMEAU, [13].

Consider Burgers' equation :

$$\frac{\partial u}{\partial t} + c \frac{\partial}{\partial x} \left( u - \frac{u^2}{2} \right) = \nu \frac{\partial^2 u}{\partial x^2}$$

$$\begin{array}{ll} \text{Replacing } \frac{\partial u}{\partial t} & \text{by } \frac{u(x, t + \Delta t) - u(x, t)}{\Delta t} \\ \text{Replacing } \frac{\partial}{\partial x} & \text{by } \frac{1}{2} \frac{u(x + \Delta x, t) - u(x - \Delta x, t)}{\Delta x} \\ \text{Replacing } \frac{\partial^2 u}{\partial x^2} & \text{by } \frac{1}{\Delta x^2} (u(x + \Delta x, t) + u(x - \Delta x, t) - 2u(x, t)) \end{array}$$

and choosing  $\Delta t, \Delta x$  such that  $2\nu = \frac{\Delta x^2}{\Delta t}$ , we obtain the discretization scheme

$$\begin{aligned} u(x, t + \Delta t) &= \frac{1 - \frac{c}{2\nu} \Delta x}{2} u(x + \Delta x, t) + \frac{1 + \frac{c}{2\nu} \Delta x}{2} u(x - \Delta x, t) \\ &+ \frac{c \Delta x}{8\nu} (u^2(x + \Delta x, t) - u^2(x - \Delta x, t)) \end{aligned}$$

This can be simulated "approximately" by the stochastic process :

$$\begin{aligned} \xi_1(x + \Delta x, t + \Delta t) &= \frac{1 + w(x, t)}{2} (\xi_1(x, t) + \xi_2(x, t)) \\ &- w(x, t) \xi_1(x, t) \xi_2(x, t) \\ \xi_2(x - \Delta x, t + \Delta t) &= \frac{1 - w(x, t)}{2} (\xi_1(x, t) + \xi_2(x, t)) \\ &+ w(x, t) \xi_1(x, t) \xi_2(x, t) \end{aligned}$$

where  $\xi_1, \xi_2$  take the values 0, 1,  $w$  is random and takes the values -1 or 1. The random variables are independent and :

$$Ew = \frac{c}{2\nu} \Delta x$$

It can be proved that :

$$u(x, t) \sim E(\xi_1 + \xi_2)$$

The process  $\xi_1, \xi_2$  is a cellular automata which can be simulated on a *massively parallel computer*.

Research on similar types of stochastic processes is important in the context of solving nonlinear P.D.E. on massively parallel machines.

## 2 CONTROL, IDENTIFICATION, ESTIMATION.

The applications of these techniques are extremely diversified and come from physical sciences as well as from economic or even social sciences.

We describe some :

- new areas of applications
- new algorithms
- new approaches .

### 2.1 *New areas of application*

#### 2.1.1 *Environmental studies. The program "Global Change"*

In view of the growing importance of environmental issues, a worldwide program of research has been developing in recent years, under the name of "Global Change". It connects specialists of Climate Dynamics, Oceanography, Planetary Physics, . . . It seems that this direction is a source of important mathematical problems, of somewhat new nature.

The basic problem deals with the *prediction* of physical quantities, solutions of a set of nonlinear evolution P.D.E., with *unknown* parameters and unknown initial state. Nonlinearity creates an important sensitivity with respect to initial data and unknown quantities, resulting in a lack of predictability beyond some length of time. A fundamental question is to identify the *important regimes* of the physical variables, those which contain the main futures of interest and are *persistent*. There are several ways to give a mathematical meaning to this question. The interesting feature is that they result in a *mixture of statistical and dynamical methods*. A lot of work is needed in that direction, even for simple nonlinear systems.

The point of view of dynamical systems is to obtain the stationary solutions of the nonlinear P.D.E. (or system of P.D.E.) and the long-time behaviour of solutions. This is the theory of *attractors*.

A complementary statistical theory has been developed, for which we describe only two ideas, that of *persistent anomalies* and that of *EOF analysis* (Empirical orthogonal functions).

Consider a vector representing physical variables (typically a flow) which is computable through a model, which is not in general completely known (this is an important difficulty, which we leave aside). We represent it by  $\psi_k(t)$ ,  $k = 1 \dots N$  where  $k$  may represent a point  $x_k$  on a grid, or a component if the solution is obtained by an expansion.

We set  $\langle \psi_k \rangle =$  average of  $\psi_k(t)$  over some record of data.

The instantaneous *anomaly* is defined by :

$$\tilde{\psi}_k(t) = \psi_k(t) - \langle \psi_k \rangle$$

The *pattern correlation* between an anomaly at time  $t$  and at a later time  $t + \tau$  is defined by :

$$p(t, \tau) = \frac{\sum_k \tilde{\psi}_k(t) \tilde{\psi}_k(t + \tau) - (\sum_k \tilde{\psi}_k(t)) (\sum_k \tilde{\psi}_k(t + \tau))}{\sigma(t) \sigma(t + \tau)}$$

where

$$\sigma(t)^2 = \sum_k \tilde{\psi}_k(t)^2 - (\sum_k \tilde{\psi}_k(t))^2.$$

We say that an anomaly  $\tilde{\psi}_k(t_0)$  *persists* from  $t = t_0$ , to  $t = t_0 + J\tau$ , if :

$$p(t_j, \tau) \geq p_0, \text{ where } t_j = t_0 + j\tau, j = 0 \dots J - 1$$

and  $p_0$  represents the persistence criterion. What is *expected* is that the anomalies which satisfy a reasonable persistence criterion fall into a *small number of easily identifiable patterns*, related to the attractors of dynamical system.

The EOF analysis goes as follows. Let :

$$\Gamma_{k\ell} = \langle \tilde{\psi}_k \tilde{\psi}_\ell \rangle$$

Consider the eigenvalues of the matrix  $\Gamma$   $\lambda^1 \dots \lambda^N$ , ranked in decreasing order and  $e^1 \dots e^N$  are the corresponding eigenvectors, called the 1st EOF, the 2nd EOF, ...

Next expand the vector  $\tilde{\psi}(t) = (\tilde{\psi}_k(t))$  on the basis  $e^1 \dots e^N$ , hence :

$$\tilde{\psi}(t) = \sum_{i=1}^N \alpha_i(t) e^i$$

then one can easily check that :

$$\langle \alpha_i \alpha_j \rangle = \lambda^i \delta_{ij}.$$

The coefficients  $\alpha_i(t)$  are called the *principal components*. The EOF are interpreted as directions of variability of the anomaly,  $\lambda^i$  representing the part of the variance related to EOF  $e^i$  (the total variance being  $\lambda^1 + \dots + \lambda^N$ ). The important *conjecture* is that the main OEF are related to the patterns associated to persistent anomalies.

In [16] these connections are exhibited experimentally on some models.

Is there a general theory for these phenomenon, at least for some class of nonlinear dynamical systems ? This is an open question, which has a crucial importance for the understanding of the variability of atmospheric dynamics.

### 2.1.2 Computer vision

**2.1.2.1 The segmentation problem** An image can be represented by a function  $g(x)$  measuring the strength of the light signal striking a plane at point  $x$ . Such a function is expected to have discontinuities reflecting edges of objects, and shadows. Outside such lines the function  $g$  is expected to behave more smoothly.

Having this in mind, one defines a segmentation of a region  $\Omega$ , as a set of open connected subsets  $\Omega_i, i = 1 \dots n$ , each one with a piecewise smooth boundary and  $\Gamma$  is the union of the parts of the boundaries of the  $\Omega_i$  inside  $\Omega$ .

An approximation of  $g$  is a function  $u$  which is differentiable on  $\Omega - \Gamma$ . One defines a cost function :

$$J(u, \Gamma) = \mu \int_{\Omega} (u - g)^2 dx + \int_{\Omega - \Gamma} |Du|^2 dx + \nu |\Gamma|$$

The *segmentation problem* consists in minimizing the functional  $J$  over the pair  $(u, \Gamma)$ . Note that if  $\nu = 0, \inf J = 0$ .

This is a new class of problems in the calculus of variations, introduced in [17].

It has attracted a lot of interest and some progress has been made, concerning existence, and approximation.

It is interesting to consider the one dimensional problem, in which  $\Omega = (0, 1), \Gamma = \{a_1; \dots; a_N, \text{ with } 0 < a_1 < a_2 < \dots < a_N < 1\}$  and  $|\Gamma| = N$ . One has :

$$J(u, a_1, \dots, a_N) = \mu \int_0^1 (u - g)^2 dx + \sum_{i=0}^N \int_{a_i}^{a_{i+1}} u'^2 dx + \nu N$$

and we have defined  $a_0 = 0, a_{N+1} = 1$ .

Since we do not impose continuity at points  $a_i$ , we may write preferably :

$$J(u_1, \dots, u_N; a_1, \dots, a_N) = \mu \sum_{i=0}^N \int_{a_i}^{a_{i+1}} [(u_i - g)^2 + u_i'^2] dx + \nu N$$



There is a probabilistic interpretation of  $J$ . Consider in the segment  $(a_i, a_{i+1})$  a process  $x_i$  such that :

$$x_i(t) = x_i(a_i) + w_i(t), \quad t \in [a_i, a_{i+1}[$$

where  $x_i(a_i)$  is not random, and  $w_i(t)$  is a standard Wiener process. We observe on  $(a_i, a_{i+1})$  the process  $y_i(t)$  with :

$$dy_i(t) = x_i(t)dt + db_i \quad y_i(a_i) = 0$$

where  $b_i$  is Wiener process independent from  $w_i$ .

The "a priori probability" of the trajectory  $x_i(t)$  to coincide with a given function  $u_i(t)$  which is  $H^1(a_i, a_i + 1)$  is :

$$\exp -\frac{1}{2} \int_{a_i}^{a_{i+1}} [(u_i'^2 + u_i^2)dt - 2u_i dy_i]$$

For details see [20].

Considering independent processes in each interval, we obtain :

$$\exp -\frac{1}{2} \sum_{i=0}^N \int_{a_i}^{a_{i+1}} [(u_i'^2 + u_i^2)dt - 2u_i dy_i]$$

and the maximization of this probability results in minimizing  $J$ , up to the correspondence  $dy_i \rightarrow g$  on  $(a_i, a_{i+1})$ . It would be extremely interesting to treat the 2 dimensional problem, which is the real one, by similar probabilistic methods. It is an open problem.

*2.1.2.2 Axiomatic derivation of image processing models* We describe here a new approach to image processing due to L. Alvarez, F. Guichard, P.L. Lions and J.M. Morel [4]. Consider the signal  $g(x)$  representing the image. We look at it at a scale  $t$ , measuring roughly speaking the size of details of the image (small  $t$  means fine scale, while large  $t$  means coarse scale). An analysis at scale  $t$  is a transformation  $T_t g$ . A multiscale analysis is thus a family, parametrized by  $t \geq 0$ , of nonlinear operators (or filters).

Of course, some conditions have to be made on the operator  $T_t$ , in order to fulfill physical requirements of the filter. These restrictions or axioms are such that the function  $u(x, t) = (T_t g)(x)$  appears as the solution of a fully nonlinear, parabolic, possibly degenerate second order equation

$$\begin{aligned} \frac{\partial u}{\partial t} &= F(Du, D^2u) \\ u(x, 0) &= g(x). \end{aligned} \tag{4}$$

In fact, the choice of the function  $F$  is equivalent to the choice of the family  $T_t$ . Among physical requirements, one has the following main one

$$F(p, A) \leq F(p, B), \forall p, A \leq B$$

which is in fact the condition which suffices to give a meaning to (4) in viscosity sense.

*Examples :*

- The Gaussian pyramid. It corresponds simply to the heat equation

$$F(p, A) = \frac{1}{2} \text{tr} A$$

- Quasilinear filters :

$$F(p, A) = a(|p|) \text{tr} A + a'(|p|) \frac{(Ap, p)}{|p|}$$

where

$$a \geq 0, \quad a(|p|) + a'(|p|)|p| \geq 0$$

- Morphological filters :

$$F(p) = \inf_{q \in S} p \cdot q$$

where  $S$  is a compact set of  $R^2$ .

- Curvature operators

$$F(p, A) = |p| G\left(\frac{1}{|p|} (\text{tr} A - \frac{Ap \cdot p}{|p|^2})\right)$$

with possible  $G(s) = s$  or  $|s|^{\alpha-1}s$  (in particular  $\alpha = \frac{1}{3}$ ).

*2.1.2.3 Mobile Robotics* Consider the problem of a mobile robot which tries to recover its environment, during its motion (the environment is assumed to be static). The robot is equipped with a camera, which takes images between time intervals. One way of approaching the problem is to extract tokens from the images in the sequence, match them from image to image and recover the motion and the structure of the environment.

Naturally, the tokens we compute in the images should be closely related to objects in the scene, if we want the matches to be meaningful. They are in general surface markings, shadows, depth discontinuities.

Let us explain the general ideas in the case of a point  $M$ , which is the object to be recognized by the mobile robot (see Figure 1). So  $M$  is the real point,  $C_1, C_2$  represent the motion of the camera (installed on the robot),  $m_1, m_2$  the images of  $M$ . The motion is decomposed into a rotation  $R$  with a rotation axis going through  $C_1$ , and a translation  $t = C_1 C_2$ .

If we consider a coordinate system attached to the camera, then we can measure  $C_1 m_1$  and  $C_2 m_2$  with the local coordinate system. The coordinates with respect to a common coordinate system, that related to  $C_1$  are  $C_1 m_1$

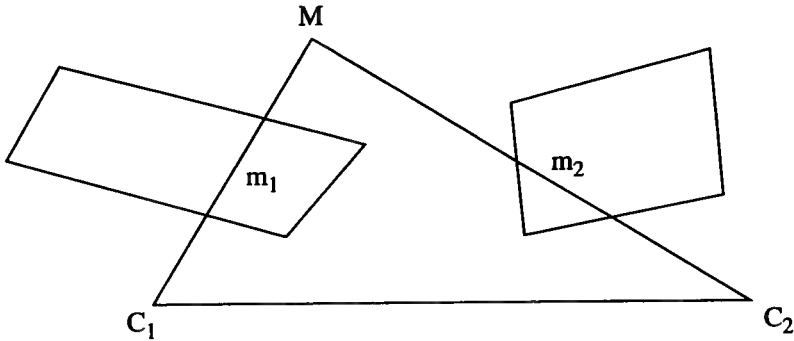


FIGURE 1.

and  $\mathcal{R}C_1m_1$ . Then one expresses the planarity constraint, namely that  $C_1m_1$ ,  $C_2m_2$  and  $t$  are coplanar ; it amounts to :

$$C_1m_1 \cdot (t \wedge \mathcal{R}C_2m_2) = 0.$$

The vector  $t$  has coordinate  $t_x, t_y, t_z$  but from the linearity, we can assume that  $\|t\| = 1$ , hence 2 parameters are enough. The matrix  $\mathcal{R}$  depends of 3 parameters which characterize the unit rotation axis (2 parameters) and the rotation angle.

Conceptually, what is important is to recognize that the previous relations amounts to :

$$f(x, a) = 0$$

where  $a$  is a vector of parameters  $\in \mathbb{R}^n$ , and  $x$  is a vector of measurement  $\in \mathbb{R}^n$  and  $f$  is a nonlinear relation.

Each successive image leads to a relation :

$$f(x_k, a) = 0$$

However the observation is not exact and rather described by the model

$$z_k = x_k + \nu_k$$

where  $\nu_k$  is a white noise of covariance  $\Gamma$ . Considering that

$$a_{k+1} = a_k = a$$

we are in the framework on nonlinear filtering if we can express  $x_k$  as a function of  $a_k$ . It is of course natural to linearize around a given estimate of  $a$ , and to

use extended Kalman filtering. Once  $t, R$  is obtained, one can recover  $M$  by expressing the relations :

$$\lambda C_1 m_1 = t + \mu R C_2 m_2$$

where  $\lambda, \mu$  are unknown scalars. In this relation again  $t, R$  are known random variables, as well as  $C_1 m_1, C_2 m_2$ . Thus we are in a situation similar to the above and can use again a Kalman filter.

These techniques have been extensively used in the context of mobile robotics by O. FAUGERAS and his team, see for instance [11].

## 2.2 New algorithms

### 2.2.1 Parallel algorithms

The development of multiprocessors has generated a substantial interest in the obtaining of *parallel algorithms*. A thorough analysis is needed, since surprises can arise in comparison with the sequential approach.

Take for instance Jacobi and Gauss Seidel iterations for obtaining a fixed point of :

$$x = f(x) \quad x \in \mathbb{R}^n$$

A Jacobi iteration is the following :

$$x_i^{k+1} = f_i(x^k), \quad i = 1 \dots n$$

and a Gauss Seidel is :

$$x_i^{k+1} = f_i(x_1^{k+1}, \dots, x_{i-1}^{k+1}, x_i^k, \dots, x_n^k)$$

The advantage of Gauss Seidel iteration is that it converges more frequently than Jacobi, and sequentially it performs much better (the convergence rate of Gauss Seidel iteration is better).

Parallel implementation will change the situation considerably.

Consider the case when there are  $n$  processors, and the sequence  $x^k$  such that :

$$x^{k+1} = f(x^k)$$

denoted by  $x^{k,U}$  (Jacobi sequence) converges towards the fixed point. Suppose also  $f$  monotone, i.e.  $f(x) \leq f(y) \forall x, y$  with  $x \leq y$ . Then take a sequence  $x^{k,U}$  defined by :

$$\begin{aligned} x_i^{k+1,U} &= f_i(x^{k,U}), & \forall i \in U_k \\ x_i^{k+1,U} &= x_i^{k,U}, & \forall i \notin U_k \end{aligned}$$

$U_k$  is a subset of  $\{1, \dots, n\}$ .

One can prove (T.N. TSITSIKLIS) that if one starts with the same initial value  $x^0$  and  $f(x^0) \leq x^0$  or  $x^0 \leq f(x^0)$ , then :

$$x^* \leq x^{k,J} \leq x^{k,U}, \quad \forall k$$

where  $x^*$  is the limit fixed point. Hence Jacobi iteration performs better than any parallel version of Gauss Seidel iteration. When they are less than  $n$  processors available, or the assumption of monotonicity is not satisfied, no general statement can be made (see [4])

### 2.2.2 Simulated Annealing and global optimization

This type of algorithm has been developed in the recent years in order to obtain a *global minimum* for a function  $U(x)$ , over  $x \in B$ ,  $B$  compact, in the case when  $U$  is smooth. It is clear that such a problem occurs in many applications. Simulated annealing has first been used in the context of image processing.

The algorithm consists in a discrete version of the following stochastic differential equation:

$$dx_t = -DU(x_t)dt + c_t \sigma(x_t)dw_t, \quad x(0) = x$$

where the following assumptions are made

- $U$  is  $C^2$  from  $B$  to  $[0, \infty)$  and

$$\text{Min}_{x \in B} U(x) = 0, \quad DU(x) \cdot x > 0, \quad \forall x \in B - \overset{\circ}{B}_1$$

where  $B$  is a ball in  $\mathbb{R}^n$ , centered at the origin, and  $B_1$  is an other ball, also centered at the origin and strictly included in  $B$ .

- $\sigma$  is Lipschitz continuous from  $B$  to  $[0, 1]$ , with  $\sigma = 1$ , for  $x \in B_1$ ,  $\sigma = 0$  for  $x \in \partial B$ ,  $\sigma > 0$  on  $\overset{\circ}{B}$
- $c_t = \frac{c}{\text{Log}t}$ , for  $t$  large,  $c > 0$ .
- $w_t$  standard Wiener process in  $\mathbb{R}^n$
- $\pi^\varepsilon(x) = \frac{1}{Z^\varepsilon} \left( \exp - \frac{2U(x)}{\varepsilon^2} \right) \mathbf{1}_B$  with  $\int \pi^\varepsilon(x) dx = 1$  converges weakly to a probability  $\pi$  as  $\varepsilon \rightarrow 0$ .

Note that  $\pi$  is a probability concentrated on the set of global minima of  $U(\cdot)$ .

Then the following result can be proved :

$$Ef(x_t) \rightarrow \pi(f)$$

$\forall f$  bounded, continuous, as  $t \rightarrow \infty$ , uniformly for  $x$  (the initial value) in  $B$ . (For more details see [7]).

### 2.3 New approaches

Let us just mention the developments related to  $H_\infty$  theory and which permit to obtain protection of dynamic systems from disturbances via *feedback control*. We just mention some recent results concerning linear systems.

Let us consider the linear system

$$\begin{aligned} \dot{x} &= Ax + Bu + Dw, \\ y &= Cx \end{aligned}$$

where  $w$  represents a disturbance, and  $u$  a control. We consider *feedback* controls,  $u = Ky$ . The *transfer* matrix  $T_K(s)$  is given by

$$T_K(s) = C[sI - (A + BK)]^{-1}D$$

and we consider those  $K$  for which  $A + BK$  is stable. The  $H_2$  norm is defined by :

$$\|T_K\|_2 = \left( \frac{1}{2\pi} \int_{-\infty}^{+\infty} \text{tr } T_K(-j\omega)^* T_K(j\omega) d\omega \right)^{\frac{1}{2}}$$

and the  $H_\infty$  norm is defined by :

$$\|T_K\|_\infty = \sup_{\omega \in R} (\text{tr } T_K(-j\omega)^* T_K(j\omega))^{\frac{1}{2}}$$

which are finite since  $A + BK$  is stable.

The problem of  $H_\infty$  or  $H_2$  control *consists* in minimizing the above norms with respect to  $K$ .

Note that

$$\|T_K\|_\infty = \sup_w \left\{ \left( \int_0^\infty |y(t)|^2 dt \right)^{\frac{1}{2}} \left| \left( \int_0^\infty |w(t)|^2 dt \right)^{\frac{1}{2}} \leq 1 \right. \right\}$$

and thus this norm expresses the *sensitivity of the system* with respect to external disturbances.

Among the important results obtained recently, it has been proven that *we can chose* a  $K$  such that  $\|T_K\|_\infty \leq \gamma, \forall \gamma$  given, *if* there exists  $\varepsilon$  such that one can solve the Riccati equation

$$PA + A^*P - \frac{1}{\varepsilon} PBB^*P + \frac{1}{\gamma} PDD^*P + \frac{1}{\gamma} CC^* + \varepsilon I = 0$$

In fact  $K = -\frac{B^*P}{2\varepsilon}$  will serve for this purpose (for more details, see [14]).

### 3 DISCRETE SYSTEMS

#### 3.1 Discrete event systems

New applications strongly related to information technology have created the need to develop a theory of DEES, *discrete event dynamic systems*. Such applications are production or assembly lines, computer/communication networks, traffic systems, ... A special issue of IEEE, Jan. 1989 is devoted to dynamics of discrete event systems.

Many new mathematical techniques have been developed in this context. We describe here one of them, the use of an algebraic structure, called *dioid*, in the modelling of *timed event graphs*.

Let us just recall the basic definition of a dioid. It is a set  $\mathcal{D}$  provided with two inner operations  $\oplus$  and  $\otimes$  (addition and multiplication) such that

- they are both associative
- addition is commutative
- multiplication is right distributive with respect to addition
- there exists a null and identity elements

$$\begin{aligned}\exists \varepsilon \in \mathcal{D} : \forall a \in \mathcal{D}, & \quad a \oplus \varepsilon = a \\ \exists e \in \mathcal{D} : \forall a \in \mathcal{D}, & \quad a \otimes e = e \otimes a = a\end{aligned}$$

- the null element is absorbing

$$\forall a \in \mathcal{D}, \quad a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon$$

- the addition is idem potent

$$\forall a \in \mathcal{D}, \quad a \oplus a = a.$$

When addition is commutative, the dioid is called commutative. As an example take  $\mathcal{D} = \mathbb{Z} \cup \{-\infty\} \cup \{+\infty\}$  and

$$\begin{aligned}\oplus &= \max, & \otimes &= + \\ \varepsilon &= -\infty, & e &= 0\end{aligned}$$

(note that we impose the rule  $(-\infty) \otimes (+\infty) = (-\infty)$ ).

We can also consider

$$\begin{aligned}\oplus &= \min, & \otimes &= + \\ \varepsilon &= +\infty, & e &= 0\end{aligned}$$

(in which case  $(-\infty) \otimes (+\infty) = +\infty$ ).



A dioid is a structure somewhere between linear algebra and lattices.  
 One can define a partial order relation

$$a \geq b \Leftrightarrow a = a \oplus b$$

and a *pseudo left inverse* denoted  $a \setminus c$  which is the greatest subsolution of

$$a \otimes x = c.$$

Starting with these premises affine equations can be solved, as well as matrices defined and a matrix calculus is available. Matrix equations can also be solved. Let us see briefly how these concepts apply to timed event graphs.

Times event graphs are a special kind of Petri nets. They are directed graphs with two types of edges, *places* and *transitions*

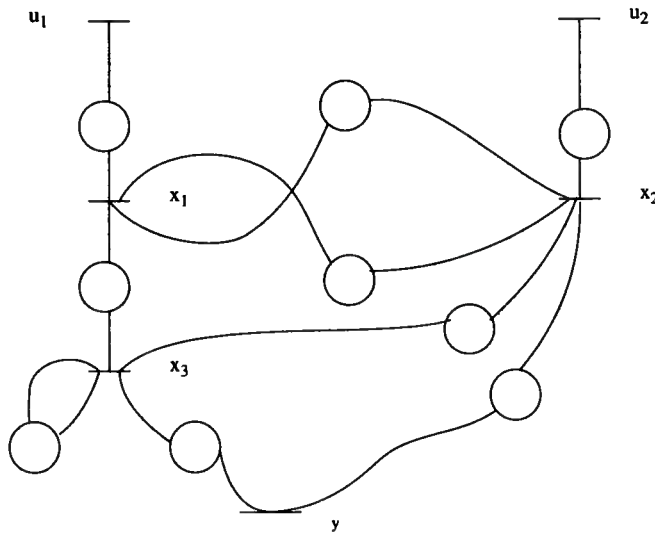


FIGURE 2.

In Figure 2,, the transitions are  $u_1, u_2, x_1, x_2, x_3, y$  and the places are denoted by  $x_1|u_1, x_2|u_2, x_3|x_1, x_3|x_2, x_3|x_3, y|x_3, y|x_2, x_1|x_2, x_2|x_1$ .

There is a single transition upstream and downstream, at each place.

In places, there are tokens or not. Tokens are created or consumed when transitions are fired, more precisely when a transition  $t$  is fired one token is consumed at each place which precedes  $t$  and one is created at each place which succeeds it.

Let us assume that transitions are immediate, but a token must stay at a place an amount of time called the holding time, which depends on the place. The following symbols are used

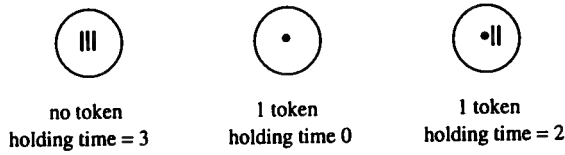


FIGURE 3.

For instance consider the places which precede  $x_1$ , we complete the information as follows

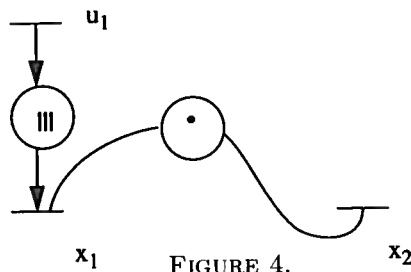


FIGURE 4.

Let for a transition  $x$ ,  $x_n$  be date at which transition  $x$  has been fired for the  $n^{\text{th}}$  time. We can write the relation

$$(x_1)_n = \max[(x_2)_{n-1}, (u_1)_n + 3]$$

and of course similar relations for other transitions.

If the dates take values in  $Z \cup \{+\infty\} \cup \{-\infty\}$ , then we can work with the dioid considered above  $\mathcal{D}$ , with the operations  $\oplus = \max$ ,  $\otimes = +$ .

The preceding relation writes

$$(x_1)_n = (x_2)_{n-1} \oplus 3(u_1)_n$$

where  $3(u_1)_n = 3 \otimes (u_1)_n$  to simplify the notation.

One of the objectives of research in these directions is to obtain a theory similar to that of linear dynamic systems. In particular a theory of *stability* is being developed. This is important to obtaining an evaluation of performances for the real system which is modelled by the event graph. (See [8]).

### 3.2 Hybrid systems

An hybrid system is a system whose state contains continuously as well as finitely valued variables. When the discrete variables take a given set of values, the continuous variables behave as the solutions of differential equations. Transitions between the possible sets of values of the discrete variables are obtained through the action of a monitor (a controller). The action of the controller may be instantaneous or require some delay. The objective is in general to keep the continuous variables within a given range. Decisions are taken as feedbacks.

An hybrid system will be characterized by a given feedback, and the problem is to prove that this feedback rule fulfills the goal.

EXAMPLE 1 Suppose we want to control the temperature of a room through a thermostat, which can turn instantaneously a heater on and off. The temperature is the continuous variable  $x(t)$ ,  $\nu(t) = 1$  or  $0$  according whether the heater is on or off is the discrete variable. We have :

$$\begin{aligned} \dot{x} &= -Kx & \text{if } \nu &= 0 \\ \dot{x} &= K(h - x) & \text{if } \nu &= 1 \end{aligned}$$

If  $d(t)$  is the decision taken by the thermostat,  $d(t) = 1$  or  $0$  and we have:

$$\nu(t+0) = d(t)$$

We want to maintain  $x(t)$  between  $m$  and  $M$ . Then we take

$$\begin{aligned} d(t) = 1 & \quad \text{if } x(t) = m \quad \text{and} \quad \nu(t) = 0 \\ d(t) = 0 & \quad \text{if } x(t) = M \quad \text{and} \quad \nu(t) = 1 \end{aligned}$$

and  $d(t) = \nu(t)$  otherwise . Such a feedback fullfills the objective.

EXAMPLE 2 Suppose we control the water level in a tank through a monitor which can turn a pump on and off. The water level is  $x(t)$ , and we set  $\nu(t) = 1$  if the pump is on, and  $\nu(t) = 0$  if it is off. We have

$$\begin{aligned} \dot{x} &= -2 & \text{if } \nu(t) &= 0 \\ \dot{x} &= 1 & \text{if } \nu(t) &= 1 \end{aligned}$$

Let  $d(t)$  be the decision taken by the monitor,  $d(t) = 1$  or  $0$  and suppose there is a delay of 2 before the decision is executed then :

$$\nu(t) = d(t - 2)$$

We wish to keep the water level between 1 and 12. We then consider the feedback

$$\begin{aligned} d(t) = 1 & \quad \text{if} \quad x(t) = 5 \text{ and } \nu(t) = 0 \\ d(t) = 0 & \quad \text{if} \quad x(t) = 10 \text{ and } \nu(t) = 1 \end{aligned}$$

and  $d(t) = \nu(t)$  otherwise such a feedback fulfills the desired behaviour. In general, proving that a specific feedback satisfies a given objective of the continuous variables is not easy. Results on decidability of such a problem are available for a particular class of Hybrid systems (cf. R. Alur et al. [1]).

#### 4 NEW AREAS OF INFORMATION TECHNOLOGY

Let us mention only some recent mathematical problems motivated by I.T. (again this is by no means exhaustive).

##### 4.1 Artificial intelligence

Since artificial intelligence needs to deal with *qualitative* aspects, more than with quantitative aspects (or in connection with them), this has motivated the development of *qualitative simulation* (or *qualitative physics*) in particular at Xerox Parc. Note that the economists needed much before similar techniques, in the context of the theory of *comparative economics* (P.A. SAMUELSON).

Our presentation here relies on some recent work of J.P. AUBIN.

We pose the problem of the *qualitative evolution* of solutions to a differential equation

$$\dot{x} = f(x_t) \quad x \in \mathbb{R}^n$$

and more precisely to the *qualitative evolution* of a set of functionals

$$V_1(x_t), \dots, V_m(x_t)$$

which are of importance (energy, entropy, indicators, ...).

The qualitative behavior is expressed by the evolution of the functions  $\text{sign} \left( \frac{d}{dt} V_j(x_t) \right)$  with values in  $\mathcal{R}^m = \{-1, 0, +1\}^m$ .

This is the problem of interest. But we want to obtain this evolution, without solving the equation, since some independence should be obtained with respect to the *initial condition*.

Since  $\text{sign} \left( \frac{d}{dt} V_j(x_t) \right) = \text{sign} (DV_j(x_t)f(x_t))$  it is convenient to introduce in the closed subspace  $K$  of  $\mathbb{R}^n$ , where lives  $x_t$ , the *qualitative cells*

$$K_a = \{x \in K \mid \text{sign} (DV_j(x)f(x)) = a_j\}$$

where  $a \in \mathcal{R}^m$ , and their closure (*large qualitative cells*)

$$\bar{K}_a = \{x \in K \mid \text{sign} (DV_j(x)f(x)) = a_j \text{ or } 0\}.$$

Let  $\mathcal{D}(f, V)$  be the subset of qualitative states  $a$  such that  $\bar{K}_a$  is not empty. Let also denote by  $x(t; x_0)$  the solution of the differential equation corresponding to an initial date  $x_0$ . One is interested in the study of *transitions between qualitative cells*.

If  $b \in \mathcal{D}(f, V)$ , we say that  $c \in \mathcal{D}(f, V)$  is a *successor* of  $b$ , if  $\forall x_0 \in \bar{K}_b \cap \bar{K}_c$ , there exists  $\tau > 0$ , such that  $x(t; x_0) \in \bar{K}_c$ , for all  $t \in ]0, \tau[$ .

A qualitative state  $a$  is a *qualitative equilibrium*, if it is its own successor. It is said to be a *qualitative repellor* if  $\forall x_0 \in \bar{K}_a$ , there exists  $t > 0$  such that  $x(t; x_0) \notin \bar{K}_a$ .

The theory developed by J.P. Aubin permits to characterize the map of successors, the qualitative equilibria, and the qualitative repellors.

It has been applied to the so-called “replicator systems”, a prototype of which is the differential system ([2])

$$\dot{x}_i = x_i \left( \alpha_i - \sum_{j=1}^n \alpha_j x_j \right)$$

#### 4.2 Neural networks

The basic neural network can be viewed as an undirected graph with  $n$  nodes, to which are attached a pair  $(W, \theta)$  where

$W$  is an  $n \times n$  symmetric matrix,  $W_{ij}$  is the weight attached to the edge  $(i, j)$ ,  $W_{ii} = 0$

$\theta$  is an  $n$  vector,  $\theta_i$  is the threshold attached to the node  $i$ .

Nodes are called *neurons*. Each neuron has two possible states  $(1, -1)$ . Let  $v$  be the state of the neural network,  $v_i$  being the state of neuron  $i$ .

Let

$$E_i(v) = - \sum_{j=1}^n W_{ij} v_j + \theta_i$$

then the following calculation is performed by the network

$$\begin{aligned} v_i^{k+1} &= \text{sign}(E_i(v^k)), \quad \text{for } i \in S^k \\ v_i^{k+1} &= v_i^k \text{ for } i \notin S^k \end{aligned}$$

where  $S^k$  is a subset of the neurons.

For instance if

$$k = hn + j \quad j = 0 \dots n - 1$$

and  $S^k = \{j + 1\}$ , the *network operates in serial mode*.

Note that in our notation

$$\text{sign}(a) = \begin{cases} 1 & \text{if } a \geq 0 \\ -1 & \text{if } a < 0 \end{cases}$$

A *stable* state is a state such that

$$v^{k+1} = v^k = v.$$

A basic theorem of HOPFIELD is that if the network operates in serial mode, then it will converge to a *stable state*.

The applicability of neural networks in practice arises from the possibility of interpreting the stable states. For instance, in pattern recognition, the stable states are known patterns, and for a given input pattern, the network will converge to the known pattern which is the closest to the input. It is clear that the neural network realizes the following search problem

$$\min E(v) = -\frac{1}{2} \sum_{ij} W_{ij} v_i v_j + \sum_i \theta_i v_i \quad v_i = \{-1, +1\}$$

and attains a *local* minimum.

One can clearly consider many variants of the above problem. For instance consider the following model in continuous time

$$\begin{aligned} v_i(t) &= g(u_i(t)) \\ \frac{du_i}{dt} &= -E_i(v(t)) \end{aligned}$$

where  $g$  is an increasing function from  $R$  to  $[0, 1]$  and  $E_i(v) = \frac{\partial}{\partial v_i} E(v)$ ,  $E(v)$  energy function (for instance the above). It will converge towards a local minimum of  $E(v)$ . It can be realized as an analog integrated circuit.

In the spirit of *simulated annealing*, considered above, one can try to attain a *global* minimum of the Energy function, by considering a stochastic version of the preceding model. This has been done by E. WONG.

Consider the model

$$\begin{aligned} v_i(t) &= g(u_i(t)) \\ du_i &= -E_i(v(t))dt + \sqrt{\frac{2T}{g'(u_i(t))}} dw_i \end{aligned}$$

where  $T$  is a constant, and  $w_i$  are independent standard Wiener processes. The stationary probability density of the process  $v(t)$  is

$$p(v) = \frac{1}{Z} \exp -\frac{1}{T} E(v)$$

where  $Z$  is the normalization factor.

The simulated annealing adaptation of the preceding algorithm (for instance take  $T(t) \rightarrow 0$  as  $t \rightarrow \infty$ ) remains to be done. For more details, see [6] and [19].

### 4.3 Analytic analysis of algorithms

Computer science leads quite frequently to combinatorial algorithms. A quite interesting approach of P. Flajolet [12] has shown how generating functions and complex analysis provide a way to treat these problems. In particular, formal languages, tree enumerations, comparison based searching and sorting, digital structures, hashing and occupancy have been interesting applications.

A class of combinatorial structures is a pair of a finite or denumerable set  $\mathcal{A}$ , whose elements are called the atoms.

Each atom  $\alpha \in \mathcal{A}$  will have a size  $|\alpha|$ . We can perform the following operations :

The product relation  $\mathcal{C} = \mathcal{A} \times \mathcal{B}$  :

$$\mathcal{C} = \{\gamma \in \mathcal{C} | \gamma = (\alpha, \beta), \alpha \in \mathcal{A}, \beta \in \mathcal{B}\} \quad \text{with} \quad |\gamma| = |\alpha| + |\beta|$$

The union relation  $\mathcal{C} = \mathcal{A} + \mathcal{B} \quad \mathcal{C} = \mathcal{A} \cup \mathcal{B}$  where  $\mathcal{A} + \mathcal{B}$  are disjoint.

The sequence  $\mathcal{C} = \mathcal{A}^*$

$$\mathcal{C} = \{\varepsilon\} + \mathcal{A} + \mathcal{A} \times \mathcal{A} + \mathcal{A} \times \mathcal{A} \times \mathcal{A} + \dots$$

where  $|\varepsilon| = 0$ . The set construction  $\mathcal{C} = \mu(\mathcal{A})$ , is the collection of all subsets of  $\mathcal{A}$ :

$$\mathcal{C} = \{\{\alpha_1, \dots, \alpha_k, \dots\} \mid \alpha_1, \dots, \alpha_k, \dots \text{ in } \mathcal{A}, \alpha_1, \dots, \alpha_k, \dots \text{ different}\}.$$

The multi set construction  $\mathcal{C} = M(\mathcal{A})$  allows repetitions.

The cycle construction  $\mathcal{C} = \mathcal{C}(\mathcal{A})$  is the set whose elements are (non empty) cycles of elements of  $\mathcal{A}$ .

Let  $A_n$  be the number of elements of  $\mathcal{A}$ , whose elements are (non empty) cycles of elements of  $\mathcal{A}$ . Let  $C_n$  be the number of elements of  $\mathcal{C}$ , whose size is  $n$ , then the interesting problem is to calculate the  $C_n$  corresponding to the more complex structure  $\mathcal{C}$ . This is where the generating functions are useful. Define

$$A(z) = \sum_n A_n z^n = \sum_{\alpha \in \mathcal{A}} z^{|\alpha|}$$

and

$$C(z) = \sum_n C_n z^n = \sum_{\gamma \in \mathcal{C}} z^{|\gamma|}$$

It is possible to express  $C(z)$  in function of  $A(z)$ . For instance, for  $\mathcal{C} = \mathcal{A} \times \mathcal{B}$  one has :

$$C(z) = \sum_{(\alpha, \beta) \in \mathcal{A} \times \mathcal{B}} z^{|\alpha| + |\beta|} = A(z)B(z)$$



For  $\mathcal{C} = \mathcal{A} + \mathcal{B}$

$$C(z) = \sum_{\alpha \in \mathcal{A}} z^{|\alpha|} + \sum_{\beta \in \mathcal{B}} z^{|\beta|} = A(z) + B(z)$$

For  $\mathcal{C} = \mathcal{A}^*$

$$\begin{aligned} C(z) &= 1 + A(z) + A(z)^2 + \dots \\ &= \frac{1}{1 - A(z)} \end{aligned}$$

For  $\mathcal{C} = \mu(\mathcal{A})$ , we note that

$$\mu(\mathcal{A}) = \prod_{\alpha \in \mathcal{A}} (\{\varepsilon\} + \{\alpha\})$$

hence

$$\begin{aligned} C(z) &= \prod_{\alpha \in \mathcal{A}} (1 + z^{|\alpha|}) = \prod_n (1 + z^n)^{A_n} \\ &= \exp(A(z) - \frac{A(z^2)}{2} + \frac{A(z^3)}{3} + \dots) \end{aligned}$$

For  $\mathcal{C} = M(\mathcal{A})$  we have

$$M(\mathcal{A}) = \prod_{\alpha \in \mathcal{A}} \{\alpha\}^*$$

hence

$$\begin{aligned} C(z) &= \prod_{\alpha \in \mathcal{A}} \frac{1}{1 - z^{|\alpha|}} = \prod_n (1 - z^n)^{-A_n} \\ &= \exp(A(z) - \frac{A(z^2)}{2} + \frac{A(z^3)}{3} + \dots) \end{aligned}$$

Consider further  $\mathcal{C} = M_2(\mathcal{A})$ , the collection of subsets of  $\mathcal{A}$  with cardinality 2, with possible repetition. Then

$$\begin{aligned} C(z) &= \sum_{|\alpha_1| > |\alpha_2|} z^{|\alpha_1| + |\alpha_2|} + \sum_{\alpha} z^{2|\alpha|} \\ &= \frac{1}{2} \sum_{|\alpha_1| \neq |\alpha_2|} z^{|\alpha_1| + |\alpha_2|} + \sum_{\alpha} z^{2|\alpha|} \\ &= \frac{1}{2} \sum_{\alpha_1, \alpha_2} z^{|\alpha_1| + |\alpha_2|} + \frac{1}{2} \sum_{\alpha} z^{2|\alpha|} \end{aligned}$$

hence

$$C(z) = \frac{1}{2}(A(z))^2 + \frac{1}{2}A(z^2).$$

From the previous structures, it is possible to construct further complex structures, which will lead to functional equations. For instance, consider in Figure 5 the structure of binary trees (the size of a binary tree is the number of leaves)

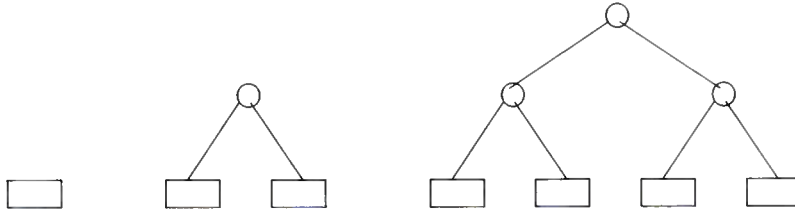


FIGURE 5.

Let  $\mathcal{A} = \{\text{leaf}\}$ , then

$$\mathcal{B} = \mathcal{A} + \mathcal{B} \times \mathcal{B}$$

hence

$$B(z) = z + (B(z))^2$$

which yields a unique formal power series solution

$$B(z) = \frac{1}{2}(1 - \sqrt{1 - 4z})$$

Similarly consider trees with multiples branches (at least 2), each branch having at least 2 leaves, one has

$$\mathcal{B} = \mathcal{A} + \mathcal{B} \times \mathcal{B} + \mathcal{B} \times \mathcal{B} \times \mathcal{B} + \dots$$

hence

$$B(z) = z + \frac{(B(z))^2}{1 - B(z)}$$

which obtains

$$B(z) = \frac{1}{4}(1 + z - \sqrt{1 - 6z + z^2}).$$

Formulas like (1) allow among other things to study the asymptotic behavior of  $B_n$ . This is governed by the singularities of the generating function  $B(z)$ , according to a famous theorem of Darboux.

Suppose we consider the class of mathematical expressions involving constants, the variable  $x$ ,  $e^x$  and additions or products of similar type of expressions. We can visualize the set of such expressions by :

$$\mathcal{E} = \{c\} \cup \{x\} \cup \left\{ \begin{array}{c} / \\ \varepsilon \end{array} + \begin{array}{c} \backslash \\ \varepsilon \end{array} \right\} \cup \left\{ \begin{array}{c} / \\ \varepsilon \end{array} \times \begin{array}{c} \backslash \\ \varepsilon \end{array} \right\} \cup \left\{ \begin{array}{c} \exp \\ | \\ \varepsilon \end{array} \right\}$$

This permits to represent an element of  $\mathcal{E}$  as a tree, for instance the expression  $x + e^{e^x+x}$  is represented by Figure 6. The size of an expression will be the

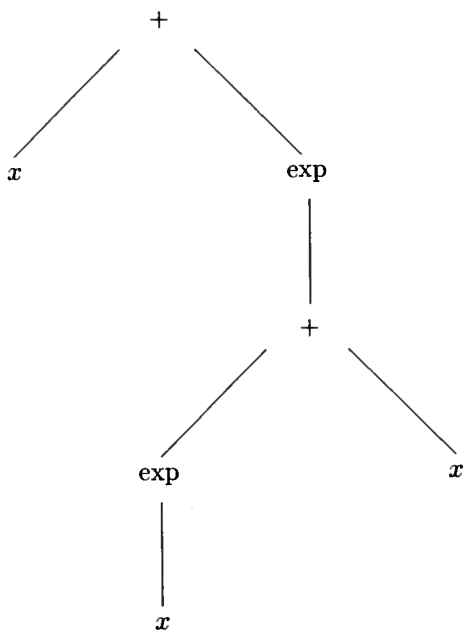


FIGURE 6.

number of nodes of the tree. The above tree has size 7.

Let  $E(z)$  to be the generating function corresponding to  $\mathcal{E}$ , then we have the functional equation

$$E(z) = 2z + 2zE(z)^2 + zE(z)$$

Let  $[z^n]E(z)$  to be the coefficient of  $z^n$  in the formal series  $E(z)$ , it represents the number of expressions of size  $n$ .

Among possible uses of this machinery, one can compute the complexity of formal differentiation. One can estimate the asymptotic average size of derivatives. Many more applications can be given.

#### REFERENCES

1. R. ALUR et al, *The algorithmic Analysis of Hybrid Systems*, Theoretical Computer Science, (Jan. 1995)
2. J.P. AUBIN, *Mathematical Methods of Artificial Intelligence*, to be published.

3. L. ALVAREZ, F. GUICHARD, P.L. LIONS, J.M. MOREL, *Axioms and Fundamental Equations of Image Processing*, Arch. Rational Mech. Anal. (1993).
4. D.P. BERTSEKAS, J.N. TSITSIKLIS, *Parallel and Distributed Computation: Numerical Methods*, Prentice Hall, Englewood Cliffs, N.J. 1989.
5. B.M. BOGHOSIAN, C.D. LEVERMORE, *Complex Systems* 1 (1987).
6. J. BRUCK, J. SANZ, *A study on neural networks*, International Journal of intelligent systems, vol. 3, 59-75, (1988).
7. CHIANG T.S., HWANG C.R., SHEU S.J., *Diffusion for global optimization in  $\mathbb{R}^n$* , SIAM Control, 25, pp. 737-752, 1987.
8. G. COHEN, P. MOLLER, J.P. QUADRAT, M. VIOT, *Algebraic tools for the performance evaluation of discrete event system*, Proceedings IEEE, special issue of dynamics of discrete event systems, Jan. 1989.
9. I. DAUBECHIES, *Orthonormal Bases of Compactly supported Wavelets*, CPAM, 1988.
10. E. DEAN, R. GLOWINSKI, C.H. LI : *Supercomputer solutions of P.D.E. problems in computational fluid dynamics and in control*, University of Minnesota, Supercomputer Institute.
11. O. FAUGERAS, *A few steps towards artificial 3D Vision*, INRIA, Technical Report series, Feb. 88, N°790.
12. P. FLAJOLET, *Analytic Analysis of Algorithms* Lecture Notes in Computer Science, Vol. 623, Springer Verlag, (1992).
13. U. FRISCH, B. HASSLACHER, Y. POMEAU, *Lattice Gas Automata for the Navier Stokes Equation*, Physical Review Letters, 1986.
14. P.P. KHARGONEKAR, I.R. PETERSEN, M. ROTEA,  *$H_\infty$  Optimal Control with State Feedback* IEEE Trans. Automatic Control, 1988.
15. Y. MEYER, *Wavelets and Operators*, Book to appear.
16. K.C. MO and M. GHIL, *Statistics and Dynamics of Persistent Anomalies*, Journal of the Atmospheric Sciences, March 1987, .
17. D. MUNFORD, J. SHAH, *Optimal Approximation by piecewise smooth functions and associated variational problems*, Communications on Pure and Applied Mathematics, 1988.
18. E.S. ORAN, J.P. BORIS, *Numerical Simulation of Reactive Flow* - Elsevier 1987.
19. E. WONG, *Stochastic neural networks*, ERL, Berkeley, Feb. 89.
20. O. ZEITOUNI, A. DEMBO, *A maximum a Posteriori Estimator for Trajectories of Diffusion Processes*, Stochastics, 1987, Vol. 20.

# A New World Underneath Standard Logic: Cylindric Algebra, Modality and Quantification

Johan van Benthem

*Institute for Logic, Language and Computation*

*University of Amsterdam*

## I. WORKING AT INTERFACES

Cor Baayen's broad interests span at least mathematics, logic, computer science and linguistics. Our paths have crossed on many occasions, starting in the early seventies, when he invited me to talk at his lively mathematics colloquium at the Free University. Through the years, Cor has been a benevolent influential presence in the background, who often came to visit scientific events in our logic community at the University of Amsterdam. It was good to know that the Lord of that fabled Mathematical Centre, though far away in a mythical country, was on our side. We have worked together in various ways – and indeed, when our new research institute ILLC was created in 1991, Cor was the unanimous choice of our mathematicians, philosophers and computer scientists for a distinguished outside board member. It is a great pleasure to be able to express my gratitude for all this on this festive occasion. I would like to add that I have always admired Cor for his personality: deeply honest, compassionate, but penetrating and incisive when needed. People with his qualities are scarce.

But enough by way of fan-mail confessions! My real offering here is a short story about some current logical research at the very interfaces where Cor has been active. Moreover, this story has a direct link with his own early work in mathematics, viz. his spell of cylindric algebra at Berkeley with the Tarski School, which resulted in the papers Baayen 1960, 1962. What I want to show is how current interests in so-called 'dynamic semantics' of information flow for natural and formal languages motivate a reappraisal of 'standard' logical semantics. And some powerful mathematical tools for this analysis can be taken from cylindric algebra. What we discover in this way is a whole landscape of dynamic logics underneath classical predicate logic, some of them very well-behaved (and even decidable). But to see all this, we have to start with the Received View in logic, and see where it can be challenged.

## 2. DECONSTRUCTING TARSKI SEMANTICS

Tarski's well-known semantics for first-order predicate logic has the following key clause explaining the existential quantifier:

$$M, \alpha \models \exists x \phi \quad \text{iff} \quad \text{for some } d \in |M|: M, \alpha^x_d \models \phi .$$

Intuitively, this clause calls a verification procedure: "keep shifting the value of state  $\alpha$  in the register  $x$  until some verifying instance is found for  $\phi$ ". Put differently,

an existential quantifier calls a procedure of random assignment to its designated variable. This is no mere curiosity. The currently emerging program of Dynamic Semantics analyzes any kind of linguistic expression via dynamic 'update conditions', rather than (just) static truth conditions. For natural language, this view is found, amongst others, in Kamp 1984, Barwise 1987, Groenendijk & Stokhof 1991, Van Benthem 1991, Veltman 1991. Its paradigmatic examples are such linguistic processes as anaphora (changing bindings for pronouns across discourse), movement of temporal reference points in narratives, changing presuppositions across texts, and many other aspects of linguistic information flow from speakers/authors to hearers/readers. (A broad survey may be found in Muskens, Van Benthem & Visser 1994.) Independently, and in even greater generality, such dynamic views have been proposed in computer science and cognitive science. For instance, the influential Gärdenfors 1988 explains propositions, not as static assertions, but as transformations of information states. Thus, 'updating' of beliefs includes learning via conditionalizing probability functions, and expansion or revision of data bases. (Both traditions meet in the volume Van Eyck & Visser 1994.) In this paper, we stick with the modest case of variable assignment in quantification.

The above dynamic move will make the semantics of linguistic sentences very much like that of computer programs, viewed in the familiar Hoare-Dijkstra style as binary transition relations between assignments. This semantic perspective is powerful and suggestive, but it has one paradoxical feature. Its complexity is at least as high as that of standard predicate logic – whereas part of the motivation for dynamic semantics is precisely the desire to get at simple computational mechanisms in human language use. Therefore, we should reflect further, and look at the bare bones of state transitions. What makes first-order predicate logic tick at a more abstract computational level? This policy is known from Propositional Dynamic Logic (cf. the new textbook Harel & Kozen 1994), which employs labeled transition systems (poly-modal Kripke models), also a favourite vehicle of mathematical theorizing at CWI concerning computation. Thus, let us see what is really involved in Tarski semantics. The answer is as follows. Much less is needed than the above concrete assignment scheme to give a compositional semantics for first-order quantification (usually taken to be its essential achievement).

The abstract core pattern which makes the semantic recursion work is this:

$$M, \alpha \models \exists x \phi \quad \text{iff} \quad \text{for some } \beta : R_x \alpha \beta \text{ and } M, \beta \models \phi .$$

Assignments  $\alpha, \beta$  are now viewed as abstract states, and the concrete relation  $\alpha =_x \beta$  (which holds between  $\alpha$  and  $\alpha^x_d$ ) has become just any binary update relation  $R_x$ . This greater freedom reflects current developments in Dynamic Semantics, where states can be much more diverse than just assignments (partial assignments, discourse stacks, or yet other data structures) and variable-value update transitions between them may vary accordingly. In this light, 'standard Tarski semantics' amounts to insisting (without explicit argumentation) on one particular set-theoretical implementation. States must be assignment functions in  $\text{IM}^{\text{VAR}}$ , all of which are to be present in our models, and 'variable update' must be the specific indifference relation  $=_x$ .

### 3. A MODAL PERSPECTIVE

The above pattern has a familiar mathematical form. It treats predicate logic as a modal logic, with existential quantifiers  $\exists x$  as existential modalities  $\langle x \rangle$ . This system has the usual possible worlds models  $M = (S, \{R_x\}_{x \in \text{VAR}}, I)$ , with  $S$  a set of 'states',  $R_x$  a binary 'transition relation' for each variable  $x$ , and  $I$  a 'valuation' giving a truth value to atomic formulas  $Px, Rxy, \dots$  in each state  $\alpha$ . Henceforth, our language is the standard first-order one, with predicates and variables (but no function symbols). Some extensions will be considered at the end. Its modal truth definition is as follows:

$M, \alpha \models Px$	iff	$I(\alpha, Px)$
$M, \alpha \models \neg \phi$	iff	<i>not</i> $M, \alpha \models \phi$
$M, \alpha \models \phi \ \& \ \psi$	iff	$M, \alpha \models \phi$ <i>and</i> $M, \alpha \models \psi$
$M, \alpha \models \exists x \phi$	iff	<i>for some</i> $\beta : R_x \alpha \beta$ <i>and</i> $M, \beta \models \phi$ .

The universal validities produced by this general semantics constitute the well-known *minimal modal logic*, whose principles are

- (i) all classical Boolean propositional laws,
- (ii) the axiom of Modal Distribution:  $\exists x (\phi \vee \psi) \leftrightarrow \exists x \phi \vee \exists x \psi$ ,
- (iii) the rule of Modal Necessitation: *if*  $\vdash \phi$ , *then*  $\vdash \neg \exists x \neg \phi$ ,
- (iv) the definition of  $\forall x \phi$  as  $\neg \exists x \neg \phi$ .

A completeness theorem may be proved here using the standard Henkin construction. This poly-modal logic can be analyzed in a standard fashion (Andréka, van Benthem & Némethi 1994 is a modern treatment), yielding the usual meta-properties such as the Craig Interpolation Theorem, and the Los-Tarski Preservation Theorem for submodels. Moreover, it is *decidable*, via any of the usual modal techniques (such as filtration). The model theory of this logic leads to interesting comparisons between 'bisimulations' for its models and 'partial isomorphism' in ordinary model theory (cf. de Rijke 1993). This modal perspective uncovers a whole *fine-structure* of predicate-logical validity. The minimal predicate logic consists of those laws which are 'very valid'. But we can analyze what other standard laws say, too, by the technique of *frame correspondence*. Recall that a modal formula  $\phi$  defines a relational condition  $C$  on state frames if  $\phi$  holds (for all states and interpretation functions) in just those frames where  $C$  obtains. Effective methods exist for finding such conditions, given suitable modal formulas (in particular, the following examples are well-behaved 'Sahlqvist forms'). Here are three illustrations involving key principles from cylindric algebra (cf. Baayen 1960):

- $\phi \ \& \ \exists x \phi \leftrightarrow \phi$  *expresses that*  $R_x$  *is reflexive*
- $\exists x (\phi \ \& \ \exists x \psi) \leftrightarrow \exists x \phi \ \& \ \exists x \psi$  *expresses that*  $R_x$  *is transitive and euclidean.*

These constraints make the  $R_x$  into equivalence relations, as with the modal logic S5. These universal conditions do not impose existence of any particular states in frames. By contrast, the following axiom is existential in nature:

- $\exists x \exists y \phi \leftrightarrow \exists y \exists x \phi$  *expresses that*  $R_x; R_y = R_y; R_x$



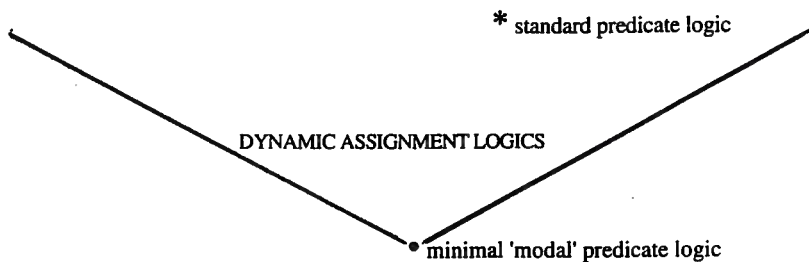
This says that sequences of state changes may be traversed in any order. Abstract state models need not have enough intermediate states to follow all these alternative routes. As a final example, consider another well-known valid quantifier shift:

- $\exists x \forall y \phi \rightarrow \forall y \exists x \phi$  expresses *Confluence of variable update*:  
whenever  $\alpha R_x \beta R_y \gamma$ , there is a state  $\delta$  with  $\alpha R_y \delta R_x \gamma$ .

This is a natural Church-Rosser property of computational processes, whose adoption again has an existential price. Thus, the valid laws of predicate logic turn out to have quite different dynamic content, when analyzed in the light of this broader semantics.

#### 4. THE LANDSCAPE OF DYNAMIC ASSIGNMENT LOGICS

Once again, we are now viewing first-order predicate logic as a dynamic logic for variable assignment, whose atomic computations shift values in registers  $x, y, z, \dots$ . This perspective yields a whole hierarchy of fine-structure underneath standard predicate logic. The latter system merely becomes the (undecidable) theory of one particular class of 'rich assignment models'. The result is a broad semantic landscape of options, rather than one canonical standard. (The same plurality is known in many other areas of logical analysis, witness the case of Modal Logic or Categorical Logic. For a principled defense of this phenomenon, cf. van Benthem 1991.) We have already found a minimal system at the bottom, and standard logic at the top, while intermediate systems arise by imposing varying requirements on assignments and updates  $R_x$ :



In this landscape, we want to find expressive logics that share important properties with predicate logic (Interpolation, Effective Axiomatizability) and that even *improve* on this, preferably by being decidable. The minimal predicate logic satisfies these demands – but what about more powerful candidates? Here Cylindric Algebra becomes important. Equational theories in the latter field correspond with modal logics in our landscape, via a well-known representation (cf. Venema 1991, Marx 1994). (Subdirectly irreducible algebras play a key role here. Cf. Baayen 1960, Blok 1977, van Benthem 1985.) Natural intermediate systems have been identified in this way (cf. Henkin-Monk-Tarski 1985, Németi 1991, 1993), by a method of 'relativization' from the algebraic literature.

One attractive candidate is **CRS**, consisting of all predicate-logical validities in the state frames satisfying all *universal frame conditions* true in standard assignment

models. These are the general logical properties of assignments, that do not make existential demands on their supply. (The latter would be more 'mathematical' or 'set-theoretic'.) CRS is known to be decidable, though non-finitely axiomatizable. Moreover, its frame definition needs only universal *Horn* clauses, from which Craig Interpolation follows (van Benthem 1994). Another way of describing CRS may have independent appeal. Consider state frames where  $S$  is a family of ordinary assignments (but not necessarily the full function space  $D^{\text{VAR}}$ ), and the  $R_x$  are the standard relations  $=_x$ . Such frames admit 'assignment gaps', which model 'dependencies' between variables: i.e., changes in value for one variable  $x$  may induce, or be correlated with changes in value for another variable  $y$  (van Lambalgen 1991, Fine 1985 give natural illustrations). This phenomenon cannot be modeled in standard Tarskian semantics, which changes values for variables completely independently. The latter is the 'degenerate case' where all interesting dependencies between variables have been suppressed. From CRS, one can move upward, by considering only families of assignments that satisfy natural closure conditions. For instance, assignment sets might be closed under local shifts in values to variables, or under reassignment of values for one variable to another. Such further structure tends to support the introduction of further operators into the language (e.g., permutation or substitution operators, as well as a predicate for identity). For the resulting logics, cf. Venema 1991, Németi 1993, Marx 1994.

## 5. EXPLORING THE RICHER SEMANTICS

The landscape of dynamic assignment logics invites obvious geographical research. What are its natural landmarks? Current research by algebraic logicians is bringing to light various interesting mathematical phenomena here. For instance, intermediate logics may have better properties than standard logic. (E.g., the strong Interpolation Theorem for CRS in van Benthem 1994 fails for predicate logic.) Next, generalized assignment semantics throws new light on old questions in standard model theory. (E.g., it improves the poor behaviour of 'finite-variable fragments' of predicate logic that are currently used in defining complexity classes semantically via query languages: cf. Andréka, Németi & van Benthem 1994.) There are also challenging issues of mathematical representation for abstract state frames (some sample results are found in Henkin-Monk-Tarski 1985, Venema 1991, van Benthem 1994). This is an area where modal logicians and algebraists have made common cause by now.

Perhaps the most striking consequence of the new perspective, however, concerns the *language* of predicate logic. A generalized semantics, with its weaker logics, often invites re-design of the original formal language. Distinctions become visible which were suppressed or overlooked in the 'standard semantics'. This general point is well-known from earlier work on, e.g., intuitionistic logic, relevant logic or linear logic. (For instance, classical 'conjunction' splits into two relevant or linear versions, and some connectives in these weaker logics have no classical counterparts at all.) Again, the algebraic tradition has been aware of this issue. Weaker cylindrical equational logics may support expanded languages with desirable items like 'discriminator terms', which allow one to pass from algebraic quasi-equations to ordinary equations (Németi 1991). Likewise, modal semantics supports an infinite hierarchy of ever more expressive formalisms (cf. de Rijke 1993). When analyzing predicate logic, two striking examples occur of such expressive enrichment. First, there is a case for adding *substitutions*. Consider the central first-order axiom of 'Existential Generalization':  $[t/x]\phi \rightarrow \exists x\phi$ . Its computational content is this: 'definite assignment implies random assignment'. To express this intuition, one

treats the substitution operator  $[t/x]$  as a new modality (metabatically, its very notation made this historically inevitable ...). The earlier state frames must then be expanded with matching update relations  $A_{x,t}$  saying that the target state has its  $x$ -value replaced by the  $t$ -value of the source state. This move brings definite assignment as such into our models. The previous modal analysis still applies. Notably, standard substitution laws show dynamic content via frame correspondence. For instance,  $[t/x](\phi \vee \psi) \leftrightarrow [t/x]\phi \vee [t/x]\psi$  is universally valid in the minimal logic, whereas  $[t/x]\neg \phi \leftrightarrow \neg [t/x]\phi$  expresses that the relation  $A_{x,t}$  must be a total *function*. (Van Benthem 1994 also considers backward 'temporal' versions of substitution.)

Secondly, generalized assignment models suggest a natural distinction between singular quantifiers and *polyadic quantifiers* (cf. Keenan & Westerståhl 1994 for extensive linguistic motivation of the latter). One can interpret a polyadic existential formula like  $\exists xy \bullet \phi$  as saying that there exists some state satisfying  $\phi$  with possibly different  $x$ - and  $y$ -values from the current one. In general, no intermediate states need exist allowing the stepwise singular decompositions  $\exists x \exists y \bullet \phi$  or  $\exists y \exists x \bullet \phi$  that would be equivalent in standard logic. In state frames, direct interpretation of polyadic quantifiers involves simultaneous updates  $R_X$  for sets or sequences  $X$  of individual variables. A similar move will be needed to model simultaneous substitutions  $[t_1/x_1, \dots, t_k/x_k]$ , which are known to be irreducible to iterations of singular substitutions. Another view of these linguistic extensions is as follows. From the earlier poly-modal logic with only atomic assignment programs, we are now passing on to a full dynamic logic with operators forming complex programs. In particular, an iterated singular quantifier  $\exists x \exists y \bullet$  involves a *sequential* composition of update relations  $R_x ; R_y$ , whereas the polyadic quantifier  $\exists xy \bullet$  involves a form of *parallel* composition. Evidently, these are just first steps on a longer road.

## 6. CONCLUSIONS

The above re-analysis of what is arguably the basic tool of modern logic may be seen as an instance of a more general philosophical enterprise. What we are trying to do is locate the 'computational core' of a phenomenon – in this case the dynamics of variable-value assignment – and detach it from its 'mathematical wrappings', i.e., more negotiable aspects of its accidental mathematical presentation. We are after the former: the rest is imported complexity. Such a philosophical program may have great practical repercussions. In particular, the hallowed 'undecidability of predicate logic' might merely reflect an infelicity of its traditional Tarskian modeling: namely, the import of extraneous set-theoretic facts about full function spaces  $D^{\text{VAR}}$  – rather than the core facts about quantification and variable assignment. Thus, adopting 'dynamic semantics' and thinking it through might lead to decreased logical complexity – once we have the courage of our convictions. This provocative statement needs to be backed up, of course, by concrete analysis of predicate-logical reasoning found in applications. Which universal validities are really used (that is, under appropriate formalizations)?

I am not quite sure that Cor Baayen will be overjoyed by this radical departure from the tenets of our Founding Fathers. But he will certainly appreciate the following points. At least, our case study demonstrates a commonality in key interests between such apparently diverse disciplines as logic, computer science and linguistics. In particular, it demonstrates that genuine 'application' is not a one-way

process, but an interaction. Standard logic has inspired an illuminating analysis of computational processes via 'dynamic logics' and their ilk. But what happens now is that, conversely, dynamic viewpoints may 'turn around' and start challenging received views of what standard logic is all about. This move does not invalidate the achievements of previous periods. On the contrary, as we have seen, it is driven by insights from cylindric algebra, an enterprise squarely within mathematical logic - and it will no doubt inspire that area too. I conclude that Cor Baayen's scientific interests, outlined at the beginning of this paper, have proved fruitful and topical: both generally, and in their technical bent.

## 7. REFERENCES

- Andréka, H., I. Németi & J. van Benthem, 1994, 'Back and Forth Between Modal Logic and Classical Logic', Mathematical Institute, Hungarian Academy of Sciences, Budapest / Institute for Logic, Language and Computation, University of Amsterdam.
- Baayen, C., 1960, 'Subdirect Oplosbare Cylinder-Algebra's', Rapport ZW 1960-006, Stichting Mathematisch Centrum, Amsterdam.
- Baayen, C., 1962, 'Cylinder-Algebra's', Rapport ZW 1962-004, Stichting Mathematisch Centrum, Amsterdam.
- Barwise, J., 1987, 'Noun Phrases, Generalized Quantifiers and Anaphora', in P. Gärdenfors, ed., *Generalized Quantifiers. Logical and Linguistic Approaches*, Reidel, Dordrecht, 1-29.
- Benthem, J. van, 1985, *Modal Logic and Classical Logic*, Bibliopolis, Napoli.
- Benthem, J. van, 1991, *Language in Action. Categories, Lambdas and Dynamic Logic*, North-Holland, Amsterdam.
- Benthem, J. van, 1994, 'Modal Foundations for Predicate Logic', CSLI Research Report, Center for the Study of Language and Information, Stanford University.
- Benthem, J. van & J. Bergstra, 1993, 'Logic of Transition Systems', Report CT-93-03, Institute for Logic, Language and Computation, University of Amsterdam. (To appear in *Journal of Logic, Language and Information*.)
- Blok, W., 1977, *Varieties of Interior Algebras*, Dissertation, Mathematical Institute, University of Amsterdam.
- Eyck, J. van & A. Visser, eds., 1994, *Dynamic Logic and Information Flow*, The MIT Press, Cambridge (Mass.).
- Fine, K., 1985, *Reasoning With Arbitrary Objects*, Blackwell, Oxford.
- Gärdenfors, P., 1988, *Knowledge in Flux. Modelling the Dynamics of Epistemic States*, The MIT Press, Cambridge (Mass.).
- Goldblatt, R., 1987, *Logics of Time and Computation*, CSLI Lecture Notes, Chicago University Press.
- Groenendijk, J. & M. Stokhof, 1991, 'Dynamic Predicate Logic', *Linguistics and Philosophy* 14, 39- 100.
- Harel, D. & D. Kozen, 1994, *Dynamic Logic*, Department of Computer Science, Technion, Haifa / Department of Computer Science, Cornell University.
- Henkin, L., D. Monk & A. Tarski, 1985, *Cylindric Algebra*, part II, North-Holland, Amsterdam.
- Jaspars, J., 1994, *Calculi for Constructive Communication*, ILLC Dissertation Series 1994-1, Institute for Logic, Language and Computation, University of Amsterdam / Institute for Language and Knowledge Technology, University of Tilburg

- Kamp, H., 1984, 'A Theory of Truth and Semantic Representation', in J. Groenendijk et al., eds., *Truth, Interpretation and Information*, Foris, Dordrecht, 1-41.
- Keenan, E. & D. Westerståhl, 1994, 'Quantifiers', a chapter in J. van Benthem & A. ter Meulen, eds., *Handbook of Logic and Language*, Elsevier Science Publishers, Amsterdam.
- Lambalgen, M. van, 1991, 'Natural Deduction for Generalized Quantifiers'. In J. van der Does & J. van Eyck, eds., *Generalized Quantifiers: Theory and Applications*, Dutch Ph. D. Network for Logic, Language and Information, Amsterdam, 143-154. To appear with Cambridge University Press.
- Marx, M., 1994, *Arrow Logic and Relativized Algebras of Relations*, Dissertation, CCSOM, Faculty of Social Sciences / Institute for Logic, Language and Computation, University of Amsterdam.
- Muskens R., J. van Benthem & A. Visser, 1994, 'Dynamics', a chapter in J. van Benthem & A. ter Meulen, eds., *Handbook of Logic and Language*, Elsevier Science Publishers, Amsterdam.
- Németi, I., 1991, 'Algebraizations of Quantifier Logics: An Introductory Overview', Mathematical Institute, Hungarian Academy of Sciences, Budapest.
- Németi, I., 1993, 'Decidability of Weakened Versions of First-Order Logic', Lecture Notes Workshop on Algebraization of Logics, Fifth European Summer School in Logic, Language and Information, Lisbon.
- Rijke, M. de, 1993, *Extending Modal Logic*, Dissertation Series 1993-4, Institute for Logic, Language and Computation, University of Amsterdam.
- Veltman, F., 1991, 'Defaults in Update Semantics', Report LP-91-02, Institute for Logic, Language and Computation, University of Amsterdam. To appear in the *Journal of Philosophical Logic*.
- Venema, Y., 1991, *Many-Dimensional Modal Logic*, Dissertation, Institute for Logic, Language and Computation, University of Amsterdam.
- Vermeulen, K., 1994, *Exploring the Dynamic Environment*, Dissertation, Onderzoeksinstituut voor Taal en Spraak, University of Utrecht.

# Introductory note to “Object-Oriented Algebraic Specification”

Jan Bergstra  
Jan Heering  
Jan Willem Klop

The following CWI report proposes a notation for OOAS (Object-Oriented Algebraic Specification). It is one of four related formalisms in the area of algebraic specification that were conceived around 1984 at CWI. The other ones were ACP (Algebra of Communicating Processes), ASF (Algebraic Specification Formalism), and BMA (Basic Module Algebra). Whereas these have generated and still generate a significant volume of research, OOAS was considered of minor importance and, apart from its use in [1], no further study of it was made by CWI researchers.

In retrospect, this is unfortunate. When Banâtre *et al.* [2] independently introduced multiset programming, which in turn led Berry and Boudol [3] to the Chemical Abstract Machine (CHAM), the underlying concepts and definitions turned out to be very close to OOAS. Since then the French researchers have made substantial progress, and the CHAM has become an important theoretical tool.

We respectfully dedicate this account of the vagaries of scientific work to Cor Baayen on the occasion of his retirement as scientific director from CWI.

## REFERENCES

1. J.C.M. Baeten, J.A. Bergstra, and J.W. Klop, An operational semantics for process algebra, in: *Mathematical Problems in Computation Theory*, Banach Center Publications, Vol. 21, PWN—Polish Scientific Publishers, Warsaw, 1988, pp. 47–81.
2. J.-P. Banâtre, A. Coutant, and D. Le Métayer, A parallel machine for multiset transformation and its programming style, *Future Generations Computer Systems*, 4 (1988), pp. 133–144.
3. G. Berry and G. Boudol, The chemical abstract machine, in: *Conference Record of the Seventeenth ACM Symposium on Principles of Programming Languages (POPL '90)*, ACM, 1990, pp. 81–94.



OBJECT-ORIENTED ALGEBRAIC SPECIFICATION: PROPOSAL FOR A NOTATION AND 12  
EXAMPLES

J.A. BERGSTRA, J. HEERING, J.W. KLOP  
*Centre for Mathematics and Computer Science, Amsterdam*

A notation is introduced for expressing the dynamic behaviour of configurations of objects. At each instant of time a configuration is just a multi-set of objects which themselves are points (values) from some algebraically specified abstract data type. Several examples should convince the reader of the attractive expressive power of our notation.

1980 MATHEMATICS SUBJECT CLASSIFICATION: 68C01, 68F20.

1982 CR CATEGORIES: F.1.1, F.3.2.

KEY WORDS & PHRASES: object-oriented specification, algebraic specification, configuration transition system, transformation rule.

NOTE: This report will be submitted for publication elsewhere.

Report CS-R8411

Centre for Mathematics and Computer Science

P.O. Box 4079, 1009 AB Amsterdam, The Netherlands



## 1. INTRODUCTION

This note has the following aim: to propose a *notation* compatible with the well-known notations for *algebraic data type specification* which captures the concept of an *object*.

The reasons for doing so are many; we list some reasons in arbitrary order:

- (a) There is an increasing interest in object-oriented approaches to software design. See Cox [4], Jamsa [6], Jonkers [7] for some discussions of object-oriented programming.
- (b) The discussion on what constitutes an object and what constitutes a value is not yet settled. See Cohen [3] and MacLennan [9] for two very interesting expositions about the nature of objects.
- (c) From the point of view of abstract data types (and their algebraic specification) it is hard to understand what an object is. The history of the subject is confusing indeed. The Simula class is meant as a class of objects. Abstract data types in the ADJ tradition are modules of structured values. In the survey by Goguen & Meseguer [5] an option to augment data types with states is discussed, thus regaining some of the dynamic aspects that were somehow lost in the "initial algebra = abstract data type" stage.
- (d) We feel that a workable distinction between objects and values can be made, taking algebraic abstract data type specifications as a point of departure.

## 2. AN ORGANISATION OF NOTIONS

Let  $\Sigma$  be a (many-)sorted algebraic signature, let  $A \in \text{Alg}(\Sigma)$  be an algebra of type (signature)  $\Sigma$ .  $A$  is called an *abstract data type*. For (algebraic) specification of abstract data types, we refer to the literature collected in Kutzler & Lichtenberger [8].

The signature  $\Sigma$  is a triple  $\$(\Sigma), \mathbb{F}(\Sigma), C(\Sigma)$  (sorts, functions and constants) of  $\Sigma$ . For  $s \in \$(\Sigma)$ ,  $A_s$  is the interpretation of sort  $s$  in  $A$ . An element of  $A_s$  will be called a *point*.  $A_s$  itself will also be called a *data space*. (See Figure 1.) A point  $p \in A_s$  may play two roles:

- (i) p may represent a value,
- (ii) p may represent an object (with a particular state).

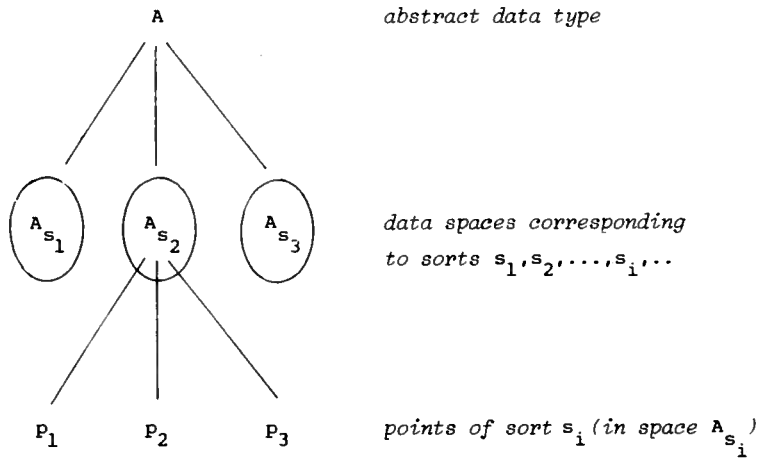


Figure 1.

A multi-set of objects (i.e. a multi-set of points seen as objects) is called a *configuration*. Configurations exhibit dynamic behaviour. In particular, configurations may perform (or allow) *transformation steps*

$$C \xrightarrow{R} C'.$$

Transformation steps are generated from *transformation rules*. In Section 3 we will present syntax and semantics of a *notation* for transformation rules.

Suppose that we know what a rule is for a given signature  $\Sigma$ . Let  $T$  be a collection of transformation rules,  $A$  a  $\Sigma$ -algebra. Then the pair  $\langle A, T \rangle$  determines a *configuration transition system*.

If  $A = T_1(\Sigma, E)$ , i.e.  $(\Sigma, E)$  is an initial algebra specification of  $A$ , and  $T$  is a collection of transformation rules for  $\Sigma$ , then

$$\langle (\Sigma, E), T \rangle$$

is an *object-oriented algebraic specification* which specifies a *configuration transition system*.

### 3. TRANSFORMATION RULES

Informally, a transformation rule is a notation of the following kind:

$$\text{rule name (parameter list)} \left( \frac{\text{configuration before transformation}}{\text{configuration after transformation}} \right)$$

Often it is convenient to divide the parameter list in three parts: one part associated with the rule name, the other two parts consisting of input values and output values respectively. This suggests the following notation:

$$\text{rule name (par. list)} \left( \begin{array}{c|c} \text{configuration before} & \text{input values} \\ \text{transformation} & \\ \hline \text{configuration after} & \text{output values} \\ \text{transformation} & \end{array} \right)$$

The input values constitute a multi-set of points which are consumed during the transformation and the output values constitute a multi-set of points which are produced during the transformation. It is understood that a configuration may be transformed inside a context (a larger configuration). So if  $C_1 \subseteq C_1 \cup C_2$  is a sub-configuration of  $C_1 \cup C_2$  (where  $\subseteq$  denotes inclusion between multi-sets and  $\cup$  their union), and

$$R = \underline{\text{name}} (\vec{p}) \left( \begin{array}{c|c} C_1 & \vec{a} \\ \hline C_1' & \vec{b} \end{array} \right)$$

is an instance of the rule with name name, then  $C_1 \cup C_2 \xrightarrow{R} C_1' \cup C_2'$  is a transformation step. (For a more elaborate explanation, see Section 9.)

Example: an instantiation  $R$  of the transformation rule

$$\underline{\text{add}} \left( \begin{array}{c|c} x & Y \\ \hline x+y & \end{array} \right)$$

used in the example below, is:  $R = \underline{\text{add}} \left( \begin{array}{c|c} 3 & 5 \\ \hline 8 & \end{array} \right)$ . (Here 3 is short for  $(1+1)+1$ , etc.) In this example  $\vec{p}$ ,  $\vec{b}$  are empty, and  $C_1 = \{3\}$ ,  $C_1' = \{8\}$ .

Now we have the transformation step

$$\{3\} \xrightarrow{R} \{8\}$$

and also e.g. for  $C_2 = (7,1)$ , the step

$$(3,7,1) \xrightarrow{R} (8,7,1).$$

Such steps can be composed into transformation sequences; e.g. if  $R'$  is the instantiation:  $\text{add} \left( \frac{7}{13} \mid 6 \right)$ , we have

$$(3,7,1) \xrightarrow{R} (8,7,1) \xrightarrow{R'} (8,13,1).$$

Here we would like to point out the relation to Plotkin [10], which addresses similar issues, where system behaviour is systematically described by means of transition relations.

The following two very simple examples will help to further explain the notation. Consider the following specification of the initial algebra A:

$$\begin{array}{l} \Sigma \quad \left\{ \begin{array}{l} \$: N \\ ER \\ \\ \text{IF}: +: N \times N \rightarrow N \\ \quad \cdot: N \times N \rightarrow N \\ \\ \text{C}: 0 \in N \\ \quad 1 \in N \\ \quad \perp \in ER \end{array} \right. \\ \\ E \quad \left\{ \begin{array}{l} x + 0 = x \\ x + (y + 1) = (x + y) + 1 \\ x \cdot 0 = 0 \\ x \cdot (y + 1) = x \cdot y + x \end{array} \right. \end{array}$$

Now  $A = T_1(\Sigma, E)$ . We will now present two different collections  $T_1$  and  $T_2$  of transformation rules for configurations over A.

$$\begin{array}{l} T_1 \quad \left\{ \begin{array}{l} \text{succ} \left( \frac{x}{x+1} \mid \mid \right) \\ \\ \text{add} \left( \frac{x}{x+y} \mid \frac{y}{y} \right) \\ \\ \text{subtract} \left( \frac{x+y}{x} \mid \frac{y}{y} \right) \\ \\ \vdots \end{array} \right. \quad \begin{array}{l} T_{1,1} \\ \\ T_{1,2} \\ \\ T_{1,3} \end{array} \end{array}$$

$$\left| \text{subtract} \left( \begin{array}{c|c|c} x & x+y+1 & \\ \hline x & & 1 \end{array} \right) \right. \quad T_{1,4}$$

If one starts with the initial configuration  $\{0\}$ , then  $T_1$  describes the behaviour of a single counter with some actions (transformations) on it; part of this behaviour is as in Figure 2.

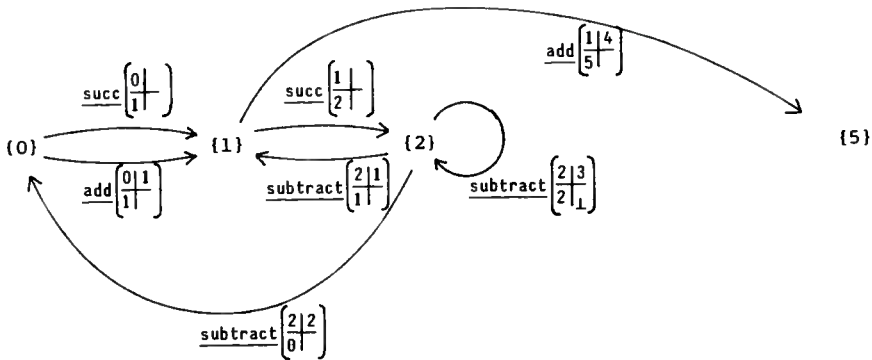


Figure 2.

Further comments on the rules of  $T_1$ :

- (i) If one of the compartments of the 'matrix' is left empty, this means that the empty multi-set  $\emptyset$  of values (or objects) is meant.
- (ii) Note the difference between rule  $T_{1,2}$  and the rule

$$\text{add} \left( \begin{array}{c|c|c} x & y & \\ \hline x+y & & \end{array} \right);$$

in  $T_{1,2}$  we focus on the transformation of one object, while in the displayed rule the *fusion* of two objects is embodied.

- (iii) The rules  $T_{1,3}$  and  $T_{1,4}$  for subtraction exhibit *polymorphism of types*: in  $T_{1,3}$  the multi-set of output values is empty, while in  $T_{1,4}$  an error message is delivered.

In the second example the same initial algebra  $A$  as above is used. The set  $T_2$  of transformation rules for configurations over  $A$  will describe the behaviour of a fixed number  $n_0$  of counters. The  $k$ -th counter ( $k \in \{0, \dots, n_0 - 1\}$ ) with content  $x$  can conveniently be represented (coded) by the natural number  $k + n_0 x$ . Below,  $k, \ell, m$  vary over  $\{0, \dots, n_0 - 1\}$ .

$T_2$	$\underline{\text{create}}(k) \left( \frac{\quad   \quad x}{k + n_0 x} \right)$	$T_{2,1}$
	$\underline{\text{add}}(k, \ell, m) \left( \frac{k + n_0 x, \ell + n_0 y \quad   \quad}{m + n_0(x + y)} \right)$	$T_{2,2}$
	$\underline{\text{mult}}(k, \ell, m) \left( \frac{k + n_0 x, \ell + n_0 y \quad   \quad}{m + n_0 xy} \right)$	$T_{2,3}$
	$\underline{\text{succ}}(k) \left( \frac{k + n_0 x \quad   \quad  }{k + n_0(x + 1)} \right)$	$T_{2,4}$
	$\underline{\text{read}}(k) \left( \frac{k + n_0 x \quad   \quad}{\quad   \quad x} \right)$	$T_{2,5}$
	$\underline{\text{compare}}(k) \left( \frac{k + n_0(x + y) \quad   \quad x}{k + n_0(x + y) \quad   \quad 0} \right)$	$T_{2,6}$
	$\underline{\text{compare}}(k) \left( \frac{k + n_0 x \quad   \quad x + y + 1}{k + n_0 x \quad   \quad 1} \right)$	$T_{2,7}$
	$\underline{\text{skip}}(k) \left( \frac{k + n_0 x \quad   \quad}{\quad   \quad} \right)$	$T_{2,8}$
	$\underline{\text{copy}}(k, \ell) \left( \frac{k + n_0 x \quad   \quad  }{k + n_0 x, \ell + n_0 x} \right)$	$T_{2,9}$

Comments: (i) The rules  $T_{2,6}$  and  $T_{2,7}$  for compare(k) compare the content a of counter k with some given number b; if  $a \geq b$  the output is 0, otherwise 1.  
(ii) Note that the copy(k,ℓ) rule can lead to confusion (in the sense that two indiscernible objects may arise) if it is applied while an object of the form  $\ell + n_0x$  is present (which can be avoided by first performing skip(ℓ) or read(ℓ)).  
(iii) The empty configuration is an adequate initial configuration for this system. Clearly  $T_{2,1-9}$  offer only limited facilities (subtraction is absent etc.). Moreover explicit naming might be a preferable alternative to the coding trick, which represents "counter k with content x" as  $k + n_0x$ , if natural number objects are to be maintained.

#### 4. THE STACK

In this section we consider object-oriented specifications of the stack. We formulate four different specifications of the dynamic behaviour of a single stack. This raises the following

Question: *is it possible to express this rich variety of operational possibilities without the object-oriented approach (i.e. in terms of the original algebraic framework)?*

We will leave this question unanswered.

$\Sigma$	$\$:$ A S ER B  IF: push: $A \times S + S$  $\Phi:$ $a_1, \dots, a_n \in A$ $\perp \in ER$ $\emptyset \in S$ $T \in B$ $F \in B$
----------	---

$E = \emptyset$

As data space we use  $T_I(\Sigma, \emptyset)$ .



$T_3$	$\text{push}$	$\frac{x}{\text{push}(a,x)} \mid a$	$T_{3,1}$
	$\text{pop}$	$\frac{\text{push}(a,x)}{x} \mid a$	$T_{3,2}$
	$\text{pop}$	$\frac{\emptyset}{\emptyset} \mid \perp$	$T_{3,3}$

The initial configuration is  $\{\emptyset\}$ . At each time the configuration will be a singleton.

$T_4$	$\text{push}$	$\frac{x}{\text{push}(a,x)} \mid a$	$T_{4,1}$
	$\text{pop}$	$\frac{\text{push}(a,x)}{x} \mid$	$T_{4,2}$
	$\text{pop}$	$\frac{\emptyset}{\emptyset} \mid \perp$	$T_{4,3}$
	$\text{top}$	$\frac{\text{push}(a,x)}{\text{push}(a,x)} \mid a$	$T_{4,4}$
	$\text{top}$	$\frac{\emptyset}{\emptyset} \mid \perp$	$T_{4,5}$

As in the previous case  $\{\emptyset\}$  should be taken as the initial configuration.

$T_5$	$\text{create}$	$\frac{\quad}{\emptyset} \mid$	$T_{5,1}$
	$\text{push}$	$\frac{x}{\text{push}(a,x)} \mid a$	$T_{5,2}$

$$\begin{array}{l}
 \left. \begin{array}{l}
 \text{pop} \left( \begin{array}{c|c}
 \text{push}(a,x) & a \\
 \hline
 x & 
 \end{array} \right) \\
 \\
 \text{pop} \left( \begin{array}{c|c}
 \emptyset & \\
 \hline
 & \perp
 \end{array} \right)
 \end{array} \right\}
 \begin{array}{l}
 T_{5,3} \\
 \\
 T_{5,4}
 \end{array}
 \end{array}$$

In the case of  $T_5$ , pop is destructive on  $\emptyset$ . Hence after  $\perp$  has been observed an empty stack must be created again. Care must be taken not to create two or more stacks at the same time, because this would lead to non-deterministic effects of pop.

In the next example  $T_6$  we replace the create facility by a test on emptiness of the stack.

$$\begin{array}{l}
 T_6 \left\{ \begin{array}{l}
 \text{push} \left( \begin{array}{c|c}
 x & a \\
 \hline
 \text{push}(a,x) & 
 \end{array} \right) \\
 \\
 \text{empty} \left( \begin{array}{c|c}
 \text{push}(a,x) & \\
 \hline
 \text{push}(a,x) & F
 \end{array} \right) \\
 \\
 \text{empty} \left( \begin{array}{c|c}
 \emptyset & \\
 \hline
 \emptyset & T
 \end{array} \right) \\
 \\
 \text{pop} \left( \begin{array}{c|c}
 \text{push}(a,x) & \\
 \hline
 a & a
 \end{array} \right) \\
 \\
 \text{pop} \left( \begin{array}{c|c}
 \emptyset & \\
 \hline
 & \perp
 \end{array} \right)
 \end{array} \right\}
 \begin{array}{l}
 T_{6,1} \\
 \\
 T_{6,2} \\
 \\
 T_{6,3} \\
 \\
 T_{6,4} \\
 \\
 T_{6,5}
 \end{array}
 \end{array}$$

In the case of  $T_6$ ,  $(\emptyset)$  is again an appropriate initial configuration. In order to prevent loss of the stack it is useful to do pop only after a test on emptiness. If the stack is not empty, pop may be safely applied; otherwise it should not be applied because in that case the object would be irreversibly destroyed.

## 5. PROCESS ALGEBRA WITHOUT COMMUNICATION

Let  $(\Sigma_{PA}, PA)$  be the following specification.

$\Sigma_{PA}$	$\$:$ PR $\text{IF: } +: PR \times PR \rightarrow PR$ $\quad \cdot: PR \times PR \rightarrow PR$ $\quad   : PR \times PR \rightarrow PR$ $\quad \underline{\underline{}}: PR \times PR \rightarrow PR$ $\$:$ $a_1, \dots, a_n \in PR$	
PA	$x + y = y + x$ A1 $(x + y) + z = x + (y + z)$ A2 $x + x = x$ A3 $(x + y) \cdot z = x \cdot z + y \cdot z$ A4 $(x \cdot y) \cdot z = x \cdot (y \cdot z)$ A5 $x    y = x \underline{\underline{}} y + y \underline{\underline{}} x$ M1 $a \underline{\underline{}} x = a \cdot x$ M2 $(a \cdot x) \underline{\underline{}} y = a \cdot (x    y)$ M3 $(x + y) \underline{\underline{}} z = x \underline{\underline{}} z + y \underline{\underline{}} z$ M4	

Here 'a' varies over  $A = \{a_1, \dots, a_n\}$ . We will write the initial algebra  $T_I(\Sigma_{PA}, PA)$  of this specification as  $A_\omega(+, \cdot, ||, \underline{\underline{}})$ . With  $A_\omega(+, \cdot)$  we denote the reduct of  $A_\omega(+, \cdot, ||, \underline{\underline{}})$  after forgetting  $||$  and  $\underline{\underline{}}$ . Let  $\Sigma_{PA}^{+, \cdot}$  be  $\Sigma_{PA}$  minus  $||, \underline{\underline{}}$  and let BPA be A1-5. It can be shown (see Bergstra & Klop [2]) that  $A_\omega(+, \cdot) = T_I(\Sigma_{PA}^{+, \cdot}, BPA)$ . The axiom system PA was introduced in [2] as the core axiomatisation of process algebra.

When we take  $A_\omega(+, \cdot)$  as a data space, and use the  $a \in A$  as rule names, the following transformation rules (without inputs and outputs) reflect the operational semantics of  $+$  (*choice, alternative composition*) and  $\cdot$  (*product, sequential composition*):

$T_{7,1-4}$

$$a \left[ \begin{array}{c|c} a & \\ \hline & \end{array} \right] \quad a \left[ \begin{array}{c|c} a+x & \\ \hline & \end{array} \right] \quad a \left[ \begin{array}{c|c} a \cdot x & \\ \hline x & \end{array} \right] \quad a \left[ \begin{array}{c|c} a \cdot x + y & \\ \hline x & \end{array} \right]$$

Now consider the configuration

$$\{x_1, \dots, x_k\}.$$

The behaviour of this configuration corresponds to that of the process

$$x_1 \parallel \dots \parallel x_k.$$

Thus the formation of configurations is represented by the operation  $\parallel$  of PA. It can be concluded that process algebra is more denotational than object-oriented system specification by means of transformation rules.

## 6. SETS OF INTEGERS

Let  $\Sigma$  be as follows:

$\Sigma$	$\$:$ N SN B ER
	$\mathbb{F}:$ eq: $N \times N \rightarrow B$ ins: $N \times SN \rightarrow SN$ del: $N \times SN \rightarrow SN$ s: $N \rightarrow N$
	$\mathbb{C}:$ T $\in B$ F $\in B$ 0 $\in N$ $\emptyset \in SN$ $\perp \in ER$

As (conditional) equational specification of the data space we take:

E	$eq(0,0) = T$ $eq(0,s(x)) = F$ $eq(s(x),0) = F$ $eq(s(x),s(y)) = eq(x,y)$ $ins(x,ins(x,X)) = ins(x,X)$ $ins(x,ins(y,X)) = ins(y,ins(x,X))$ $del(x,\emptyset) = \emptyset$
---	---

$$\begin{cases} \text{del}(x, \text{ins}(x, Y)) = \text{del}(x, Y) \\ \text{eq}(x, y) = F \rightarrow \text{del}(x, \text{ins}(y, X)) = \text{ins}(y, \text{del}(x, X)) \end{cases}$$

We will now describe a configuration transformation system starting from  $\{\emptyset\}$  as an initial configuration.

$T_8$	$\frac{\text{ins}}{\left( \begin{array}{c c} X & a \\ \hline \text{ins}(a, X) & \end{array} \right)}$	$T_{8,1}$
	$\frac{\text{del}}{\left( \begin{array}{c c} X & a \\ \hline \text{del}(a, X) & \end{array} \right)}$	$T_{8,2}$
	$\frac{\text{get}}{\left( \begin{array}{c c} \text{ins}(a, X) & \\ \hline X & a \end{array} \right)}$	$T_{8,3}$
	$\frac{\text{get}}{\left( \begin{array}{c c} \emptyset & \\ \hline \emptyset & \perp \end{array} \right)}$	$T_{8,4}$
	$\frac{\text{elt}}{\left( \begin{array}{c c} \text{ins}(a, X) & a \\ \hline \text{ins}(a, X) & T \end{array} \right)}$	$T_{8,5}$
	$\frac{\text{elt}}{\left( \begin{array}{c c} \text{del}(a, X) & a \\ \hline \text{del}(a, X) & F \end{array} \right)}$	$T_{8,6}$
	$\frac{\text{empty}}{\left( \begin{array}{c c} \emptyset & \\ \hline \emptyset & T \end{array} \right)}$	$T_{8,7}$
	$\frac{\text{empty}}{\left( \begin{array}{c c} \text{ins}(a, X) & \\ \hline \text{ins}(a, X) & F \end{array} \right)}$	$T_{8,8}$

Remark: note the implicit non-determinism present in  $T_{8,3}$ . Namely, by the instance

$$R = \frac{\text{get}}{\left( \begin{array}{c|c} \text{ins}(a, \text{ins}(b, \emptyset)) & \\ \hline \text{ins}(b, \emptyset) & a \end{array} \right)}$$

we have the step  $\{\text{ins}(a, \text{ins}(b, \emptyset))\} \xrightarrow{R} \{\text{ins}(b, \emptyset)\}$ . Further, by E we have

$\text{ins}(a, \text{ins}(b, \emptyset)) = \text{ins}(b, \text{ins}(a, \emptyset))$ , hence the configuration in the LHS of the displayed step can also be transformed to  $\{\text{ins}(a, \emptyset)\}$  by the instance of  $T_{8,3}$ :

$$R' = \underline{\text{get}} \left( \frac{\text{ins}(b, \text{ins}(a, \emptyset))}{\text{ins}(a, \emptyset)} \mid b \right).$$

## 7. A SIMPLE EDITOR

This example has been taken from Bergstra & Klop [1]. Let  $A = \{a_1, \dots, a_n\}$  be an alphabet of symbols. Consider the following signature:

$$\Sigma_F \left| \begin{array}{l} \$: F \\ \text{Edf} \\ E \\ \text{IF}: * : F \times F \rightarrow F \\ \text{edobj}: F \times F \rightarrow \text{Edf} \\ \Phi: \epsilon \in F \\ a \in F \text{ (all } a \in A) \\ \perp \in E \\ \text{OK} \in E \end{array} \right.$$

with equations

$$E_F \left| \begin{array}{l} x * \epsilon = x \\ \epsilon * x = x \\ (x * y) * z = x * (y * z) \end{array} \right.$$

We use the initial algebra  $T_I(\Sigma_F, E_F)$  as data space. With  $\text{edobj}(x, y)$  we denote a text  $x*y$  which is being edited with the cursor between  $x$  and  $y$ .

The following set of rules  $T_9$  presents an object-oriented specification of an editor. Here it is assumed that there are some means to inspect the object being edited; i.e. the fact that the user is watching the string being edited, is not explicitly modeled by these transformation rules. A possibility for modeling this would be to output  $x*_y$  whenever  $\text{edobj}(x, y)$  is formed, where  $'_'$  is some new symbol denoting the cursor (by putting  $x*_y$  in the lower-righthand corner of the appropriate rule).

$T_9$	<u>editor</u>	$\left( \begin{array}{c c} & x \\ \hline \text{edobj}(\epsilon, x) & \text{OK} \end{array} \right)$		$T_{9,1}$
	<u>quit</u>	$\left( \begin{array}{c c} \text{edobj}(x, y) & \\ \hline & x*y \end{array} \right)$		$T_{9,2}$
	<u>left</u>	$\left( \begin{array}{c c} \text{edobj}(\epsilon, y) & \\ \hline \text{edobj}(\epsilon, y) & \perp \end{array} \right)$		$T_{9,3}$
	<u>left</u>	$\left( \begin{array}{c c} \text{edobj}(x*a, y) & \\ \hline \text{edobj}(x, a*y) & \end{array} \right)$	$(a \in A)$	$T_{9,4,a}$
	<u>right</u>	$\left( \begin{array}{c c} \text{edobj}(x, \epsilon) & \\ \hline \text{edobj}(x, \epsilon) & \perp \end{array} \right)$		$T_{9,5}$
	<u>right</u>	$\left( \begin{array}{c c} \text{edobj}(x, a*y) & \\ \hline \text{edobj}(x*a, y) & \end{array} \right)$	$(a \in A)$	$T_{9,6,a}$
	<u>delete</u>	$\left( \begin{array}{c c} \text{edobj}(x, a*y) & \\ \hline \text{edobj}(x, y) & \end{array} \right)$	$(a \in A)$	$T_{9,7,a}$
	<u>delete</u>	$\left( \begin{array}{c c} \text{edobj}(x, \epsilon) & \\ \hline \text{edobj}(x, \epsilon) & \perp \end{array} \right)$		$T_{9,8}$
	<u>insert</u>	$\left( \begin{array}{c c} \text{edobj}(x, y) & a \\ \hline \text{edobj}(x*a, y) & \end{array} \right)$	$(a \in A)$	$T_{9,9,a}$

Taking care that at most one edobj is active at any time this will work. Note that  $T_{9,3-9}$  constitute the heart of the matter. These rules describe the editing activities proper.

The next step is to describe a storage and retrieval mechanism for files. Consider the following signature:



$\Sigma_{FSR}$	$\mathcal{F}$ : FD	(file directory)
	F	(texts/files)
	FN	(file names)
	P	(pairs)
	B	(booleans)
	IF: present: FN $\times$ FD $\rightarrow$ FD	(introduction of name)
	absent: FN $\times$ FD $\rightarrow$ FD	(deletion of name)
	contents: FN $\times$ F $\times$ FD $\rightarrow$ FD	(constructor of the file directories)
	pair: FN $\times$ FD $\rightarrow$ P	
	*: F $\times$ F $\rightarrow$ F	(concatenation on files)
	$\bar{*}$ : FN $\times$ FN $\rightarrow$ FN	(concatenation on names)
	eq: FN $\times$ FN $\rightarrow$ B	(equality test on names)
	$\mathcal{C}$ : T $\in$ B	(true)
	F $\in$ B	(false)
	$\emptyset \in$ FD	(empty structure)
	$a_1, \dots, a_n \in$ F	(alphabet for file)
	$b_1, \dots, b_m \in$ FN	(alphabet for names)
	$\epsilon \in$ F	
	$\bar{\epsilon} \in$ FN	
	Variables: x, y, z $\in$ F	
	u, v, w $\in$ FN	
	X $\in$ FD	

(Conditional) equations:

$E_{FSR}$	$(x * y) * z = x * (y * z)$	
	$x * \epsilon = x$	
	$\epsilon * x = x$	
	$u \bar{*} (v \bar{*} w) = (u \bar{*} v) \bar{*} w$	
	$u \bar{*} \bar{\epsilon} = u$	
	$\bar{\epsilon} \bar{*} u = u$	
	$eq(\bar{\epsilon}, \bar{\epsilon}) = T$	
	$eq(b_i \bar{*} x, b_i \bar{*} y) = eq(x, y)$	(i $\in$ {1, ..., m})

$$\begin{aligned}
& \text{eq}(b_i \bar{x}, b_j \bar{y}) = F && (i \neq j, i, j \in \{1, \dots, m\}) \\
& \text{eq}(\bar{e}, b_i \bar{x}) = F && (i \in \{1, \dots, m\}) \\
& \text{eq}(b_i \bar{x}, \bar{e}) = F && (i \in \{1, \dots, m\}) \\
& \text{contents}(u, x, \text{contents}(u, y, X)) = \text{contents}(u, x, X) \\
& \text{eq}(u, v) = F \rightarrow \text{contents}(u, x, \text{contents}(v, y, X)) = \\
& \quad \text{contents}(v, y, \text{contents}(u, x, X)) \\
& \text{present}(u, \emptyset) = \text{contents}(u, \epsilon, \emptyset) \\
& \text{present}(u, \text{contents}(u, x, X)) = \text{contents}(u, x, X) \\
& \text{eq}(u, v) = F \rightarrow \text{present}(u, \text{contents}(v, x, X)) = \\
& \quad \text{contents}(v, x, \text{present}(u, X)) \\
& \text{absent}(u, \emptyset) = \emptyset \\
& \text{absent}(u, \text{contents}(u, x, X)) = \text{absent}(u, X) \\
& \text{eq}(u, v) = F \rightarrow \text{absent}(u, \text{contents}(v, x, X)) = \\
& \quad \text{contents}(v, x, \text{absent}(u, X))
\end{aligned}$$

The initial algebra  $T_I(\Sigma_{FSR}, E_{FSR})$  is an appropriate data space for the permanent environment of the editor. Working in

$$T_I((\Sigma_{FSR}, E_{FSR}) \cup (\Sigma_F, E_F))$$

we can specify the system as follows (with  $\{\emptyset\}$  as an initial configuration):

$$\begin{array}{l}
T_{10} \left[ \begin{array}{c|c} \text{introduce} & \left( \frac{\text{absent}(u, X)}{\text{contents}(u, \epsilon, X)} \mid \frac{u}{\text{OK}} \right) \\ \hline \text{introduce} & \left( \frac{\text{present}(u, X)}{\text{present}(u, X)} \mid \frac{u}{\perp} \right) \\ \hline \text{skip} & \left( \frac{\text{present}(u, X)}{\text{absent}(u, X)} \mid \frac{u}{\text{OK}} \right) \\ \hline \text{skip} & \left( \frac{\text{absent}(u, X)}{\text{absent}(u, X)} \mid \frac{u}{\perp} \right) \\ \hline \text{edit} & \left( \frac{\text{contents}(u, x, X)}{\text{edobj}(\epsilon, x), \text{pair}(u, X)} \mid \frac{u}{\text{OK}} \right) \end{array} \right] \begin{array}{l} T_{10,1} \\ T_{10,2} \\ T_{10,3} \\ T_{10,4} \\ T_{10,5} \end{array}
\end{array}$$

$\underline{\text{edit}}$	$\left( \begin{array}{l l} \text{absent}(u,X) & u \\ \text{absent}(u,X) & \perp \end{array} \right)$	$T_{10,6}$
$\underline{\text{save}}$	$\left( \begin{array}{l l} \text{edobj}(x,y), \text{pair}(u,X) & \\ \text{contents}(u, x * y, X) & \end{array} \right)$	$T_{10,7}$
(plus:) $T_{9,3-9}$		

### 8. A MULTI-USER ENVIRONMENT FOR THE SIMPLE EDITOR

We now consider the following organisation:

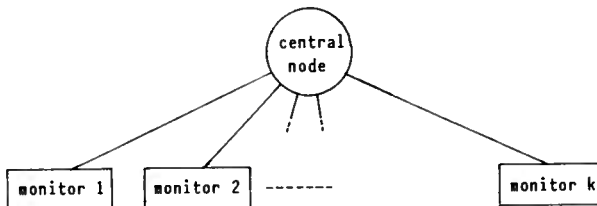


Figure 3.

At monitor  $k$  edit sessions act on an object  $\text{edobj}(k,x,y)$ . A user must log in at a terminal with a user name which should be known to the system (by having been introduced at the central node). Each user name is also the index of a file in the permanent central file directory. This file is updated after each edit session.

As before we start with a signature and a specification for the data space. Like in example 7 we proceed in two phases. The central file directory is introduced in the second phase.

#### First phase.

$\Sigma_{KME}$	$\$:$	F (files) Edf (files being edited) MN (monitor names) AMO (active monitor objects) PMO (passive monitor objects) B (booleans) UN (user names) E (signals)
----------------	-------	--

IF:  $*$ :  $F \times F \rightarrow F$   
 $\bar{*}$ :  $UN \times UN \rightarrow UN$   
edobj:  $MN \times F \times F \rightarrow Edf$   
amo:  $MN \times UN \rightarrow AMO$   
pmo:  $MN \rightarrow PMO$   
eq:  $UN \times UN \rightarrow B$

$\Phi$ :  $T \in B$   
 $F \in B$   
 $\epsilon \in F$   
 $a_1, \dots, a_n \in F$   
 $\bar{\epsilon} \in UN$   
 $b_1, \dots, b_m \in UN$   
 $l, \dots, k \in MN$   
 $\perp \in E$   
 $OK \in E$

Variables:  $x, y, z \in F$   
 $u, v, w \in UN$   
 $k \in MN$

$E_{KME}$

$(x * y) * z = x * (y * z)$   
 $x * \epsilon = x$   
 $\epsilon * x = x$   
 $u \bar{*} (v \bar{*} w) = (u \bar{*} v) \bar{*} w$   
 $u \bar{*} \bar{\epsilon} = u$   
 $\bar{\epsilon} \bar{*} u = u$   
 $eq(\bar{\epsilon}, \bar{\epsilon}) = T$   
 $eq(b_i \bar{*} x, b_i \bar{*} y) = eq(x, y) \quad (i \in \{1, \dots, m\})$   
 $eq(b_i \bar{*} x, b_j \bar{*} y) = F \quad (i \neq j, i, j \in \{1, \dots, m\})$   
 $eq(\bar{\epsilon}, b_i \bar{*} x) = F \quad (i \in \{1, \dots, m\})$   
 $eq(b_i \bar{*} x, \bar{\epsilon}) = F \quad (i \in \{1, \dots, m\})$

As before we work in  $T_I(\Sigma_{KME}, E_{KME})$ . As initial configuration we assume

$\{pmo(1), \dots, pmo(k)\}$ .

The first system description is  $T_{11}$ . The transition rules  $T_{11,4-10}$  describe

the actual working of the editor. The other rules will be replaced in the second phase.

$T_{11}$	$\underline{\text{login}}(k) \left( \frac{\text{pmo}(k)}{\text{amo}(k, u), \text{edobj}(k, \epsilon, x)} \mid \begin{array}{l} u, x \\ \text{OK} \end{array} \right)$	$T_{11,1}$
	$\underline{\text{login}}(k) \left( \frac{\text{amo}(k, u)}{\text{amo}(k, u)} \mid \perp \right)$	$T_{11,2}$
	$\underline{\text{logout}}(k) \left( \frac{\text{amo}(k, u), \text{edobj}(k, x, y)}{\text{pmo}(k)} \mid x * y \right)$	$T_{11,3}$
	$\underline{\text{logout}}(k) \left( \frac{\text{pmo}(k)}{\text{pmo}(k)} \mid \perp \right)$	$T_{11,4}$
	$\underline{\text{left}}(k) \left( \frac{\text{edobj}(k, x * a, y)}{\text{edobj}(k, x, a * y)} \mid \right)$	$T_{11,5,a}$
	$\underline{\text{left}}(k) \left( \frac{\text{edobj}(k, \epsilon, x)}{\text{edobj}(k, \epsilon, x)} \mid \perp \right)$	$T_{11,6}$
	$\underline{\text{right}}(k) \left( \frac{\text{edobj}(k, x, a * y)}{\text{edobj}(k, x * a, y)} \mid \right)$	$T_{11,7,a}$
	$\underline{\text{right}}(k) \left( \frac{\text{edobj}(k, x, \epsilon)}{\text{edobj}(k, x, \epsilon)} \mid \perp \right)$	$T_{11,8}$
	$\underline{\text{delete}}(k) \left( \frac{\text{edobj}(k, x, a * y)}{\text{edobj}(k, x, y)} \mid \right)$	$T_{11,9,a}$
	$\underline{\text{delete}}(k) \left( \frac{\text{edobj}(k, x, \epsilon)}{\text{edobj}(k, x, \epsilon)} \mid \perp \right)$	$T_{11,10}$
	$\underline{\text{insert}}(k) \left( \frac{\text{edobj}(k, x, y)}{\text{edobj}(k, x * a, y)} \mid a \right)$	$T_{11,11,a}$

Notice that the monitor objects prevent two or more users from being logged in at the same monitor simultaneously.

Second phase.

In the second phase we add a central file directory for maintaining user names and for the storage and retrieval of each user's own file.

We need a new signature:

$\Sigma_{FD}$	$\$:$ F UN FD B  $\mathbb{F}:$ known: UN $\times$ FD + FD unknown: UN $\times$ FD + FD active: UN $\times$ FD + FD silent: UN $\times$ F $\times$ FD + FD eq: UN $\times$ UN + B  $\Phi:$ T $\in$ B F $\in$ B $\emptyset \in$ FD  Variables: x, y, z $\in$ F u, v, w $\in$ UN X, Y, Z $\in$ FD
$E_{FD}$	active(u, active(u, X)) = active(u, X) active(u, active(v, X)) = active(v, active(u, X)) active(u, silent(u, x, X)) = active(u, X) eq(u, v) = F + active(u, silent(v, x, X)) = silent(v, x, active(v, X)) silent(u, x, active(u, X)) = silent(u, x, X) silent(u, x, silent(u, y, X)) = silent(u, x, X) eq(u, v) = F + silent(u, x, silent(v, y, X)) = silent(v, y, silent(u, x, X)) known(v, $\emptyset$ ) = silent(v, $\epsilon$ , $\emptyset$ ) known(u, active(u, X)) = active(u, X)

```

known(u, silent(u,x,X)) = silent(u,x,X)

eq(u,v) = F + known(u, active(v,X)) = active(v, known(u,X))

eq(u,v) = F + known(u,silent(v,x,X)) = silent(v,x,known(u,x,X))

unknown(u,∅) = ∅

unknown(v, active(u,X)) = unknown(u,X)

unknown(u, silent(u,x,X)) = unknown(u,X)

eq(u,v) = F + unknown(u, active(v,X)) = active(v, unknown(u,X))

eq(u,v) = F + unknown(u, silent(v,x,X)) = silent(v,x, unknown(u,X))

```

Now let

$$\Sigma_{KME}^{FD} = \Sigma_{KME} \cup \Sigma_{FD}$$

and

$$E_{KME}^{FD} = E_{KME} \cup E_{FD}.$$

We will work in the data space  $T_I(\Sigma_{KME}^{FD}, E_{KME}^{FD})$ .

Comment. Some remarks about  $E_{FD}$  may be in order. Let  $Z$  be the "current file directory". If  $Z = \text{active}(u,X)$ , then this expresses that a user with name  $u$  is active on some monitor. If  $Z = \text{known}(u,X)$  this expresses that user name  $u$  is known to  $Z$ . Similarly if  $Z = \text{unknown}(u,X)$  this expresses that  $u$  is not known to  $Z$ . Finally,  $Z = \text{silent}(u,x,X)$  expresses the fact that the user with name  $u$  is not active and that his (her) file is presently containing the text  $x$ .

We can now present example  $T_{12}$ : a multi-user environment for the simple editor. The system  $T_{12}$  contains  $T_{11,4-10}$  (the standard editing operations) and in addition the following transformation rules:

$$T_{12} \left| \begin{array}{l} \text{introduce} \left( \frac{\text{unknown}(u,X)}{\text{silent}(u,\epsilon,X)} \mid \frac{u}{\phantom{u}} \right) \\ \text{introduce} \left( \frac{\text{known}(u,X)}{\text{known}(u,X)} \mid \frac{\phantom{u}}{\perp} \right) \end{array} \right. \begin{array}{l} T_{12,1} \\ T_{12,2} \end{array}$$



<u>omit</u>	$\left( \begin{array}{c c} \text{known}(u, X) & u \\ \text{unknown}(u, X) & \perp \end{array} \right)$	$T_{12,3}$
<u>omit</u>	$\left( \begin{array}{c c} \text{unknown}(u, X) & \\ \text{unknown}(u, X) & \perp \end{array} \right)$	$T_{12,4}$
<u>login</u> (k)	$\left( \begin{array}{c c} \text{pmo}(k), \text{silent}(u, x, X) & u \\ \text{amo}(k, u), \text{edobj}(k, \epsilon, x), \text{active}(u, X) & \text{OK} \end{array} \right)$	$T_{12,5}$
<u>login</u> (k)	$\left( \begin{array}{c c} \text{active}(u, X) & u \\ \text{active}(u, X) & \perp \end{array} \right)$	$T_{12,6}$
<u>login</u> (k)	$\left( \begin{array}{c c} \text{unknown}(u, X) & u \\ \text{unknown}(u, X) & \perp \end{array} \right)$	$T_{12,7}$
<u>login</u> (k)	$\left( \begin{array}{c c} \text{amo}(k, v) & u \\ \text{amo}(k, v) & \perp \end{array} \right)$	$T_{12,8}$
<u>logout</u> (k)	$\left( \begin{array}{c c} \text{amo}(k, u), \text{edobj}(k, x, y), X & \\ \text{pmo}(k), \text{silent}(u, x * y, X) & \end{array} \right)$	$T_{12,9}$
<u>logout</u> (k)	$\left( \begin{array}{c c} \text{pmo}(k) & \\ \text{pmo}(k) & \perp \end{array} \right)$	$T_{12,10}$
<u>display</u> (k)	$\left( \begin{array}{c c} \text{edobj}(k, x, y) & \\ \text{edobj}(k, x, y) & x * y \end{array} \right)$	$T_{12,11}$

Remarks. (a) Notice that a user can only be omitted when not active. An active user could logout as if nothing has happened and thereafter his or her name would be known to the system again.

(b) It is entirely feasible to augment this specification with a mechanism for passwords or other protection mechanisms.

## 9. SEMANTICAL CONSIDERATIONS

In Section 3 we have given an informal explanation of the semantics of transformation rules. In this section we will elaborate that explanation, in particular, concerning the mechanism by which the *transformation rules* generate the *transformation steps*

$$C \xrightarrow[R]{} C'$$

where  $C, C'$  are configurations, i.e. multisets of objects.

Let  $A \in \text{Alg}(\Sigma)$  be a given data space; then we may write a transformation rule, written above as

$$r(\vec{v}) \left( \begin{array}{c|c} X & V \\ \hline Y & W \end{array} \right)$$

in simplified notation as follows:

$$r(\vec{v}, V, W): X \longrightarrow Y.$$

Here  $\vec{v} = v_1, \dots, v_n$  are  $\Sigma$ -terms and  $V, W, X, Y$  are finite multisets of  $\Sigma$ -terms. These terms may contain free variables and matching works as usual in term rewrite rules.  $X, Y$  themselves are not yet configurations of objects in  $A$ ; they become so after dividing out term equality in  $A$ . Further,  $V, W$  denote multisets of input and output values - properly speaking this is again true after dividing out term equality. The  $v_1, \dots, v_n$  are parameters of the rule names.

Let us introduce a constant  $\emptyset$  for the empty configuration and an operator  $\cup$  for the union of configurations. The following axioms are obviously valid:

$$\begin{aligned} X \cup Y &= Y \cup X \\ X \cup \emptyset &= X \\ (X \cup Y) \cup Z &= X \cup (Y \cup Z). \end{aligned}$$

Note that  $\cup$  is represented in process algebra [2] by  $\parallel$ , the merge operator. This connection is not quite smooth: there seems to be a difference in level of abstraction between process algebra and behavioural specification via transformation rules.

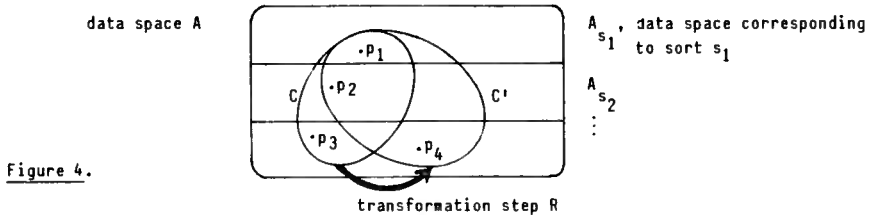
The propagation of transformations through larger configurations is as follows:

$$\frac{r(\vec{v},V,W): X \longrightarrow Y}{r(\vec{v},V,W): X \cup Z \longrightarrow Y \cup Z} .$$

Writing  $\llbracket t \rrbracket$  for the interpretation of the  $\Sigma$ -term  $t$  in the data space  $A$ , and  $\llbracket X \rrbracket = \{\llbracket t \rrbracket \mid t \in X\}$  for the multiset of objects in  $A$  denoted by the multiset of  $\Sigma$ -terms  $X$ , we can now state more precisely what a transformation step is:

if  $R = r(\vec{v},V,W): X \cup Z \longrightarrow Y \cup Z$  is obtained from the instance  $r(\vec{v},V,W): X \longrightarrow Y$  of some transformation rule, then  $R$  allows the *transformation step* of configuration  $C = \llbracket X \cup Z \rrbracket$  to  $C' = \llbracket Y \cup Z \rrbracket$ ; notation:  $C \xrightarrow{R} C'$ . (See Figure 4.)

Such transformation steps can be activated sequentially. In fact, the situation is similar to the case of *term rewriting modulo some given congruence* (apart from the multiset feature).



In other words, the transformation step  $C \xrightarrow{R} C'$  where  $C = \{p_1, p_2, \dots\}$  is obtained by choosing a *particular representation* of  $C$ , e.g.  $\{t_1, t_2, \dots\}$  such that  $\llbracket t_i \rrbracket = p_i$ , and applying some transformation rule on it as explained, to transform this representation into another (of  $C'$ ).

In an intuitive sense, such a representation of a configuration  $C$  can be considered as an *aspect* of  $C$ . E.g. in the last example  $(T_{12})$ ,  $\text{known}(v, \emptyset)$  is the file directory  $X = \emptyset$  revealing as an aspect that it knows user name  $v$  (usually such a fact would have type boolean, here it is of type file directory). And in  $\text{silent}(v, \epsilon, \emptyset)$  the same  $X = \emptyset$  reveals another aspect. The transformation rules, then, operate on such aspects.

## 10. CONCLUDING REMARKS

We feel that the object-oriented notation explained above captures at least a useful fragment of "object-oriented thinking". Clearly we have to pay a

price in terms of manageability of the transformation rules. One can, in view of Section 9, add  $\phi$  and  $\cup$ , and view the transformation rules as ordinary rewrite rules. From the point of view of algebraic specifications, adding  $\phi$ ,  $\cup$  and, in general, a type of configurations, leads to the problem that configurations have no fixed type. Any object can be an element of a configuration. In fact,  $\phi$  and  $\cup$  are polymorphic operations and this explains their flexibility which is vital for modular and incremental systems design.

#### REFERENCES

- [1] BERGSTRA, J.A. & J.W. KLOP, *Algebraisch programmeren*, (in Dutch), contained in the lecture notes for the PAO course on software engineering, Centrum voor Wiskunde en Informatica, Amsterdam 1984.
- [2] BERGSTRA, J.A. & J.W. KLOP, *Process algebra for communication and mutual exclusion*, Report IW218/83, Mathematisch Centrum, Amsterdam 1983.
- [3] COHEN, A.T., *Data abstraction, data encapsulation and object-oriented programming*, Sigplan Notices, Vol.19, No.1 (1984).
- [4] COX, B.J., *The object-oriented precompiler*, Sigplan Notices, Vol.18, No.1 (1983).
- [5] GOGUEN, J.A. & J. MESEGUER, *An initiality primer*, to appear in: Application of Algebra to Language Definition and Compilation (eds.: M. Nivat and J. Reynolds), North-Holland 1983.
- [6] JAMSA, K.A., *Object-oriented design versus structured design, a students perspective*, Software Engineering notes, Vol.9, No.1 (1984)
- [7] JONKERS, H.B.M., *On the design of an object-oriented design language*, paper presented at the Colloquium 'Van Specificatie tot Implementatie', Centrum voor Wiskunde en Informatica, Amsterdam 1983.
- [8] KUTZLER, B. & F. LICHTENBERGER, *Bibliography on abstract data types*, Springer Informatik-Fachberichte, No.68, 1983.
- [9] MACLENNAN, B.J., *Values and objects in programming languages*, Sigplan Notices, Vol.17, No.2 (1982).
- [10] PLOTKIN, G.D., *A structural approach to operational semantics*, Report Daimi FN-19, Computer Science Dept., Aarhus University, Denmark 1981.



# Polling systems

O.J. Boxma

*CWI*

*P.O. Box 94079, 1090 GB Amsterdam, The Netherlands*

*Faculty of Economics, Tilburg University*

*P.O. Box 90153, 5000 LE Tilburg, The Netherlands*

A polling system is a queueing system in which several queues are attended by a single server. Spurred by various important applications, the field of polling systems is going through a period of feverish activity. The first part of this paper surveys some of the main developments. The second part generalizes the theory of polling systems to the case in which the customer arrival process depends on the position of the server, and to the case in which customers travel from queue to queue.

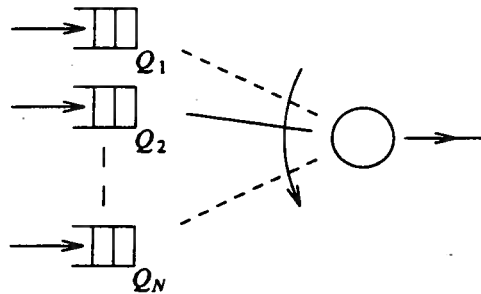
## 1 INTRODUCTION

It has been a great pleasure to write this paper on the mathematical analysis of the single-server polling system in honour of a truly devoted server. In a sometimes almost literally painstaking way, Cor Baayen saw to it as director of SMC that both LAW and CWI, and also both its mathematics and computer science groups, were served in an equally fair manner. He has strongly stimulated research at the interface of mathematics and computer science. His far-reaching vision has been crucial in realizing the INSP support for CWI in the eighties, which in its turn made it possible to build up a research group on the mathematical analysis of the performance of computer systems.

Consider the following situation. A director of a research institute divides his attention among several activities: scientific, financial, personnel matters, representative activities. Suppose that he devotes his energy for a while (a 'session') to tasks of a scientific nature, then switches to finance, etc. During a session other new tasks of the same type, as well as of different type, may be generated; furthermore, a task may have to be reconsidered in future sessions ('feedback'). The director is interested in the evolution of his workload, the

numbers of tasks of all types, etc. These quantities clearly depend on the way in which the offered load fluctuates over time; but the director can also influence the process by a judicious choice of the order of his activities and of the time he reserves for a session. The framework in which these matters can be studied is that of single-server queueing models. More precisely, it is the framework of *polling models*.

A polling model is a queueing model in which customers (tasks) arrive at a set of queues  $Q_1, \dots, Q_N$  according to some stochastic arrival process, requiring some stochastic amount of service. A single server  $B$  visits the queues in a fixed order to provide service. We assume throughout the paper that it is the cyclic order  $Q_1, \dots, Q_N, Q_1, \dots$  (cf. Fig. 1).



**FIGURE 1.** Queueing model of a polling system

When  $B$  visits  $Q_i$  and it is not empty, then  $B$  serves customers in a session at  $Q_i$  according to some service discipline. The most common service disciplines are:

- *1-limited*: serve just one customer (if at least one is present)
- *exhaustive*: serve customers until the queue is empty
- *gated*: serve precisely those customers that were already present at the start of the session

When  $Q_i$  is empty, or the session is completed, then  $B$  switches to  $Q_{i+1}$ . This may require some switchover time, which is represented by a stochastic variable.

The assumptions about the stochastic nature of the arrival process, service times and switchover times are introduced to represent the usually inherently random nature of customer behaviour, as well as a lack of detailed information. Moreover, a probability distribution for, say, service times may also represent an aggregate of in itself known, constant but distinct, service times of several types of customers. The purpose of the analysis of a polling model is to determine the performance of (several variants of) the underlying system, and

eventually to optimize system behaviour. Due to the stochasticity assumptions one can at most make probabilistic statements about the main performance measures of a polling model, like workload of the server, numbers of customers at the various queues, or their waiting times.

The analysis and optimization of polling systems has in recent years received an enormous amount of attention, and much progress has been made. It has also been one of the key research topics of the performance analysis group at CWI; cf. the PhD Theses of W.P. Groenendijk [10] and S.C. Borst [1]. Therefore it seems appropriate to briefly review the main developments, with some emphasis on contributions from the latter group. This review is presented in Section 2. In Section 3 we discuss a generalization of the standard polling model, in two directions that so far have received hardly any attention:

(i) The arrival rate of customers at the various queues may depend on the position of the server: information on which queue the server is presently visiting, and hence on which queue it will visit next, may influence the generation of new tasks.

(ii) Instead of leaving the system, customers may be routed to another (or the same) queue after having received a service. A customer's required service time at a queue may depend both on that queue and on the number of services it has already received.

We show how, for an important class of service disciplines, these generalizations can be analyzed in full detail. Crucial in this analysis is the application of the theory of multitype branching processes.

The above-mentioned features of feedback and customer information arise quite naturally in our director example; in the remainder of this section we mention several other applications of polling models.

#### *Applications of polling models*

Polling models arise in situations in which there are multiple customer classes sharing a common resource which is available to only one customer class at a time. The oldest polling model in the queueing literature concerns a patrolling repairman, who consecutively inspects a number of machines to check whether a breakdown has occurred and to restore such breakdowns [12]. In this example the server is the repairman, the queues are the machines, and the customers represent the breakdowns.

The application that gave polling models their name is a time-sharing computer system consisting of a number of terminals connected by multidrop lines to a central computer. The data transfer from the terminals to the computer (and back) is controlled via a 'polling scheme' in which the computer 'polls' the terminals, requesting their data, one terminal at a time. In this example the server represents the central computer, the queues are the terminals and the customers are the data.

The interest in polling models was strongly revived by the study of message transmission protocols in local area networks. Many communication systems provide a broadcast channel which is shared by all connected stations. When



two or more stations wish to transmit simultaneously, a conflict arises. The rules for either resolving or preventing such conflicts are referred to as ‘multi-access protocols’. An important conflict-free protocol is the *token ring* protocol. In a token ring local area network, several stations (terminals, file servers, hosts, gateways, etc.) are connected to a common transmission medium in a ring topology. A special bit sequence called the *token* is passed from one station to the next; a station that ‘possesses the token’ is allowed to transmit a message. After completion of its transmission the station releases the token, giving the next station in turn an opportunity to transmit. This situation can be represented by a polling model with 1-limited service at each queue; the server is the token, the queues are the stations and the customers are the messages. Variants of the above-described token-passing mechanism give rise to related polling models, with e.g. exhaustive service at the queues. A queueing analysis of these polling models yields insight into the (dis)advantages of the various access protocols, and allows system designers to make performance predictions. We refer the reader to Takagi [18] and Grillo [9] for surveys on polling applications in respectively computer- and communication networks. Other application areas of polling models include:

- robotics in manufacturing (a single machine processes several types of parts, incurring switchover times for changing tools)
- traffic signal control (the green light represents the availability of the server for a queue of vehicles)
- the operation of elevators (*multiple* servers are interesting here: is it better to have a concentration of elevators in a central area, or should they be dispersed over the building?)
- packet transfer protocols in B-ISDN (in such Broadband Integrated Services Digital Networks, channel access will be alternately granted to voice, video and data messages, all digitized into 53-byte packets)

The characteristic feature of all these applications is that the server is ‘moving’ between queues, implying that the priorities of the queues are dynamically (e.g., cyclically) changing. This sharply contrasts with classic *static* priority queueing models, where one type of customers always has priority over other customer types.

## 2 ANALYSIS OF POLLING SYSTEMS

In this section we briefly review the exact analysis of the standard cyclic polling system. After a detailed model description we consecutively consider workloads, waiting times and queue lengths.

### *Model description*

We here describe the standard cyclic polling model; in Section 3 we extend this

model in several ways. Customers arrive at  $N$  queues  $Q_1, \dots, Q_N$  with infinite waiting rooms according to  $N$  independent Poisson processes, with rates  $\lambda_1, \dots, \lambda_N$ . Customers who arrive at  $Q_i$  are called type- $i$  customers. Server  $B$  visits the queues in the cyclic order  $Q_1, \dots, Q_N, Q_1, \dots$ . Upon his visit to a queue, he serves one or more customers (if present) according to some service discipline like 1-limited, gated or exhaustive service (cf. Section 1). The service times of type- $i$  customers are independent, identically distributed stochastic variables; their distribution is  $B_i(\cdot)$ , with first moment  $\beta_i$ , second moment  $\beta_i^{(2)}$  and Laplace-Stieltjes Transform (LST)  $\beta_i(\cdot)$ . The switchover times of  $B$  between  $Q_i$  and  $Q_{i+1}$  are independent, identically distributed stochastic variables, with first moment  $s_i$ , second moment  $s_i^{(2)}$  and LST  $\sigma_i(\cdot)$ . The total switchover time of  $B$  in one cycle has first and second moment  $s$  respectively  $s^{(2)}$ . We assume that the interarrival, service and switchover processes are mutually independent.

The offered traffic  $\rho_i$  at  $Q_i$  is defined as  $\rho_i := \lambda_i \beta_i$ , and the total offered traffic load is  $\rho := \sum_{i=1}^N \rho_i$ . Obviously  $\rho < 1$  is a necessary condition for steady-state distributions of workloads, waiting times and queue lengths etc. to exist. When all switchover times are zero, this condition is also sufficient; otherwise the situation may be much more complicated, and in particular the service disciplines may influence the stability condition (e.g., in 1-limited service  $B$  is forced to spend time switching after each service). See Fricker and Jaïbi [8] for an extensive discussion of these stability issues. We assume in the sequel that steady-state distributions of all quantities under consideration exist.

### *The workload process*

Consider first the case that all switchover times are zero. Then  $B$  is always working as long as there is at least one customer anywhere in the system. The amount of work in the system evolves in a way that does not depend on the order of service *of* the queues and *within* the queues, or on the service disciplines at the queues; this is the principle of *work conservation* (cf. Heyman and Sobel [13], p. 418). Hence, for any service discipline at the queues of the cyclic polling system, the amount of work is distributed as the amount of work in the ‘corresponding single server queue’ with FCFS (First Come First Served) order of service. Since the superposition of  $N$  independent Poisson processes is again a Poisson process, that ‘corresponding single server queue’ is an M/G/1 queue with arrival rate  $\Lambda := \sum_{i=1}^N \lambda_i$  and with service time distribution

$$B(\cdot) := \sum_{i=1}^N (\lambda_i / \Lambda) B_i(\cdot).$$

Now consider the case that not all switchover times are zero. The principle of work conservation is clearly violated. However, it has been shown in [4] that a principle of *work decomposition* holds: the steady-state amount of work  $\mathbf{V}_{with}$  in the polling system *with* switchover times is related to the steady-

state amount of work  $\mathbf{V}_{without}$  in the ‘corresponding polling system’ *without* switchover times (hence in the above-mentioned ‘corresponding M/G/1 queue’) via

$$\mathbf{V}_{with} \stackrel{d}{=} \mathbf{V}_{without} + \mathbf{Y}, \quad (1)$$

where  $\mathbf{Y}$  is the steady-state amount of work present in the system at an epoch in which  $B$  is not serving;  $\stackrel{d}{=}$  denotes equality in distribution. Moreover,  $\mathbf{V}_{without}$  and  $\mathbf{Y}$  are independent. The distribution of  $\mathbf{V}_{without}$  is known from M/G/1 theory. The distribution of  $\mathbf{Y}$  can be determined in a number of cases, but with considerable effort. The mean  $E\mathbf{Y}$ , on the other hand, is very easily determined for virtually any set of service disciplines at the various queues - which turns out to be most useful for deriving mean waiting times, as we’ll see in formula (4) below.

REMARK 2.1

The proof of (1) as presented in [4] is based on three concepts which are sketchily indicated below.

(i) As long as  $B$  is serving, the amount of work evolves in exactly the same way as if  $B$  would be serving according to the LCFS (Last Come First Served) rule.

(ii) Characteristically for LCFS, an amount of work  $\mathbf{Y}$  found by a customer  $C$  upon his arrival in a switchover period is not served until  $C$  has been served, plus all customers who arrive during  $C$ ’s service ( $C$ ’s offspring), plus all customers who arrive during those services, etc. (together - including himself - forming  $C$ ’s ‘ancestral line’).

(iii) The time period required to serve the ancestral line of  $C$  is distributed as the busy period in the above-mentioned ‘corresponding M/G/1 queue’.

Since the principle of work conservation implies that *during* such a busy period the amount of work evolves in the same way, regardless whether service is FCFS or LCFS, combination of (i), (ii) and (iii) shows that the workload  $\mathbf{V}_{with}$  is distributed as the superposition of  $\mathbf{Y}$  and  $\mathbf{V}_{without}$ .

Another proof of (1), communicated to the author by B.T. Doshi, proceeds as follows. Assume for simplicity that the densities of the distributions of  $\mathbf{V}_{with}$  and  $\mathbf{Y}$  exist; denote them by  $v(\cdot)$  and  $y(\cdot)$ , and denote their Laplace transforms by  $\phi(\cdot)$  and  $\eta(\cdot)$ . Equating the downcrossing and upcrossing rates of level  $x > 0$  gives:

$$v(x) - (1 - \rho)y(x) = \Lambda \int_{0-}^x (1 - B(x - z))v(z)dz.$$

Combining this relation with  $v(0) = (1 - \rho)y(0)$  and taking Laplace transforms leads (with  $\beta(\cdot)$  the LST of  $B(\cdot)$ ) to:

$$\phi(\omega) - (1 - \rho)\eta(\omega) = \Lambda \frac{1 - \beta(\omega)}{\omega} \phi(\omega).$$

Hence

$$\phi(\omega) = \frac{(1 - \rho)\omega}{\omega - \Lambda + \Lambda\beta(\omega)}\eta(\omega), \quad (2)$$

which proves the decomposition into two independent components:  $\phi(\omega)$  is the product of the transform of the distribution of  $\mathbf{V}_{without}$  (a well-known M/G/1 expression) and the transform  $\eta(\omega)$  of the distribution of  $\mathbf{Y}$ . See [3] for a generalization of this principle of work decomposition, and for applications to various polling models with a *non-cyclic* visit pattern.

### Waiting times

We restrict ourself here to *mean* waiting times. Denote the mean waiting time of type- $i$  customers by  $\mathbf{E}\mathbf{W}_i$ , and the mean number of waiting type- $i$  customers by  $\mathbf{E}\mathbf{X}_i$ . These quantities are related via Little's formula:  $\mathbf{E}\mathbf{X}_i = \lambda_i\mathbf{E}\mathbf{W}_i$ . It is easy to relate the mean workload in queueing models with Poisson arrivals to mean queue lengths, and hence to mean waiting times. Indeed, under mild restrictions that are fulfilled in the standard polling model described earlier in this section, we can write (cf. [3]):

$$\mathbf{E}\mathbf{V}_{with} = \sum_{i=1}^N \beta_i \mathbf{E}\mathbf{X}_i + \sum_{i=1}^N \rho_i \frac{\beta_i^{(2)}}{2\beta_i}. \quad (3)$$

Now take means in (1) and combine the resulting formula with (3). Application of Little's formula and  $\mathbf{E}\mathbf{V}_{without} = \frac{\sum_{i=1}^N \lambda_i \beta_i^{(2)}}{2(1-\rho)}$  then yields the *pseudo-conservation law* [4]:

$$\sum_{i=1}^N \rho_i \mathbf{E}\mathbf{W}_i = \rho \frac{\sum_{i=1}^N \lambda_i \beta_i^{(2)}}{2(1-\rho)} + \mathbf{E}\mathbf{Y}. \quad (4)$$

Here (cf. the notation introduced in the model description)

$$\mathbf{E}\mathbf{Y} = \rho \frac{s^{(2)}}{2s} + \frac{s}{2(1-\rho)} \left[ \rho^2 - \sum_{i=1}^N \rho_i^2 \right] + \sum_{i=1}^N \mathbf{E}\mathbf{Z}_{ii}, \quad (5)$$

with  $\mathbf{Z}_{ii}$  the amount of work left behind at  $Q_i$  by the departing server.  $\mathbf{E}\mathbf{Z}_{ii}$ , and hence  $\mathbf{E}\mathbf{Y}$ , can be explicitly determined for polling models with standard service disciplines like 1-limited, gated, or exhaustive.  $\mathbf{E}\mathbf{Y} = 0$  for the case of zero switchover times, and then (4) reduces to the well-known *conservation law* [11]. The origin of the term conservation law is that the weighted sum  $\sum_{i=1}^N \rho_i \mathbf{E}\mathbf{W}_i$  of the mean waiting times remains the same, regardless of any changes in the service disciplines at the various queues. In the case of switchover times this weighted sum *does* change when a service discipline is changed, but

only via a - usually simple - change in EY.

The remarkably simple exact expression for  $\sum_{i=1}^N \rho_i E\mathbf{W}_i$  has in the past few years turned out to be extremely useful for a variety of purposes: testing simulation results, the development of approximations for mean waiting times, and the optimization of server routing and server visit times.

### *Queue lengths*

For the above-described  $N$ -queue cyclic polling model, with exhaustive service at all queues, Eisenberg [7] obtains the joint queue length PGF (Probability Generating Function) at epochs in which  $B$  reaches one of the queues. His solution method may also be used to handle the case of gated service at all queues. Furthermore, he also allows a fixed non-cyclic visit pattern. In a series of publications following Eisenberg's paper, an exact queue length analysis has been performed for several other  $N$ -queue polling models, with exhaustive or gated service, or mixtures and variants of these service disciplines; for an overview we refer to the survey of Takagi [19]. In contrast, polling models with limits on the number of customers to be served during a session, or on the session time, have mostly defied an exact analysis. The joint queue length distribution for the 2-queue model with 1-limited service at both queues can be obtained by transforming the problem into a Riemann- or Riemann-Hilbert boundary value problem (see, e.g., [6]), but for  $N > 2$  it is not clear at all how the queue length problem can be attacked.

In an important paper, written at CWI, Resing [15] clarifies this sharp separation between 'easy' and 'hard' polling models. He considers a class of service disciplines with the following property:

### *Branching property*

If there are  $k_i$  customers present at  $Q_i$  at the start of a visit, then during the course of the visit each of these  $k_i$  customers will effectively be replaced in an i.i.d. manner by a random population having some PGF  $h_i(z_1, \dots, z_N)$  which may be any  $N$ -dimensional PGF.

Resing demonstrates that, if the branching property holds at all queues, then the joint queue length process at successive moments that  $B$  reaches a fixed queue, say  $Q_1$ , is a *Multi-Type Branching Process* (MTBP) 'with immigration'. The theory of MTBP now yields stability conditions as well as an exact expression for the joint queue length PGF.

The 1-limited service discipline does not have the branching property. The gated and exhaustive disciplines, on the other hand, do possess this property, with respectively

$$h_i(z_1, \dots, z_N) = \beta_i \left( \sum_{j=1}^N \lambda_j (1 - z_j) \right), \quad (6)$$

(note that this is the PGF of the joint distribution of the numbers of arrivals at the various queues during one service at  $Q_i$ ), and

$$h_i(z_1, \dots, z_N) = \theta_i\left(\sum_{j \neq i} \lambda_j(1 - z_j)\right), \quad (7)$$

where  $\theta_i(\cdot)$  denotes the LST of a busy period in an M/G/1 queue with arrival rate  $\lambda_i$  and service time distribution  $B_i(\cdot)$ .

In the next section we shall extend the queue length results for the polling model of the present section, with the branching property at all queues, to some more general polling models. Therefore we now go into more detail concerning the theory of MTBP with immigration and the results of Resing [15]. Consider a system with  $N$  particle types. Let  $p^{(i)}(j_1, \dots, j_N)$  denote the probability that a type- $i$  particle ‘produces’ as offspring  $j_k$  particles of type  $k$ ,  $k = 1, \dots, N$ . The offspring PGF of  $p^{(i)}(j_1, \dots, j_N)$  is denoted by  $f^{(i)}(z_1, \dots, z_N)$ , and the mean number of particles of type  $j$  produced by one type- $i$  particle is denoted by  $m_{ij}$ . The matrix  $M = (m_{ij})$  plays an essential role in the theory of MTBP.  $M$  is called primitive if there is an  $n$  such that all entries of the matrix  $M^n$  are strictly positive. The well-known Perron-Frobenius theorem implies that a nonnegative primitive matrix  $M$  has a positive real eigenvalue  $\nu_{max}$  such that  $|\nu| < \nu_{max}$  for all other eigenvalues  $\nu$  of  $M$ .

Not only are particles produced by other particles; new particles can also enter the system via immigration (this corresponds to the arrival of customers during a period in which  $B$  is not serving). Let  $q(j_1, \dots, j_N)$  denote the probability that a group of immigrants consists of  $j_k$  particles of type  $k$ ,  $k = 1, \dots, N$ . Denote its PGF by  $g(z_1, \dots, z_N)$ , and inductively define the functions  $f_n(z_1, \dots, z_N)$  by

$$f_0(z_1, \dots, z_N) := (z_1, \dots, z_N),$$

$$f_n(z_1, \dots, z_N) := (f^{(1)}(f_{n-1}(z_1, \dots, z_N)), \dots, f^{(N)}(f_{n-1}(z_1, \dots, z_N))).$$

Resing cites the following theorem, due to Quine [14]:

**THEOREM 2.1**

Let  $\mathbf{Z}_n = (\mathbf{Z}_n^{(1)}, \dots, \mathbf{Z}_n^{(N)})$  be an MTBP with immigration in each state, with offspring PGF  $f^{(i)}(z_1, \dots, z_N)$ ,  $i = 1, \dots, N$ , and immigration PGF  $g(z_1, \dots, z_N)$ . Let the mean matrix  $M$  corresponding to the branching process be primitive and its maximal eigenvalue  $\nu_{max} < 1$ . Assume the Markov chain  $\mathbf{Z}_n$  is irreducible and aperiodic. The stationary distribution  $\pi(j_1, \dots, j_N)$  of the process  $\mathbf{Z}_n$  exists iff

$$\sum_{j_1 + \dots + j_N > 0} q(j_1, \dots, j_N) \log(j_1 + \dots + j_N) < \infty. \quad (8)$$

When this condition is satisfied, the PGF  $P(z_1, \dots, z_N)$  of the distribution

$\pi(j_1, \dots, j_N)$  is given by

$$P(z_1, \dots, z_N) = \prod_{n=0}^{\infty} g(f_n(z_1, \dots, z_N)). \quad (9)$$

Resing proves the following theorem ([15], Theorem 3):

**THEOREM 2.2**

Assume that the service discipline at each queue  $Q_i$  of the cyclic polling model satisfies the branching property with PGF  $h_i(z_1, \dots, z_N)$ ,  $i = 1, \dots, N$ . Then the numbers of customers in the queues at successive time points that  $B$  reaches  $Q_1$  constitute an MTBP with immigration in each state, where the offspring PGF's  $f^{(i)}(z_1, \dots, z_N)$  are given by

$$f^{(i)}(z_1, \dots, z_N) = h_i(z_1, \dots, z_i, f^{(i+1)}(z_1, \dots, z_N), \dots, f^{(N)}(z_1, \dots, z_N)), \quad (10)$$

and the immigration PGF  $g(z_1, \dots, z_N)$  is given by

$$g(z_1, \dots, z_N) = \prod_{i=1}^N \sigma_i \left( \sum_{k=1}^i \lambda_k (1 - z_k) + \sum_{k=i+1}^N \lambda_k (1 - f^{(k)}(z_1, \dots, z_N)) \right). \quad (11)$$

**REMARK 2.2**

The proof of Theorem 2.2 is established by considering the evolution of the joint queue length process between two successive time points, say  $t_n$  and  $t_{n+1}$ , that  $B$  reaches  $Q_1$ . Let  $c_A$  be a customer in the system at  $t_n$ . Define the *ancestral line* of  $c_A$  as  $c_A$  plus the set of all  $g_1$  customers who have arrived during the service of  $c_A$ , plus the set of all  $g_2$  customers who have arrived during the service of those  $g_1$  customers, plus ... . Define the *effective replacements* of  $c_A$  as those customers from the ancestral line of  $c_A$  who are still present at  $t_{n+1}$ . If  $c_A$  is not served in this cycle, the effective replacements of  $c_A$  consist of only  $c_A$  itself.

In a similar way the effective replacements of a customer  $c_B$  who arrives during a switchover interval between  $t_n$  and  $t_{n+1}$  are defined.

The total collection of customers in the various queues at  $t_{n+1}$  consists of the effective replacements of all customers present at  $t_n$  plus the effective replacements of all customers who have arrived during a switchover interval between  $t_n$  and  $t_{n+1}$ . The fact that all arrival processes are Poisson processes, combined with the fact that all service disciplines satisfy the *branching property*, implies that the joint queue length process at successive epochs when  $B$  reaches  $Q_1$  constitutes an MTBP with immigration. The offspring, in the sense of the MTBP, of one type- $j$  customer is the set of effective replacements of that customer, and the immigration corresponds to the effective replacements of all arrivals during the switchover periods in one cycle. In particular,  $f^{(N)}(z_1, \dots, z_N) = h_N(z_1, \dots, z_N)$ , but  $f^{(N-1)}(z_1, \dots, z_N) = h_{N-1}(z_1, \dots, z_{N-1}, f^{(N)}(z_1, \dots, z_N))$ . The latter formula reflects the fact that



type- $N$  arrivals during a type- $(N - 1)$  service may still generate their own offspring during the cycle. To arrive at the nested structure of the last PGF, the following property is used: The PGF of  $\mathbf{A}_1 + \dots + \mathbf{A}_K$ , with  $\mathbf{A}_1, \mathbf{A}_2, \dots$  and  $\mathbf{K}$  independent nonnegative integer-valued stochastic variables with PGF  $A(\cdot)$  respectively  $K(\cdot)$ , is given by

$$\begin{aligned} & \sum_{n=0}^{\infty} \sum_{j=0}^{\infty} \Pr(\mathbf{K} = j) \Pr(\mathbf{A}_1 + \dots + \mathbf{A}_j = n) z^n \\ &= \sum_{j=0}^{\infty} \Pr(\mathbf{K} = j) A(z)^j = K(A(z)). \end{aligned}$$

**REMARK 2.3**

It follows from the above two theorems that the PGF of the joint queue length process at moments that  $B$  reaches  $Q_1$  is given by the infinite-product expression (9). Let us explain and illustrate this result by considering the 2-queue case. Denote by  $P_i(z_1, z_2)$  ( $G_i(z_1, z_2)$ ) the PGF of the joint queue length distribution when  $B$  reaches (leaves)  $Q_i$ ; so  $P_1(z_1, z_2)$  is the PGF we are looking for. Then we have the following four relations.

$$\begin{aligned} P_1(z_1, z_2) &= \sigma_2(\lambda_1(1 - z_1) + \lambda_2(1 - z_2))G_2(z_1, z_2), & (12) \\ G_2(z_1, z_2) &= P_2(z_1, h_2(z_1, z_2)), \\ P_2(z_1, z_2) &= \sigma_1(\lambda_1(1 - z_1) + \lambda_2(1 - z_2))G_1(z_1, z_2), \\ G_1(z_1, z_2) &= P_1(h_1(z_1, z_2), z_2). \end{aligned}$$

Here we have used the memoryless property of the Poisson arrival processes, and the nested structure outlined above for the sum of a random number of stochastic variables, as well as the following property of PGF's:

The PGF of the sum of two independent stochastic variables is the product of their PGF's.

Combination of the four relations in (12) yields:

$$P_1(z_1, z_2) = \sigma_1(z_1, h_2(z_1, z_2))\sigma_2(z_1, z_2)P_1(h_1(z_1, h_2(z_1, z_2)), h_2(z_1, z_2)).$$

Remembering the definitions of the immigration PGF  $g(z_1, z_2)$  and the offspring PGF's  $f^{(i)}(z_1, z_2)$ , we can rewrite this into

$$P_1(z_1, z_2) = g(z_1, z_2)P_1(f_1(z_1, z_2)). \quad (13)$$

Iteration of this functional equation leads to the infinite-product expression (9), with  $N = 2$ .

**REMARK 2.4**

Polling models *with* and *without* switchover times are usually treated separately



in the literature, often via different approaches; the difficulty with simply letting the switchover times tend to zero in a polling model with switchover times is that the number of visits in an idle period tends to infinity, leading to degenerate distributions at such visit epochs. However, the following way out is possible. Let us assume that  $B$  in an empty system rests at, say,  $Q_1$ . For this situation Resing [15] shows, for the class of polling models with the branching property, that the joint queue length process at successive moments that  $B$  visits  $Q_1$  is again an MTBP - but now with immigration only in state zero. In [2] it is subsequently shown how the identical offspring PGF's of the MTBP's corresponding to the polling model *with* respectively *without* switchover times give rise to a strong relation between their respective joint queue length processes (see also [17]).

### 3 POLLING SYSTEMS WITH SMART OR PERSISTENT CUSTOMERS

In this section we shall generalize the polling model of Section 2 in two directions: polling models with arrival rates that depend on the server position ('smart customers') and polling models with feedback and customer routing ('persistent customers'). For each of these directions we outline (because of space restrictions without detailed proofs) how the model can be analyzed completely when the service discipline at each queue satisfies the branching property.

#### 3.1 Smart customers

In some polling applications, knowledge about the server position may influence the arrival rates of the customer types. In the director's example, the knowledge that the director will next turn to personnel matters may generate some new personnel tasks, while there is less hurry in creating tasks of another nature. Let us model this as follows, making a few adaptations in the model described in the previous section. The arrival process of customers at  $Q_i$ , when  $B$  is at  $Q_j$ , is Poisson with rate  $\lambda_{ij}$ ; the arrival process of customers at  $Q_i$ , when  $B$  is switching from  $Q_j$  to  $Q_{j+1}$ , is Poisson with rate  $\mu_{ij}$ . When the service discipline at each queue satisfies the branching property, with PGF  $h_i(z_1, \dots, z_N)$  at  $Q_i$ , then it is easy to check that the joint queue length process at successive moments that  $B$  visits, say,  $Q_1$  is an MTBP with immigration. The immigration PGF is given by (cf. (11)):

$$g(z_1, \dots, z_N) = \prod_{i=1}^N \sigma_i \left( \sum_{k=1}^i \mu_{ki}(1 - z_k) + \sum_{k=i+1}^N \mu_{ki}(1 - f^{(k)}(z_1, \dots, z_N)) \right). \quad (14)$$

In the case of gated service at  $Q_i$  the offspring PGF is (cf. (6)):

$$h_i(z_1, \dots, z_N) = \beta_i \left( \sum_{j=1}^N \lambda_{ji}(1 - z_j) \right), \quad (15)$$

and in the case of exhaustive service at  $Q_i$  the offspring PGF is (cf. (7)):

$$h_i(z_1, \dots, z_N) = \theta_i \left( \sum_{j \neq i} \lambda_{ji} (1 - z_j) \right). \quad (16)$$

The reasoning presented in Remark 2.3 should make it clear that the present model again gives rise to a functional equation of the type (13), iteration of which leads to an infinite-product expression for  $P_1(z_1, \dots, z_N)$  like (9). The PGF of the joint queue length distribution at the end of a switchover from  $Q_i$  to  $Q_{i+1}$  is simply expressed in the PGF at the beginning of that switchover (the end of a visit to  $Q_i$ ), and the latter PGF can be expressed in the PGF of the joint queue length distribution at the beginning of that visit to  $Q_i$  by substitution of the offspring PGF  $h_i(\cdot)$  at the  $i$ -th position in the PGF.

Several interesting special cases deserve further attention. E.g.,  $\lambda_{ij} = \Lambda p_{ij}$  and  $\mu_{ij} = \Lambda q_{ij}$  with  $p_{ij}, q_{ij} \geq 0$  and  $\sum_{i=1}^N p_{ij} = \sum_{i=1}^N q_{ij} = 1$  for all  $j$  corresponds to a fixed total arrival rate  $\Lambda$ . If the service discipline at each queue is gated (hence when  $B$  visits  $Q_i$ , he will only serve customers that were already present at the start of the session), the smartest thing for an arriving customer to do is to go to the *next* queue:  $\lambda_{i+1,i} = \mu_{i+1,i} = \Lambda$ , and  $\lambda_{ij} = \mu_{ij} = 0$  for all  $i \neq j+1$ . The most foolish behaviour, on the other hand, is represented by  $\lambda_{i,i} = \mu_{i,i} = \Lambda$ , and  $\lambda_{ij} = \mu_{ij} = 0$  for all  $j \neq i$ . The former choice clearly minimizes the waiting time of each individual arriving customer. Let us now moreover assume that  $B_i(\cdot) \equiv B(\cdot)$ . Then the above choice also minimizes, in the sense of stochastic ordering, the workload of the server. This may be proven using coupling methods; see [5] for the more restricted fully symmetric case.

In the case of identical service time distributions and fixed total arrival rate  $\Lambda$ , the work decomposition (1) still holds (check the level crossing argument presented in Remark 2.1), and EY can easily be calculated. But if not all service time distributions are the same, or the total arrival rate is not constant, then the whole work decomposition concept breaks down. Some reflection will make it clear that when switchover times are zero, even the concept of work *conservation* is destroyed.

### 3.2 Feedback and customer routing

In the director's example, a completed task may have to be reconsidered in future sessions. This feature can be incorporated in the model of Section 2 in the following way. A newly arriving customer at  $Q_i$  (Poisson with arrival rate  $\lambda_i$ ) is called a type- $(i, 1)$  customer. After completion of its service, it moves to  $Q_k$  with probability  $p_{ik}^{(1)}$ , becoming a type- $(k, 2)$  customer, and it leaves the system with probability  $p_{i0}^{(1)}$ . More generally, a type- $(i, j)$  customer denotes a customer at  $Q_i$  who has to be served for the  $j$ -th time; after having received service, it moves to  $Q_k$  with probability  $p_{ik}^{(j)}$ , and it leaves the system with

probability  $p_{i0}^{(j)}$ . A type- $(i, j)$  customer requires a service time at  $Q_i$  with distribution  $B_{ij}(\cdot)$ , with LST  $\beta_{ij}(\cdot)$ . We assume that  $p_{i0}^{(L)} = 1$  for all  $i$ , i.e., each customer requires at most  $L$  services.

Customer routing has hardly been studied in the context of polling, although several applications in token ring networks, robotics and computer systems exist; cf. Sidi et al. [16]. The latter paper analyzes the case of fixed transition probabilities  $p_{ij}$  of customers from  $Q_i$  to  $Q_j$ , with fixed service time distribution  $B_i(\cdot)$  at  $Q_i$  and exhaustive or gated service at all queues.

In this section we study the  $NL$ -dimensional queue length process  $\mathbf{X} = (\mathbf{X}_{11}, \dots, \mathbf{X}_{1L}; \dots; \mathbf{X}_{N1}, \dots, \mathbf{X}_{NL})$ , where  $\mathbf{X}_{ij}$  denotes the number of customers of type- $(i, j)$  at a moment at which  $B$  reaches  $Q_1$ .

The *branching property* of Section 2 has to be adapted in the sense that one has to distinguish  $L$  PGF's  $h_{ij}(z_1, \dots, z_N)$ ,  $j = 1, \dots, L$ , in  $Q_i$ .

It is easily seen that  $\mathbf{X}$  is an MTBP with immigration in each state. For the general case, determination of the offspring PGF's and the immigration PGF is somewhat involved. For example, one has to take the possibility into account that a customer is fed back to the same queue; and in the case of exhaustive service, such a customer may then receive more than one service during the same session. We shall refrain from formulating and proving the generalization of Theorem 2.2 here in its full generality. Instead, we illustrate the structure of the MTBP by considering a two-queue example with gated service at both queues. Similar to Remark 2.3, we denote by  $P_i(z_{11}, \dots, z_{2L})$  ( $G_i(z_{11}, \dots, z_{2L})$ ) the PGF of the joint queue length distribution when  $B$  reaches (leaves)  $Q_i$ . We have the following four relations:

$$P_1(z_{11}, \dots, z_{2L}) = \sigma_2(\lambda_1(1 - z_{11}) + \lambda_2(1 - z_{21}))G_2(z_{11}, \dots, z_{2L}), \quad (17)$$

$$G_2(z_{11}, \dots, z_{2L}) = P_2(z_{11}, \dots, z_{1L}; y_{21}, \dots, y_{2L}),$$

$$P_2(z_{11}, \dots, z_{2L}) = \sigma_1(\lambda_1(1 - z_{11}) + \lambda_2(1 - z_{21}))G_1(z_{11}, \dots, z_{2L}),$$

$$G_1(z_{11}, \dots, z_{2L}) = P_1(y_{11}, \dots, y_{1L}; z_{21}, \dots, z_{2L}).$$

Here, for  $i = 1, 2$ ,  $j = 1, \dots, L$ ,

$$y_{ij} := \beta_{ij}(\lambda_1(1 - z_{11}) + \lambda_2(1 - z_{21})) [p_{i0}^{(j)} + p_{i1}^{(j)} z_{1,j+1} + p_{i2}^{(j)} z_{2,j+1}].$$

Note that  $\beta_{ij}(\lambda_1(1 - z_{11}) + \lambda_2(1 - z_{21}))$  is the PGF of the numbers of new arrivals at the various queues during a type- $(i, j)$  service, and that  $p_{i0}^{(j)} + p_{i1}^{(j)} z_{1,j+1} + p_{i2}^{(j)} z_{2,j+1}$  is the PGF of the numbers of type- $(k, j + 1)$  customers,  $k = 1, 2$ , generated by the feedback of one type- $(i, j)$  customer.

Combination of the four relations in (17) leads to a recursion for  $P_1(z_{11}, \dots, z_{2L})$ , of similar form as (13), which can be solved iteratively.

### REMARK 3.1

We thus obtain the PGF of the joint queue length distribution at time points in which  $B$  reaches  $Q_1$ . But the four relations in (17) then also yield the PGF's

of the joint queue length distributions at time points in which  $B$  leaves  $Q_1$ , reaches  $Q_2$  and leaves  $Q_2$ . The PGF of the joint steady-state queue length distribution may also be determined from these results, once the service order at the queues is specified (e.g., serve type- $(i, j+1)$  before type- $(i, j)$  customers).

#### REMARK 3.2

The case of a *single* queue with feedback, contained in the present model, is also interesting in itself. We can obtain the joint queue length distribution of the numbers of customers that are present for the first, ...,  $L$ -th time, at the time points at which  $B$  starts a new session.

#### REMARK 3.3

Several variants and generalizations can also be handled in the framework of an MTBP. For example, one can allow zero switchover times between sessions, obtaining an MTBP with immigration only in state zero. Furthermore, instead of assuming  $p_{i0}^{(L)} = 1$ , we may also assume that  $p_{ik}^{(j)} = p_{ik}$  and  $B_{ij}(\cdot) \equiv B_i(\cdot)$  for all  $j \geq L$ ,  $k = 0, 1, \dots, L$ . The resulting MTBP still has a finite number of  $NL$  variables. This generalizes the model of Sidi et al. [16] in various ways.

We may generalize our model even further, while retaining the MTBP structure. For example, we can allow 'smart customers' *in combination with* feedback and routing; and we can also allow the possibility that a served customer not just feeds back, but branches into several customers: a task of type- $(i, j)$  that has been handled by the director may simultaneously give rise to tasks  $(k_1, j+1)$  and  $(k_2, j+1)$ . While these possibilities may make the job of a director rather complicated, they do not fundamentally complicate the analysis of his workload.

#### ACKNOWLEDGMENT

The author gratefully acknowledges several useful discussions with Sem Borst and Jacques Resing.

#### REFERENCES

1. S.C. Borst (1994). *Polling Systems*. PhD Thesis, Tilburg University.
2. S.C. Borst, O.J. Boxma (1994). Polling models with and without switchover times. *Report CWI, BS-R9421*.
3. O.J. Boxma (1989). Workloads and waiting times in single-server systems with multiple customer classes. *Queueing Systems* 5, 185-214.
4. O.J. Boxma, W.P. Groenendijk (1987). Pseudo-conservation laws in cyclic service systems. *J. Appl. Probab.* 24, 949-964.
5. O.J. Boxma, M. Kelbert (1994). Stochastic bounds for a polling system. *Annals of Oper. Res.* 48, 295-310.
6. J.W. Cohen, O.J. Boxma (1983). *Boundary Value Problems in Queueing System Analysis* (North-Holland Publ. Cy., Amsterdam).

7. M. Eisenberg (1972). Queues with periodic service and changeover times. *Oper. Res.* 20, 440-451.
8. C. Fricker, R. Jaïbi (1994). Monotonicity and stability of periodic polling models. *Queueing Systems* 15, 211-238.
9. D. Grillo (1990). Polling mechanism models in communication systems - some application examples. In: *Stochastic Analysis of Computer and Communication Systems*, ed. H. Takagi (North-Holland, Amsterdam), 639-658.
10. W.P. Groenendijk (1990). *Conservation Laws in Polling Systems*. PhD Thesis, University of Utrecht.
11. L. Kleinrock (1964). *Communication Nets - Stochastic Message Flow and Delay* (Dover, New York).
12. C. Mack, T. Murphy, N.L. Webb (1957). The efficiency of  $N$  machines unidirectionally patrolled by one operative when walking times and repair times are constants. *J. Roy. Statist. Soc. B* 19, 166-172.
13. D.P. Heyman, M.J. Sobel (1982). *Stochastic Models in Operations Research*, Vol. I (McGraw-Hill Book Company, New York).
14. M.P. Quine (1970). The multitype Galton-Watson process with immigration. *J. Appl. Probab.* 7, 411-422.
15. J.A.C. Resing (1993). Polling systems and multitype branching processes. *Queueing Systems* 13, 409-426.
16. M. Sidi, H. Levy, S.W. Fuhrmann (1992). A queueing network with a single cyclically roving server. *Queueing Systems* 11, 121-144.
17. M.M. Srinivasan, S.-C. Niu, R.B. Cooper (1993). Relating polling models with nonzero and zero switchover times. Report Univ. of Tennessee; to appear in *Queueing Systems*.
18. H. Takagi (1991). Application of polling models to computer networks. *Comp. Netw. ISDN Syst.* 22, 193-211.
19. H. Takagi (1990). Queueing analysis of polling models: An update. In: *Stochastic Analysis of Computer and Communication Systems*, ed. H. Takagi (North-Holland, Amsterdam), 267-318.

# Finite graphs in which the point neighbourhoods are the maximal independent sets

A.E. Brouwer

We determine all graphs as in the title.

In [vdH] certain graphs  $L_k$  occur. Noticing that they have the property mentioned in the title, I wondered whether they are the only such graphs. This note shows that, essentially, this is indeed the case.

For  $k \leq 1$ , let  $L_k$  be the graph with vertex set  $\mathbf{Z}_{3k-1}$  (the integers mod  $3k-1$ ) and adjacencies  $x \sim y$  iff  $y-x \in \{1, 4, 7, \dots, 3k-2\}$ . (Thus,  $L_1$  is the complete graph on two vertices, and  $L_2$  is the pentagon.) The *neighbourhood* of a vertex  $x$  is the set  $N(x) = \{y | y \sim x\}$ . A graph  $G$  is called *reduced* when distinct vertices have distinct neighbourhoods.

**THEOREM 0.1** *The finite reduced triangle-free graphs in which each independent set is contained in a point neighbourhood are precisely the graphs  $L_k$  ( $k \geq 1$ ).*

**PROOF:** First we show that the graphs  $L_k$  have the stated property. That they are finite, reduced and triangle-free is clear. Now it suffices to show that if  $S$  is an independent set contained in  $N(x)$ , and  $S \cup \{y\}$  is independent for some  $y$ ,  $y \not\sim x$ , then  $S \cup \{y\} \subseteq N(z)$  for some  $z$ . But  $y = x + 3i - 1$  or  $y = x + 3i$  for some  $i$  ( $1 \leq i \leq k-1$ ), and we can take  $z = x + 3i$  or  $z = x + 3i - 1$ , respectively.

Conversely, let the graph  $G$  have the stated property. We show that  $G \simeq L_k$  for some  $k \leq 1$ . Since  $\emptyset$  is independent,  $G$  has a vertex, and since a singleton is independent, each vertex has a neighbour, and since two nonadjacent vertices have a common neighbour,  $G$  has diameter at most 2. Clearly, if  $G$  is complete, then  $G \simeq L_1$ , so we may assume that  $G$  has diameter 2.

**Step 1.** *Given two nonadjacent vertices  $x, y$ , there is a unique vertex  $z = \sigma(x; y)$  such that  $y \sim z$  and  $N(x) \cap N(z) = N(x) \setminus (N(x) \cap N(y))$ .*



PROOF: The set  $\{y\} \cup N(x) \setminus (N(x) \cap N(y))$  is independent and hence contained in  $N(z)$  for some  $z$ . If it is also contained in  $N(z')$ , then, since  $G$  is reduced, the vertices  $z$  and  $z'$  have distinct neighbourhoods, and we may assume that  $z \sim u$ ,  $z' \not\sim u$  for some vertex  $u$ . But now  $\{x, u, z'\}$  is independent and not contained in a point neighbourhood. Contradiction.

Step 2.  $G$  is regular of valency  $k$ , say. If  $k > 1$ , then there is a pair of nonadjacent vertices with  $k - 1$  common neighbours.

PROOF: Let  $x, y$  be nonadjacent. If  $|N(y) \setminus N(x)| > 1$ , then choose  $u \in N(y) \setminus N(x)$ ,  $u \neq \sigma(x; y)$ . By the uniqueness part of the previous step, there is a vertex  $v \in N(x) \setminus (N(y) \cup N(u))$ , so that also  $|N(x) \setminus N(y)| > 1$ . Now  $(N(x) \cap N(y)) \cup \{u, v\}$  is independent, and hence contained in  $N(z)$  for some  $z$ . By downward induction on  $|N(x) \cap N(y)|$  it follows that  $|N(x)| = |N(y)|$  (since we have either  $|N(x)| = |N(x) \cap N(y)| + 1 = |N(y)|$ , or, by induction,  $|N(x)| = |N(z)| = |N(y)|$ ). Now regularity of  $G$  follows since its complementary graph  $\overline{G}$  is connected.

Step 3.  $G \simeq L_k$ .

PROOF: Let  $x_0 \not\sim y_0$  and  $|N(x_0) \cap N(y_0)| = k - 1$ . Define vertices  $x_i, y_i$  ( $i \in \mathbf{Z}$ ) by  $y_{i+1} = \sigma(x_i; y_i)$  and  $x_i = \sigma(y_i; x_{i-1})$ . Then  $|N(x_i) \cap N(y_i)| = k - 1$  and  $N(x_i) \cap N(y_{i+1}) = \{x_{i-1}\} = \{y_{i+2}\}$  for all  $i$ . By induction on  $j$  ( $1 \leq j \leq k - 1$ ) we see that  $|N(x_0) \cap N(x_{3j})| = k - j$ , and that  $x_0 \sim x_1, x_4, \dots, x_{3j-2}$  and  $x_{3j} \sim x_2, x_5, \dots, x_{3j-1}$ . Indeed, for  $j = 1$  this is clear, since  $x_0 = y_3$ . But  $x_{3j}$  and  $x_{3j+3}$  have the same neighbours except for  $x_{3j+1}, x_{3j+2}$ , and  $x_0$  and  $x_{3j}$  have the same neighbours except for the vertices  $x_{3i+1}, x_{3i+2}$  ( $0 \leq i \leq j - 1$ ), so  $x_0 \sim x_{3j+1}$  and similarly  $x_2 \sim x_{3j+3}$ . As long as  $x_0$  and  $x_{3j}$  have common neighbours, it follows that  $x_0 \neq x_{3j \pm 1}$ . However,  $x_0$  and  $x_{3k-1}$  have the same neighbours, so  $x_0 = x_{3k-1}$ . If there is a vertex  $z$  distinct from all  $x_i$ , then  $z$  is adjacent to either all or none of the  $x_i$ , contradiction, since  $G$  is triangle-free and connected.  $\square$

This theorem can be generalized by deleting the hypothesis that  $G$  is reduced. Now the conclusion becomes that  $G$  is a coclique extension of one of the  $L_k$ . (In particular, if  $G$  is regular, that  $G$  is a lexicographic product  $L_{k,m} := L_k[\overline{K_m}]$ .) Probably the finiteness hypothesis can be dropped as well, but the conclusion becomes more complicated, and I have not investigated this further.

The reason that the graphs  $L_{k,m}$  occur in [vdH] is that (for  $m \geq 3$ ) they have the maximal possible toughness  $t = n/k - 1$  for triangle-free regular graphs. (The toughness  $t(G)$  of a connected non-complete graph  $G$  with vertex set  $V$  is by definition  $\min |V \setminus X|/\omega(X)$  taken over all subsets  $X$  of  $V$  such that the number of connected components  $\omega(X)$  of  $X$  is at least two. Clearly,  $t(G) \leq (|V| - 2)/2$ .)

LEMMA 0.2 Let  $G$  be a connected non-complete graph. The toughness of the lexicographic product  $G[\overline{K_m}]$  equals  $\min |V \setminus X|/w(X)$ , where  $w(X)$  is the number of singleton components of  $X$  plus  $1/m$ -th of the number of other components of  $X$ , and  $X$  runs through the subsets of  $V$  with  $\omega(X) > 1$ .  $\square$

PROPOSITION 0.3 *The toughness of  $L_{k,m}$  equals  $\min(2 - \frac{1}{k}, 2 - \frac{2}{m(k-1)+1})$  ( $k \geq 1, m \geq 1$ ).*

PROOF: By the above lemma, we only have to investigate  $G = L_k$ . Taking  $X = N(0)$  shows that  $t(G) \leq (3k - 1 - k)/k = 2 - 1/k$ . Taking  $X = N(0) \cup \{2\}$  shows that  $t(G) \leq ((3k - 1) - (k - 1))/(k - 1 + 1/m) = 2 - 2/(m(k - 1) + 1)$ . Conversely, if  $\{x, y\}$  is an edge of  $G$ , then  $V \setminus (N(x) \cup N(y))$  is complete bipartite or a coclique. Thus, if some subgraph  $X$  of  $G$  has at least two non-singleton components, then  $w(X) = 2/m$  and  $|V \setminus X|/w(X) \geq 4/(2/m) = 2m \geq 2$  so that  $X$  does not determine the toughness. If  $X$  has precisely one non-singleton component, say containing the edge  $\{0, 3t + 1\}$ , then the set  $S$  of all vertices  $s$  such that  $\{s\}$  is a component of  $X$  is contained in one part of the bipartition on the vertices nonadjacent to both  $0$  and  $3t + 1$ ; say,  $S \subseteq \{3t + 3, \dots, 3k - 3\}$ . Now  $|V \setminus X|/w(X) \geq |N(S)|/(|S| + 1/m)$ . But when  $|S|$  is given,  $|N(S)|$  is minimal when  $S$  is ‘consecutive’:  $S = \{3a, 3a + 3, \dots, 3a + 3r\}$ , and then  $|N(S)|/(|S| + 1/m) = (k + r)/(r + 1 + 1/m)$ . This again is minimal when  $|S|$  is maximal, i.e., for  $t = 0$  and  $r = k - 2$ , and then  $|N(S)|/(|S| + 1/m) = 2 - 2/(m(k - 1) + 1)$ . Finally, if  $X$  has only singleton components, a similar but easier argument again shows that we get the smallest quotient by taking  $X$  a maximal coclique, and then this quotient equals  $2 - 1/k$ .  $\square$

#### REFERENCES

- [vdH] Jan van den Heuvel, *Degree and Toughness Conditions for Cycles in Graphs*, Ph.D. thesis, Techn. Univ. Twente, 1993.





# A Framework for Adaptive Networked Multimedia

Dick C.A. Bulterman

*The Multimedia Kernel Systems Project*  
*CWI: Centrum voor Wiskunde en Informatica*  
*The Netherlands*

Email: [dcab@cwi.nl](mailto:dcab@cwi.nl)

Accessing multimedia information in a networked environment introduces problems that don't exist when the same information is accessed locally. These problems include: competing for network resources within and across applications, synchronizing data arrivals from various sources within an application, and supporting multiple data representations across heterogeneous hosts. Often, special-purpose algorithms can be defined to deal with these problems, but these solutions usually are restricted to the context of a single application. A more general approach is to define an adaptable infrastructure that can be used to manage resources flexibly for all currently active applications. This paper describes aspects of a research program into adaptive, networked multimedia that started at CWI in 1991.

## 1. Problem Overview

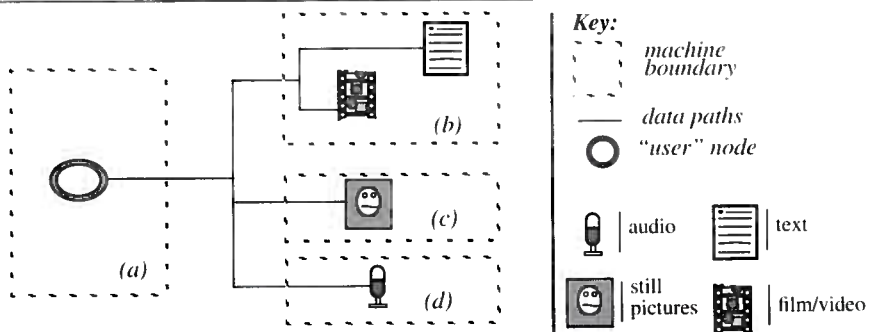
*Networked multimedia* is a generic term that describes a model of information distribution in which data sources are located separately from data sinks. Networked multimedia offers a number of advantages to applications: the network provides a convenient means of distributing information to other sites, it provides access to compute servers where special-purpose processing of multimedia data can take place, and it provides access to central servers that can be used to store the often vast amounts of data required to represent multimedia information fragments. At the same time, however, networked multimedia presents an application with a number of disadvantages when compared to accessing and manipulating multimedia data locally: the data delivery characteristics of the network are difficult to predict and control, the contention for critical system and data resources across the network makes balanced data access difficult to achieve, and differences among network hosts may make data objects difficult to share.

In order to make networked multimedia more useful to application designers and users, considerable effort needs to be devoted to studying the way that data servers, operating systems and network infrastructures provide access to time-sensitive data. Most current approaches define extensions to "conventional" means of accessing remote data to provide predictable network service and performance. For example, predictability is provided in data object servers (either file servers or database systems) by supporting efficient object storage and retrieval/delivery [DBL92,RV91] and in operating systems by supporting *quality of service* guarantees for delivery of (pos-

sibly) complex data types [ABL92,D91,GA91,HKN91,LMM92,TNP90]. At the network level, support for predictable multimedia is provided by, among others, admission control techniques that regulate use of resources and by technologies that provide deterministic network/data access [CSZ92, HM91,JST92,LG91,VF90,T90]. The basic premise of this work is that an application will request a data object (or a collection of objects) requiring a specific amount of resources during a specified time. If these resources are available, the application can execute; if not, the application is either delayed or it is denied access to the resources.

An implicit assumption in current approaches is that the application program bears a significant control burden in requesting and coordinating multimedia information. Consider, for example, the application environment shown in Fig. 1. Here, an application running on node *a* requests multimedia data from four sources located on three separate servers. The application must know the resource requirements of each *stream* of data it uses (where we use the term “stream” to mean either a single object or a collection of similarly-typed objects from a single server), it must coordinate the arrival and manipulation of multiple independent streams, and it must take any actions necessary if a given stream cannot be provided by the infrastructure. In general, the application software must control all content-based actions in or among the streams in the context of the application, while the infrastructure will control representation-based actions within a single stream in the context of service guarantees or network/server activity.

The content/representation division of control works well in environments where sufficient resources exist to handle an application’s request fully or where insufficient resources exist to handle the requests at all. It is less effective when an application can receive only partial support of its requests. In Fig. 1, suppose that one of the requested data streams could not be made available at the required level of service. An application may decide to skip this data object (or the collection of objects associated with that stream), or it may decide to substitute another data object or object server. In effect, the application program is engaged in a process of resource allocation: it is attempting to match its data needs to the resources available at various locations in the support infrastructure. Unfortunately, to do efficient resource allocation—even if this means only selecting from a set of available data streams—the applica-



**Figure 1.** Simple multimedia information client/server example. The client (a) is fed by three servers (b, c, d), one of which supplies two data types. (The structure of the client and the client’s application is not shown.)

tion needs to know how to best make use of the available infrastructure. This involves issues that most applications programs are ill-equipped to resolve. (It also requires applications to be rewritten when they are moved to new environments.) Alternatively, the operating system or the data servers could handle all resource allocation, but the (local) operating system will have only limited knowledge of the state of each of the servers and other applications active within the networked environment, and the data servers will be able to manage only their own streams, not other streams in the infrastructure.

This paper presents an alternative approach to supporting networked multimedia that is being studied within the Multimedia Kernel Systems group at CWI. Our work is aimed at studying coordinated application and infrastructure-based support for *adaptable* applications. Here, adaptable means that an infrastructure can be defined so that an application can adapt to the resources available at the time the application is run. The types of adaptability we consider include responding to (possibly transient) variations in the number and composition of network and remote resources that are available during application execution, as well as application and server support for heterogeneous collections of input/output devices. Our approach is based on two mechanisms. First, we define an application specification that explicitly describes the data objects used by an application, the manner in which the objects interact, and the available ranges of alternatives that are acceptable to the application at run-time. Second, we define an interface to the data objects that allows alternative representations to be selected at run-time by a process of application-transparent negotiation at run-time. This approach is specifically geared to applications that have a *document* or *presentation* structure. An authoring system (such as [RJM93]) can be used to generate a specification that can be accessed/executed at some later time. By allowing the execution to be adaptable, one specification can potentially allow an application to be available within a heterogeneous environment under a range of resource availability conditions. As will be discussed, this can help to reduce the high cost of authoring multimedia applications and it can lead to more efficient use of multimedia infrastructures.

In the sections below, we describe the framework for partitioning control responsibility within the system infrastructure to support adaptable applications. This framework, the *Amsterdam Multimedia Framework*, distinguishes itself from other approaches because of the cooperative and distributed nature of resource allocation and control among a collection of independent multimedia applications.

## 2. Requirements for Adaptable Networked Multimedia

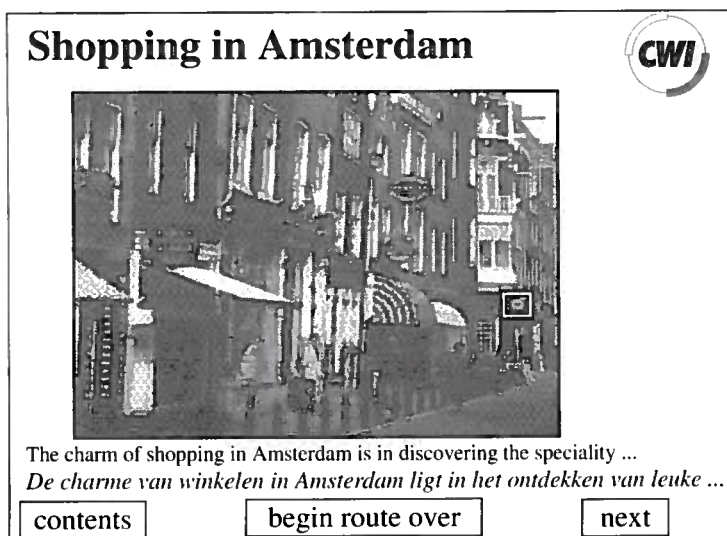
In order to support adaptable networked multimedia, an underlying framework is necessary that defines how information is structured, composed, accessed, and manipulated, as well as how it is stored and transmitted among sources and sinks. In this section, AMF: the *Amsterdam Multimedia Framework* is presented. To put AMF in context, its description is prefaced with a discussion of the type of multimedia applications it was intended to support and a review of the control issues that the framework must address.

### 2.1. Multimedia Application Descriptions: The Document

Our abstraction for organizing multimedia information is the *document*. A document defines a collection of *data objects* and a description of how these objects interact. Each object may consist of previously-stored information or information that is generated dynamically. Such information can be of either a single data type (such as pure audio or video) or of a composite data type (such as video with embedded audio). An active document is called a *presentation*.

Fig. 2 provides an example of a document-based multimedia application—in this case, a fragment of a walking tour of Amsterdam. This fragment contains a title bar using text data, a description of typical shopping street using video data, several “buttons” using text data that control navigation through the document, a CWI logo using still-image data, and two sets of captions (one in English, one in Dutch) using text data. The document from which this example is taken also has two sound tracks (one in Dutch, one in English) that provide audio commentary during the tour. The data objects can be stored on various servers located throughout the environment. When the document is accessed, each of the individual object streams is sent to a document *player*, which implements any high-level (non-embedded) synchronization constraints among the streams (such as matching the subtitle text with the audio data). Each document, such as the tour of Amsterdam in our example, is specific to a particular application; the player is a general-purpose program that must be able to play many different documents.

The primary advantage of using a document model is that it provides an explicit behavioral specification. This behavioral description can be used to fetch individual data objects by a player, but it can also be used prior to execution to analyze expected application resource use and feasibility for a given environment [BZ92a]. Assuming



**Figure 2.** An example multimedia application.

*The rectangles along the bottom are navigation controls; the square in the picture is a hyperbutton. The lines of text are captions that accompany multi-lingual audio.*

the specification was defined to run in a general-purpose environment (that is, it was not designed for use on one particular platform), the specification can also be used to determine how (and if) the synchronization needs of the application can be supported at run-time [BRL91].

Creating documents using authoring systems or program-based toolkits is typically an arduous task. One motivation for investigating adaptable networked multimedia was to provide reduce the overall effort of producing multimedia presentations by a means of reusing document structures in multiple environments once they were authored [BRL91,HBR93b].

## 2.2. Supporting Adaptable Documents:

### *Data-Representation and Document-Content Issues*

During analysis of a document, it is typically assumed that the specification provides a precise description of the needs and characteristics of the application. Our work investigates the use of a specification as a guide to *possible* resource and data use, depending on the resources available at execution time of the document. While pre-execution analysis can provide a useful first step in determining specification feasibility, it cannot resolve all of the issues that may influence the run-time needs or run-time behavior of an application. In defining a basis for adaptable documents, two classes of issues can be identified that influence document analysis and support: issues associated with the physical representations of multimedia data and issues associated with the content-based interactions of users with multimedia data.

#### 2.2.1. Representation-based issues.

One major difference between multimedia data and “conventional” electronic data is that multimedia information can require specific service guarantees to preserve synchronization properties of the data. These properties are the consequence of how multimedia data is represented, not the meaning of the data itself. While the representations of each data type vary, there are several common issues that are relevant for all time-sensitive multimedia data:

- *intra-object synchronization*: each component can have synchronization constraints that are related to the type of data being retrieved. For example, the video, audio and caption-text data in Fig. 2 each have their own synchronization constraints. These constraints must be supported by the source environment, the network infrastructure being traversed and the destination environment. These constraints can usually be managed on an end-to-end basis [D90,D91];
- *inter-object synchronization*: in general documents, data will be encoded in separate streams of objects, each of which may be located at different hosts. While inter-object synchronization is often controlled in the context of an application, the composite transfer of data may need to be coordinated to improve system efficiency. For example, synchronization of audio data and caption-text can be done by the application, but it can be done more efficiently using markers placed in the data objects and evaluated by the support software;
- *heterogeneity*: in general environments, all of the presentation workstations will not be identical. Information may need to be adapted at either the source or the



sink to meet the needs of a presentation environment, where the adaptation process may itself have an influence over which parts of a document are available to a user—a process that may also impact scheduling, resource allocation and synchronization with the network.

Bandwidth management can also be included among the representation-related issues. In spite of the trend toward faster networks and more highly-encoded information, the transfer capacity of the various interconnects will remain a critical resource that must be managed—either because application demands will grow or because multiple types of networks will coexist at a site, requiring a degree of coordination and management when allocating local and global resources efficiently.

### 2.2.2. *Content-based issues.*

The reason for isolating representation-based issues is to consider ways of providing other than worst-case resource allocation in an adaptable environment. In a similar manner, the actions that occur based on the content of a document will also affect the way that documents are fetched, composed and delivered. These include:

- *user selectivity*: not all of the information available in a document may be used each time the document is accessed; for example, although the document in Fig. 2 supports multilingual audio and/or captions, users usually don't want to hear or read all of the available languages simultaneously. (Note that the selection of desired information is made at run-time—not author-time—and that the selection may be influenced by the facilities available on a given playback platform.)
- *presentation non-linearity*: the order in which objects are accessed and presented depends on the document structure *and* the result of user interaction at run-time. For example, users may want to jump around in a document by scrolling forward or backward or by following *hyperlinks* that have been defined statically or dynamically in the document; in Fig. 2, a small rectangle is visible over a traffic sign in the mid-right portion of the street—selecting this button will transfer the user to a section discussing the merits of getting around by bicycle, car and tram in the city.
- *user flexibility*: in general, documents are activated because a user wishes to obtain information. Given a choice, it is our experience that users will tolerate a lower quality presentation instead of being denied access to a presentation totally. Such lower quality may manifest itself as (slight) delays in the presentation of parts of a document or in the substitution of a lower-resolution form of information for a higher-resolution one. (The term “resolution” is used broadly: it could mean substituting a piece of text for a picture or an audio fragment for a piece of video.)

Each of these factors affects the support mechanisms required to provide adaptability in a document. The notion of *user selectivity* means that static analysis of a document before it is executed may not provide an insight into how a document will actually be used. Similarly, *presentation non-linearity* could result in “jumping” to various parts of a document, each with its own quality of service requirements. As a result, efficient use of an infrastructure will require dynamic rather than static assignment of resources across the network. *User flexibility* means that some degree of run-time

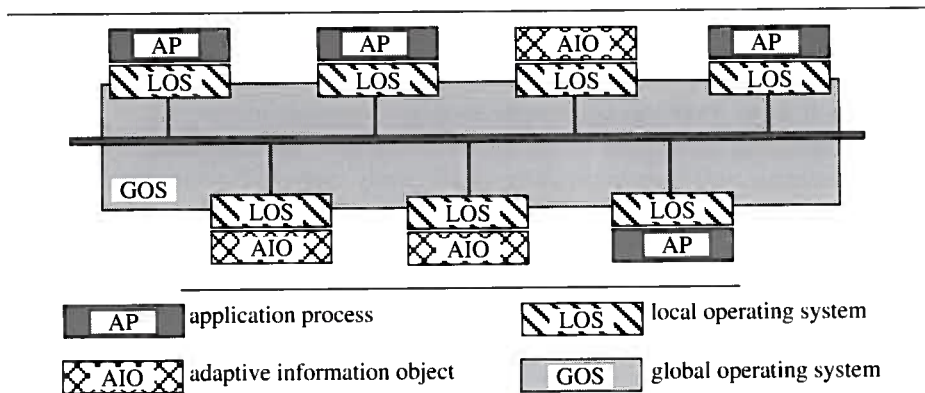


Figure 3. AMF "active" components.

negotiation may need to be supported so that the information presented to the user can be matched to the resources available at the time individual data access requests are made.

### 3. AMF: The Amsterdam Multimedia Framework

Although many of the techniques required to support representation-based control and, to a lesser extent, content-based control can be taken from existing research results, it is important that these results be applied within a framework that provides an explicit partitioning of control concerns across components in a network infrastructure. This provides a definition of the scope of each technique and can result in better interaction among components. The AMF provides this partitioning for our work.

Fig. 3 illustrates AMF. Here, many applications (AP) communicate with *adaptive information objects* (AIOs) via an infrastructure that is managed by a set of local operating systems and a global operating system. The LOSs and GOS coordinate resource allocation, while the APs and AIOs request and deliver information, respectively. Note that the AMF does not solve the multimedia data transfer problem, it only characterizes the components in an environment and it indicates their interactions. Individual models still need to be developed that implement the general functionality of the framework.

The general structure of AMF is similar to client/server models of networked computing. The difference is that within AMF, the control of multimedia is a cooperative process that requires content-based coordination among all components. For example, assume that one of the APs requests two object streams, each from separate AIOs on two separate hosts. Assume further that one of the AIO is able to meet the service quality request of the application directly, while the other one is not. In this case, both could inform the application of their available degree of service (leaving the application to select an appropriate recovery action) or the two AIOs could communicate with each other to determine if there was a common level of service that both could provide that was acceptable for that application. This could be possible if:

- each of the AIOs was aware of the other's presence,
- each AIO was aware of other's service constraints, either directly from copies of the application specification or by intervention of the GOS and/or each LOS, and



- both AIOs were aware of the range of options acceptable to the application and supportable by the LOS/GOS.

Standard client/server architectures do not provide a basis for this type interaction. As we will show, AMF was specifically designed to provide it.

The underlying assumption of the AMF is that none of the individual components in a transfer has sufficient information to efficiently control resource allocation and inter-object synchronization. A pair of components, such as an AP and a single AIO, is also insufficient, since both end-points could *think* they could provide a degree of service without realizing that the network interconnect was overloaded or that other applications were about to request service. Instead, by using the information in a document specification to be able to look ahead into an application's future behavior, new techniques for resource allocation in its broadest form can be studied for each component. Unlike typical client/server models, these techniques are not based on a notion of lower-level protocol data independence, but rather, on distributing control so that support decisions can be made in light of the needs of applications throughout the network. The scope of AMF control activity is discussed in the following paragraphs.

### 3.1. *The application process (AP).*

The role of the AP is to supply the other components within the AMF with a specification of the object streams used by an application, as well as a definition of any inter-object-stream synchronization requirements and a set of options that can be used in providing adaptable control (see section 3.1 for an example). The AP itself functions like the *player* described in section 2.1: it provides a control interface to the user to provide high-level interaction with the network. ("High-level" means operations like *start, stop, pause, fast-forward, seek*, etc.)

In terms of the issues defined in section 2.2, the player provides a user interface to the execution environment, allowing the user to select the parts of a document that need to be played, to navigate through the document and to define the degree to which a document can be adapted. (For example, if a user plays a document on a disconnected portable machine, more tolerance for missing data object may be specified). The player has only a limited role in *implementing* any representation or content-based control operations other than possibly supporting heterogeneous data—this is because the player is a general-purpose interface, while the specification provides the other AMF components with the information necessary to adapt to the needs of the multimedia application.

### 3.2. *The local operating system (LOS).*

The LOS serves as a scheduling authority that controls access to I/O devices attached to the local workstation. The LOS would typically allocate resources based on its architecture-specific knowledge of the local operating environment and the document specification provided by the application. While the LOS has the responsibility for controlling the flow of information in and out of the local environment—including presenting information to and receiving information from the network controller(s)—it cannot control activity outside of its environment because it has only a limited view

of what is happening across the network: individual sources may need to sub-sample or pre-synchronize streams within a document or there may be other active documents generating competing requests for resources that are totally outside the scope of a local operating system.

The LOS can participate in managing various data streams for an application by implementing a negotiation process among data providers within the network. The LOS (together with the LOS of an information provider) can also be used to implement the end-to-end protocols associated with intra-object synchronization. Both of these types of service can be provided directly or in conjunction with a GOS. In general, local resource control should be as light-weight as possible; this provides the user with a responsive environment and the rest of the network with a non-intrusive element.

### 3.3. *The global operating system (GOS).*

The role of the GOS is to allocate resources on a network-wide basis. It has a view of network activity that is more comprehensive than the APs, the AIOs or the LOS, since it can coordinate activity among independent applications that use the central network but which originate from different workstations. The GOS can provide support that is independent of any particular workstation architecture, acting as moderator or mediator if conflicts arise. (Such a role may be more appropriate in wide-area implementation than in local area networks.) Note that it would be possible for a given implementation model to combine the functions of the LOS and the GOS, although from the point of view of the framework, it is important to recognize that the functions served by both abstractions are different. The primary practical motivation for keeping the LOS and GOS separate is that workstations in a heterogeneous environment cannot be assumed to have similar local operating systems. (They will also most likely have local systems that cannot be altered or adapted to provide extended multimedia support.) The architecture of the GOS allows global concerns to be factored out of the local environment, even to the point that it is possible to design attached-processor implementations supporting GOS functions [BL91].

### 3.4. *Adaptive information objects (AIO).*

The AIO provides applications with an interface to stored, synthesized or interactive information. In supporting access requests, the AIO separates the notions of *multimedia information* and *multimedia information representation*. In this way, AIO presents an abstract interface that is used to control access to one of several representations of a block of 'information.' For example, it can be used to substitute an audio description of a video if the user, the user's workstation, the network or the server's host cannot support video delivery. By providing alternative representations of information, the AIO provides *quality of information* support rather than *quality of service* support. (The latter term is more appropriate for representation-dependent manipulations, while the former is more appropriate for content-based selection.) Note that the AIO does not give you something for nothing: it simply provides a general framework that needs to be filled in by data-dependent code and, if appropriate, alternative representations.

Based on the contents of an application specification, the AIO can enter a process of negotiation to provide an application with an appropriate representation of information

that meets the constraints of conditions in the AP, LOS and GOS. The goal of the AMF is that individual implementation models do this negotiation transparently; the motivation for this is that by the time a user goes through the operations necessary to interactively select an alternative representation, the resource constraints that prompted the original negotiation request could have changed. We also assume that most authors would prefer to select the alternative representations which should be used, based on the author's insight into the application domain. (Note that individual AP implementation models may provide both types of control.)

#### 4. Current Status and Summary

The AMF is based on the assumption that resource control in a multimedia network should be adaptable, and that the adaptive process should be distributed over the application, the local operating system, the global (distributed) operation system and the AIOs involved in a transfer. Each of these layers has a specific insight that is important in controlling multimedia transfers. Although each of these insights are necessary, AMF also attempts to limit the scope of any one layer by giving each layer a specific set of concerns to process.

Support for AMF is an on-going research activity. At present, an authoring system has been developed to capture document models in a form that are suited to implementation within the AMF. We have also defined a hyper-information architecture that can be used to describe application-level interactions at runtime within an AMF context. Of the implementation projects, the CMIF authoring environment and its run-time player is the most advanced, while support for general AIO manipulations is at an early stage. Work on the AIO is tied to the development of an LOS/GOS infrastructure and the development of semantic facilities that can be provided to support a wide range of resource, synchronization, and representation control operations. We have performed initial presentation mapping experiments [BW93], but it is too early to draw any conclusions on the utility of this approach.

All of our activity in the Multimedia Kernel Systems project is aimed at understanding the basic relationships that exist in supporting multiple multimedia applications in a heterogeneous network environment. In the current version of our work, this global function is replaced by a separate client and server pair that transparently negotiate the format of the information to be used to satisfy a particular object reference based on the characteristics of the target system, the load on the network, the types of alternative representations that the client will accept, etc. This transparent interaction is important because it offers an opportunity for the system to respond quickly to transient conditions in the environment, but it is difficult to achieve in the light of closed operating systems and multimedia devices. It is our long-term intention to investigate the support of distributed operating systems technology that will allow CMIF specifications (or its successor) to get passed among all of the components of the AMF, each of which will pick out the information it needs to support the synchronization and resource requirements of the application [B92].

## Acknowledgments

The general frameworks and various implementation projects described in this article have developed during the past three years as part of the Multimedia Kernel Systems project at CWI. Chief contributors to this project have been Guido van Rossum, Lynda Hardman, Jack Jansen, K. Sjoerd Mullender, Robert van Liere, and Dik Winter. The researchers wish to state their appreciation to Prof.dr. P.C. Baayen, during whose tenure this research program started—and without whose granting of critical resources during the startup phase of the project, this work would never have seen the light of an optical fiber.

## References

- 1 [ABL92]Anderson, T.E., B.N. Bershad, E.D. Lazowska, and H.M. Levy, "Scheduler Activations: Effective Kernel Support for the User-Level Management of Parallelism," *ACM Transactions on Computer Systems*, 10(1), February 1992.
- 2 [B92]Bulterman, D.C.A., "Synchronization of Multi-Sourced Multimedia Data for Heterogeneous Target Systems," *Proc. 3rd Int. Workshop on Network and Operating Systems Support for Digital Audio and Video*, San Diego, Nov. 1992.
- 3 [BL91]Bulterman, D.C.A. and R. van Liere: "Multimedia Synchronization and Unix," *Proc. 2nd Int. Workshop on Network and Operating Systems Support for Digital Audio and Video*, Heidelberg, Nov. 1991.
- 4 [BRL91]Bulterman, D.C.A., G. van Rossum and R. van Liere: "A Structure for Transportable, Dynamic Multimedia Documents," *Proc. Summer 1991 Usenix Conference*, Nashville TN, June 1991.
- 5 [BW93]Bulterman, D.C.A. and D. T. Winter, "A Distributed Approach to Retrieving JPEG Pictures in Portable Hypermedia Documents," *Proc. IEEE Symp. on Multimedia Technologies and Future Applications*, Southampton, UK, April 1993.
- 6 [BZ92a]Buchanan, M.C. and P. T. Zellweger, "Scheduling Multimedia Documents Using Temporal Constraints", *Proc. 3rd Int. Workshop on Network and Operating Systems Support for Digital Audio and Video*, San Diego, CA, Nov. 1992.
- 7 [BZ92b]Buchanan, M.C. and P. T Zellweger, "Specifying Temporal Behavior in Hypermedia Documents", *Proceedings of the ACM ECHT'92 Conference on Hypertext*, Milano, Italy Nov 30 - Dec 4 1992.
- 8 [CSZ92]Clark, D.D., S. Shenker, and L. Zhang. "Supporting Real-Time Applications in an Integrated Services Packet Network", in *Proceedings of ACM SIGCOMM'92*, 1992.
- 9 [D90]Ferrari, D. "Client Requirements for Real-Time Communication Services", *IEEE Communications Magazine*, November 1990. See also RFC 1193, November, 1990.
- 10[D91]Ferrari, D, "Design and Implementation of a Delay Jitter Control Scheme for Packet-Switching Internetworks," *Proc. 2nd Int. Workshop on Network and OS Support for Digital Audio and Video*, Heidelberg, Nov. 1991.

- 11[DBL92]Danthin, A., Y. Baguette, G. Leduc, and L. Leonard, "The OSI 95 Connection-mode Transport Service: The Enhanced QoS," in *Proceedings of 4th IFIP Conference on High Speed networking*, Dec. 1992, Liege, Belgium.
- 12[FSM91]Fujikawa, K., S. Shimojo, T. Matsuura, S. Nishio and H. Miyahara, "Multimedia Presentation System 'Harmony' with Temporal and Active Media", *Proc. USENIX Multimedia Conference*, June 1991 Nashville TN.
- 13[GA91]Govindan, R. and D.P. Anderson. "Scheduling and IPC Mechanisms for Continuous Media," in *Proceedings of the Thirteenth ACM Symposium on Operating Systems Principles*, October 1991.
- 14[HBR93a]Hardman, L., D.C.A. Bulterman, and G. van Rossum, "The Amsterdam Hypermedia Model: Extending Hypertext to Real Multimedia," *Hypermedia Journal*, Vol 5(1), May 1993
- 15[HBR93b]Hardman, L., D.C.A. Bulterman and G. van Rossum, "Structured Multimedia Authoring," *Proc. ACM Multimedia '93*, Anaheim CA, August 1993.
- 16[HKN91]Hanko, Kuerner, Northcutt, Wall, "Workstation Support for Time-Critical Applications," *Proc. 2nd Int. Workshop on Network and Operating Systems Support for Digital Audio and Video*, Heidelberg, Nov. 1991.
- 17[HM91]Hayter, M. and D. McAuley, "The Desk Area Network," *Technical Report No. 228*, Cambridge University Computing Laboratory, 1991.
- 18[JST92]Jeffay, K. D.L. Stone T. Talley, and F.D. Smith, "Adaptive, Best-Effort Delivery of Digital Audio and Video Across Packet-Switched Networks," *Proc. 3rd Int. Workshop on Network and Operating Systems Support for Digital Audio and Video*, San Diego, CA, Nov. 1992.
- 19[LG91]Little, T.D.C and A. Ghafoor, "Scheduling of Bandwidth-Constrained Multimedia Traffic," *Proc. 2nd Int. Workshop on Network and Operating Systems Support for Digital Audio and Video*, Heidelberg, Nov. 1991.
- 20[LMM92]Lesley, I.M., D. McAuley, and S.J. Mullender, "PEGASUS - Operating Systems Support for Distributed Multimedia Systems," *ACM Operating Systems Review* 1(27), January, 1992
- 21[RV91]Rangan, P.V. and H. Vin, "Designing File Systems for Digital Video and Audio," *ACM Operating Systems Review*, 25 (5), 1991.
- 22[RJM93]van Rossum, G., J. Jansen, K.S. Mullender and D.C.A. Bulterman, "CMIFed: A Presentation Environment for Portable Hypermedia Documents," *Proc. ACM Multimedia '93*, Anaheim CA, August 1993.
- 23[T90]Topolcic, C., "Experimental Internet Stream Protocol, Version 2 (ST-II), *Internet RFC 1190*, Oct. 1990.
- 24[TNP90]Tokuda, H., T. Nakajima, and P. Rao, "Real-time Mach: Towards a Predictable Real-Time System," in *Proceedings of USENIX Mach Workshop*, October 1990.
- 25[VF90]Verma, D. and D. Ferrari. "A Scheme for Real-Time Channel Establishment in Wide-Area Networks", *IEEE JSAC*, Vol. 8, No. 3, April 1990.

# Yet Another Lecture on the Icosahedron<sup>‡</sup>

Arjeh M. Cohen\*

## 1 Introduction

In the period 1984–1992, one of my research goals was to establish the existence of certain (non-abelian) finite subgroups of exceptional Lie groups. My main collaborators on this topic were R.L. Griess, Jr. and D.B. Wales.

Some of these embeddings could be done entirely by theoretic arguments and hand calculations. For the others, the best we could do was to reduce the problem to a form suitable for the computer to finish off the computations. I would like to sketch the nature of such computations using a few simple examples, thereby illustrating the improved possibilities of polynomial system solving.

Also, I will sketch roughly how, very recently, Serre has shown that the reduction techniques we developed can be pushed so far that at least the most spectacular of the existence proofs can also be done without recourse to a computer.

I will write about one more issue, as it represents some of the interactions between mathematics and computer science that Cor Baayen enjoys seeing. It is the use of rewriting techniques in group theory, in much the same way they are used in Buchberger's Gröbner basis approach to polynomials—the technique that lies at the heart of the present polynomial system solvers.

Before going into some of these details, I will present an elementary introduction into group representations. The quaternion group (of order 8) and the icosahedral group (of order 120) will be used to illustrate the ideas. The rotation group of the latter is the nonabelian finite simple group of smallest order. This may explain a bit why it is a gateway to understanding finite simple groups.

## 2 The quaternion group

Let  $G$  be a finite group. A classical group theoretic question is to determine all possible realisations of  $G$  as a group of matrices. To be more precise, one would like to know all possible morphisms  $\rho : G \rightarrow GL(V)$  from  $G$  into the group  $GL(V)$  of all linear transformations of a vector space  $V$  over a fixed field  $k$ .

---

\*Written for Cor Baayen in gratitude for his rôle in my professional life.

<sup>‡</sup>Inspired by the 100 year old [K] and the introduction to [BCN].



Such a morphism is called a linear representation of  $G$  (over  $k$ ). If  $n = \dim V$ , then  $\rho$  is said to be  $n$ -dimensional.

In fact, we are only interested in representations up to equivalence; we recall that a representation  $\rho' : G \rightarrow GL(V')$  over  $k$  equivalent to  $\rho$  if there is a linear invertible map  $A : V \rightarrow V'$  such that  $\rho(g) = A^{-1}\rho'(g)A$  for all  $g \in G$ .

Another restriction we make here is the field: we shall only look at representations in characteristic 0 here. In fact, we shall take  $k = \mathbf{C}$  for the time being, in which case we speak of complex representations. Consider representations of the quaternion group

$$Q = \{\pm 1, \pm \mathbf{i}, \pm \mathbf{j}, \pm \mathbf{k}\}$$

with multiplication determined by

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = -1 \quad \text{and} \quad \mathbf{ij} = -\mathbf{ji} = \mathbf{k}$$

(and the fact that  $-1$  is a central element of order 2).

It makes sense to restrict to irreducible representations, i.e., those that have no “subrepresentations” but for the zero-dimensional and the full vector space  $V$ . Every complex representation can be decomposed as a sum of irreducible representations.

There always is the trivial representation, sending every element to the  $1 \times 1$  matrix (1). But  $Q$  can also be represented as a group of  $1 \times 1$  matrices by the morphism

$$\pm 1 \mapsto 1, \quad \pm \mathbf{i} \mapsto 1, \quad \pm \mathbf{j} \mapsto -1, \quad \pm \mathbf{k} \mapsto -1.$$

The trivial representation and this one are not the only 1-dimensional representations. There are two more 1-dimensional representations. (one sending  $\pm \mathbf{j}$  to 1, the other sending  $\pm \mathbf{k}$  to 1, instead of  $\pm \mathbf{i}$ ). None of these provides a faithful (that is, injective) representation. But the following 2-dimensional representation is faithful:

$$\begin{aligned} \pm 1 &\mapsto \pm \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, & \pm \mathbf{i} &\mapsto \pm \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix} \\ \pm \mathbf{j} &\mapsto \pm \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, & \pm \mathbf{k} &\mapsto \pm \begin{pmatrix} 0 & i \\ i & 0 \end{pmatrix}. \end{aligned}$$

How do we find such a representation? Suppose  $Q$  has a 2-dimensional faithful representation  $\rho$ . Then, from the fact that  $\rho$  must be irreducible (sums of 1-dimensional representations are not faithful!), we know that  $\rho(1)$  is the identity matrix  $I_2$ , and, similarly, that  $\rho(-1) = -I_2$ . Furthermore,  $\rho(\mathbf{j})$ , being an element squaring to  $-I_2$ , can be chosen, up to conjugacy, to be

$$\rho(\mathbf{j}) = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

Now all we need to find is  $\rho(\mathbf{i})$ , because the morphism law  $\rho(xy) = \rho(x)\rho(y)$  will then determine the images of all remaining elements. Write

$$\rho(\mathbf{i}) = \begin{pmatrix} a & b \\ c & d \end{pmatrix},$$

for certain  $a, b, c, d \in \mathbf{C}$ . Working out that  $\rho(\mathbf{i})^2 = -I_2$  and that  $(\rho(\mathbf{i})\rho(\mathbf{j}))^2 = -I_2$  yields a set of equations in these four variables. Solving these equations readily leads to the conclusion that, for any  $a, b \in \mathbf{C}$  with  $a^2 + b^2 = -1$ , the morphism  $\rho_{a,b}$  given by

$$\begin{aligned} \pm \mathbf{1} &\mapsto \pm \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, & \pm \mathbf{i} &\mapsto \pm \begin{pmatrix} a & b \\ b & -a \end{pmatrix} \\ \pm \mathbf{j} &\mapsto \pm \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, & \pm \mathbf{k} &\mapsto \pm \begin{pmatrix} -b & a \\ a & b \end{pmatrix} \end{aligned}$$

is a 2-dimensional representation of  $Q$ .

The choice  $a = i, b = 0$  gives the representation  $\rho$  mentioned before. Any representation  $\rho_{a,b}$  is conjugate to  $\rho$ ; if  $b \neq 0$ , then the matrix

$$A = \begin{pmatrix} i-t & -s \\ s & i-t \end{pmatrix},$$

where  $t = i - s(a+i)/b$ , conjugates  $\rho$  to  $\rho_{a,b}$ .

The four 1-dimensional representations and the 2-dimensional one are all we need to build up the full set of linear representations of  $Q$  over the field  $\mathbf{C}$ . Up to conjugacy, these are the only irreducible representations. The theory on which this assertion is based is known as character theory. A consequence of this beautiful theory is that the sum of all squares of the dimensions of the distinct (non-conjugate) irreducible representations equals the order of the group. Here, this amounts to

$$1^2 + 1^2 + 1^2 + 1^2 + 2^2 = 8.$$

It is of interest for the study of representations over finite fields to know minimal extension fields  $k$  over the rationals such that the represented group embeds in a version of  $GL(V)$  defined over  $k$ . A look at  $\rho$  for the quaternion group shows that the 2-dimensional representation is realised over  $\mathbf{Q}(i)$ . But if we take  $a = 3, b = 2\sqrt{-2}$ , then  $\rho_{a,b}(Q)$  is realised over the field  $\mathbf{Q}(\sqrt{-2})$  and clearly no conjugate of  $\rho$  can be realised over  $\mathbf{Q}$ . This indicates that there is no minimal extension field of  $\mathbf{Q}$  attached to the class of representations in  $GL(V)$  containing  $\rho$ . Later we shall see that this seeming lack of a unique minimal “splitting field” for  $Q$  is due to the restricted notion of representation handled here.

### 3 The group of the icosahedron

The isometry group of the icosahedron (the usual Platonic solid in 3-dimensional Euclidean space) can be abstractly defined as the group  $W$  generated by the 3 elements  $x, y$  and  $z$  subject to the relations

$$x^2 = y^2 = z^2 = 1,$$



$$(xy)^3 = (yz)^5 = (xz)^2 = 1.$$

Such a definition by means of generators  $X = \{x, y, z\}$  and relators  $Y = \{x^2, y^2, z^2, (xy)^3, (yz)^5, (xz)^2\}$ , often succinctly written as

$$W = \langle X \mid Y \rangle,$$

is called a presentation by generators and relations.

The abstract presentation of the icosahedral group can be understood by looking at the classical icosahedron. Cut the surface of the icosahedron into domains by means of the hyperplanes that are the mirrors of reflections preserving the icosahedron. By doing so, and selecting one of the 120 domains, we can identify the three generators  $x, y, z$  with the reflections whose mirror hyperplanes bound the selected domain of the icosahedron.

Surprisingly enough, we can go the other way around: by constructing the most general graph whose vertices are (transitively) permuted by the elements of the group  $W$ , we find the icosahedral graph. Let us perform this construction in some more detail. Start with a vertex, and label it with the trivial element of the group. We make three neighbours of 1, labeled  $x, y, z$  (the three generators of the group  $W$ ). We also label the edges  $\{1, x\}, \{1, y\}, \{1, z\}$  with the respective labels  $x, y, z$ . The graph under construction must allow for an action (on the left) of the generators as a group of automorphisms. It will be most convenient to think of the graph under construction as one whose edges are labeled with  $x, y, z$ .

Since the three generators are elements of order 2 (see the first line of relations for  $W$ ), we can think of view each of them as a permutation interchanging the vertices of an edge on 1 whose label coincides with its name. The vertex of that edge distinct from 1 will then be labelled with that name as well. But the picture is still far from being complete: it has not yet been described to which node  $y$  maps the vertex  $x$ . Left multiplication by  $y$ , being an automorphism of the graph, must send the edge  $\{1, x\}$  labeled  $x$  to the edge  $\{y, yx\}$ , labeled  $x$ . Thus, we find a new vertex  $yx$ , connected to  $y$  with an edge labeled  $x$ . Leaving alone  $z$  for a while, we continue this way, joining  $xyx$  to  $yx$  with an edge labeled  $y$ , joining  $xyxy$  to  $xyx$  with an edge labeled  $x$ . Then we reach  $xyxyx$ , which is joined to  $xyxy$  with an edge labeled  $y$ . The relation  $(xy)^3 = 1$  (on the second line of relations for  $W$ ) and the fact that  $x$  and  $y$  are their own inverses (being of order 2), tell us that the element  $xyxyxy$  coincides with  $x$ . Moreover, the edge  $\{xyxyxy, xyxyx\}$  can be rewritten as  $\{x, xy\}$ . Thus, we have found a circuit of length 6, with nodes  $1, y, yx, xyxy = xyx, xy, x$  whose edges are alternately labeled  $y$  and  $x$ . This circuit is, all by itself, a graph on which the group with presentation  $\langle x, y \mid x^2 = y^2 = (xy)^3 = 1 \rangle$  acts (regularly) as a group of automorphisms. Thus, we have found a realisation for this group. Apparently it has order 6 (the number of vertices) and is isomorphic to the symmetric group on 3 letters (which can be seen by verifying that the group is fully determined by its permutation behaviour on the three edges labeled  $x$ ).

Returning to  $W$ , we can throw in  $z$  and continue in much the same way. Cor Baayen is encouraged to try this. If the edges labeled  $x, y, z$  are drawn as

dotted lines, ordinary lines, fattened lines, respectively, the result is as depicted in Figure 1.

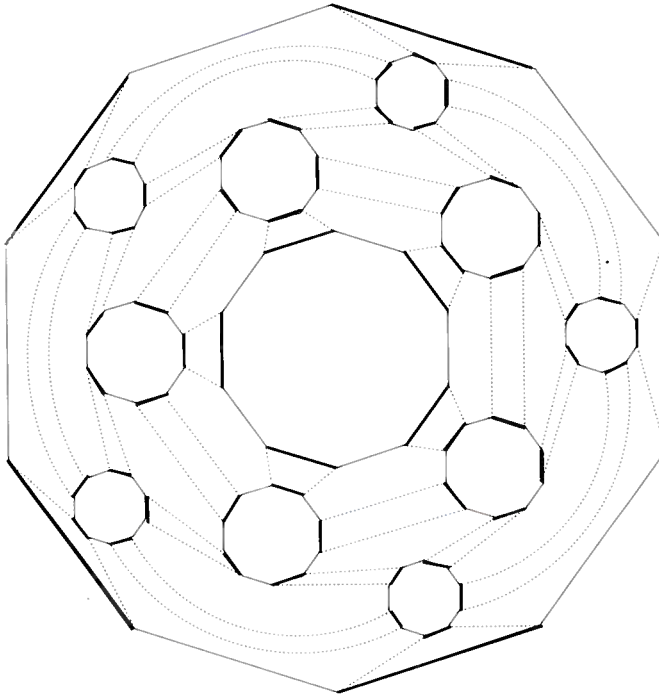


Figure 1. The Cayley graph of the icosahedral group

The number of vertices is 120, which is the order of the group  $W$ . In fact, the vertices of the graph can be identified with the elements of the group. In order to do so, select a vertex (which may be taken to be the starting point of the construction procedure that we just described) and identify it with the trivial element  $1$  of  $W$ . Next, associate any other vertex  $v$  with the element of  $W$  that can be found as follows: select a path from  $v$  to  $1$ , and write down the consecutive labels of the edges of a path from  $1$  to  $v$ . This produces a word expressing  $v$  as a product of the generators  $x, y, z$  of  $W$ .

So far, we have obtained a very geometric description of the abstractly defined icosahedral group. The reader may wonder how much of a miracle just happened. In general, that is, for arbitrary presentations by generators and relations, the technique we have carried out a special “icosahedral” case of, is known as the Todd-Coxeter coset enumeration method. The construction of the graph will not always be as straightforward as in the above example. The reason is that collapses of a more drastic nature than the identification of  $xyxy$  with  $x$  above may occur. It usually happens that a whole collection of

new vertices has to be created before a collapse is found to occur. In fact, presentations by generators and relations of the trivial group are known which only produce the graph on a single vertex after an enormous intermediate growth of (temporary) vertices.

An even bigger problem is that, especially when nothing is known a priori about the presentation of the group, termination is not even guaranteed. The single positive (but very powerful) result regarding coset enumeration is that, due to a result of Mendelsohn, cf. [Suz], it terminates if the resulting group is finite. (There is no a priori indicator known though as to how long it might take before termination takes place.)

The more general coset enumeration takes as input not only a group specified by generators and relations, but also a subgroup. The resulting vertices of the graph will then correspond to the cosets of the subgroup. Once a coset enumeration has been completed, a permutation representation for the group results. The upshot, for finite groups  $G$ , is great in that many good algorithms exist for determination of the structure of a permutation group (certainly when compared to the algorithms available for groups presented by generators and relations).

#### 4 How to find 3-dimensional representations

In this section, we show how using Gröbner basis methods, one can find 3-dimensional real (or complex) representations for the icosahedral group  $W$ . The construction will be similar to the one for the 2-dimensional quaternion group. Only this time the computations are done by use of a computer algebra package (for finding a Gröbner basis).

Thus, suppose  $\phi : W \rightarrow GL(\mathbf{R}^3)$  is a 3-dimensional representation of  $W$ . We assume that  $x$  and  $z$  are mapped to distinct elements in  $GL(\mathbf{R}^3)$ . Observe that, without loss of generality, we are in one of the following cases:

$$\text{I. } \phi(x) = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ and } \phi(z) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}; \text{ or}$$

$$\text{II. } \phi(x) = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ and } \phi(z) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}.$$

Since, for every representation  $\phi$ , there is also a representation  $\psi$  of  $W$  with  $\psi(u) = -\phi(u)$  for  $u$  equal to  $x$ ,  $y$  and  $z$ , we only have to consider representations  $\phi$  as in I. Let us do so. Then  $\phi(u)$  is a reflection for  $u$  equal to  $x$ ,  $y$  or  $z$ .

In order to extend  $\phi$  we need to find a matrix  $\phi(y) = (y_{i,j})_{1 \leq i,j \leq 3}$ .

Since  $\phi(y)$  is a reflection, its trace is 1. This gives us the following linear equation for the entries of  $y$ :

$$y_{1.1} + y_{2.2} + y_{3.3} = 1.$$

Similarly, as  $\phi(xy)$  is a real element of order 3 (it cannot be of order 1 because  $\phi(x)$  and  $\phi(z)$  are distinct), its trace must be 1. This gives another linear equation for the entries of  $y$ , namely

$$-y_{1,1} + y_{2,2} + y_{3,3} = 0.$$

The following is a Maple programme that creates the equations for the coefficients of  $\phi(y)$  that follow from the relations between the elements  $x$ ,  $y$  and  $z$  of  $W$ :

```
with(linalg):

#The three matrices we start out with:

x := matrix(3,3,[[ -1,0,0],[0,1,0],[0,0,1]]);
z := matrix(3,3,[[ 1,0,0],[0,1,0],[0,0,-1]]);
y := matrix(3,3,[[y11,y12,y13],[y21,y22,y23],[y31,y32,y33]]);

#putting the unknown in a list:

vars := [y11,y12,y21,y13,y31,y22,y23,y32,y33];

# Create the identity matrix of dimension n:

idmat := proc(n)
local ans,i,j;
ans := matrix(n,n);
for i to n do for j to n do ans[i,j] := 0;
if i=j then ans[i,j] := 1 fi od od:
evalm(ans)
end;

#use it to construct the 3-dimensional identity matrix:

idm := idmat(3);

# Given a matrix, derive the equations
# for its coefficients to be zero.

mkeq := proc(a)
local i,j,answ;
answ := {};

```

```

    for i to rowdim(a)
    do
        for j to coldim(a)
        do
            answ := answ union {a[i,j]}
        od
    od;
    answ
end;

```

# The relations for x, y and z imply the following equations:

```

y2 := evalm(evalm(y^2) - idm);
eqy := mkeq(y2);

xyx := evalm( x y x);
yxy:= evalm( y x y);
eqxy := mkeq(evalm(xyx - yxy));

zyzy := evalm( y z y z y);
zyyz := evalm( z y z y z);
eqyz := mkeq (evalm(zyzy -zyyz));

```

#loading the Groebner basis package:

```
with(grobner);
```

# the linear equations coming from the traces are

```
lineqs := {trace(evalm(y ) ) -1, trace(evalm(x y ) )};
```

# We do the Groebner basis computation in 3 steps.

# After each step one can simplify the equations by hand!

```

gby := gbasis(eqy union lineqs ,vars,plex);
gbxy := gbasis(eqxy union convert(gby, set),vars,plex);
gbxyz := gbasis(eqyz union convert(gbxy,set),vars,plex);

```

The Gröbner basis found by the computer algebra package has the following form:

$$\{2y_{11} - 1, y_{12} + 4y_{32}y_{13}y_{33} + 2y_{13}y_{32}, \\ 2y_{23}y_{31} + y_{21} + 4y_{31}y_{23}y_{33}, 2y_{13}y_{31} + y_{33} - 1, \\ 2y_{22} + 2y_{33} - 1, 4y_{23}y_{32} - 1, -1 + 4y_{33}^2 - 2y_{33}\}$$

From the “upper-triangular” structure of the Gröbner basis, the general shape of a solution up to algebraic conjugacy is readily seen to be

$$\phi(y) = \begin{bmatrix} \frac{1}{2} & \frac{-y_{32}}{2y_{31}} & -\frac{\sqrt{5}-3}{8y_{31}} \\ -\frac{y_{31}(\sqrt{5}+3)}{4y_{32}} & \frac{1-\sqrt{5}}{4} & \frac{1}{4y_{32}} \\ y_{31} & y_{32} & \frac{\sqrt{5}+1}{4} \end{bmatrix}$$

with  $y_{31}$ ,  $y_{32}$  both nonzero. In fact conjugation by suitable diagonal matrices shows that all solutions lead to equivalent representations (up to algebraic conjugacy, so in fact to two classes of representations).

By the way, using the same computer algebra package, checks can be easily carried out to verify that the solution  $\phi(y)$  indeed gives a linear representation.

In a subsequent section, we shall show that a 3-dimensional representation can easily be written down directly by applying the theory of Coxeter groups to  $W$ .

## 5 Representations in algebraic groups

As we have seen, faithful representations for a finite group  $G$  are embeddings of  $G$  in a group of the form  $GL(n, k)$ . This point of view raises the question whether we can determine all embeddings of such a group  $G$  in other linear algebraic groups. Algebraic groups can be viewed as subgroups of  $GL(n, k)$  stabilizing certain forms. For instance, the so-called symplectic groups are subgroups of even-dimensional linear groups stabilizing a non-degenerate bilinear alternating form. The crucial point is that such subgroups are algebraic subvarieties of  $GL(n, k)$  as they are zeros of the polynomial equations obtained by writing out for the entries of a matrix in  $GL(n, k)$  what it means to stabilize such a form (or more forms).

For the classical (infinite) series of algebraic groups, this viewpoint gives little news with respect to the usual representation theory, so naturally the attention is led to the exceptional types  $E_6, E_7, E_8, F_4, G_2$ . By use of the normal subgroup structure of a finite group, the problem can be reduced to three problems, the most salient of which concerns the study of embeddings of finite nonabelian simple groups in complex algebraic groups. Systematic searches for such embeddings received an impetus by *Kostant's conjecture*, formulated in 1983. It asserts that every simple complex algebraic group  $G(\mathbf{C})$  with a Coxeter number  $h$  such that  $2h + 1$  is a prime power, has a subgroup isomorphic to  $L(2, 2h + 1)$ . Here,  $L(2, q)$ , for  $q$  a prime power, stands for

the group of functions (so-called fractional linear transformations) of the form  $z \mapsto az + b/(cz + d)$  defined on the projective line of order  $q$ .

For  $G(\mathbf{C})$  of classical type, Kostant's conjecture is readily checked using ordinary representation theory and the Frobenius-Schur index. For  $G(\mathbf{C})$  of exceptional type the table below and the knowledge that  $h = 6, 12, 12, 18, 30$  for the five respective exceptional types give an affirmative case-by-case answer.

A quick overview of the state of the art is supplied by Table 1.

<b>Table 1.</b>	
Nonabelian simple groups $L$ a central extension of which embeds in a complex Lie group of exceptional type $X_n$	
$X_n$	$L$
$G_2$	$Alt_5, Alt_6, L(2, 7), L(2, 8), L(2, 13), U(3, 3)$
$F_4$	$Alt_7, Alt_8, Alt_9, L(2, 25), L(2, 27),$ $L(3, 3), {}^3D_4(2), U(4, 2), O(7, 2), O^+(8, 2)$
$E_6$	$Alt_{10}, Alt_{11}, L(2, 11), L(2, 17), L(2, 19),$ $L(3, 4), U(4, 3), {}^2F_4(2)', M_{11}, J_2$
$E_7$	$Alt_{12}, Alt_{13}, L(2, 29)^\?, L(2, 37), U(3, 8), M_{12}$
$E_8$	$Alt_{14}, Alt_{15}, Alt_{16}, Alt_{17}, L(2, 16), L(2, 31), L(2, 41)^\?,$ $L(2, 32)^\?, L(2, 49)^\?, L(2, 61), L(3, 5), Sp(4, 5), G_2(3), Sz(8)^\?$

There are two meanings to be attached to this table:

**Theorem.** *Let  $L$  be a finite simple group and let  $G$  be a simple algebraic group of exceptional type  $X_n$ .*

- (i) *If  $L$  occurs on a line corresponding to  $X_n$  in Table 1, then a central extension of it embeds in  $G(\mathbf{C})$ , with a possible exception for the five groups marked with a “?”.*
- (ii) *If  $X_n$  is as in some line of Table 1 and  $L$  appears neither in the line corresponding to  $X_n$  nor in a line above it, then no central extension of  $L$  embeds in  $G(\mathbf{C})$ .*

Here, to simplify the presentation,

- a. we have deliberately neglected questions of conjugacy classes of embeddings, and
- b. we have not specified the particular nonsplit central extensions of the simple groups involved.

During my years at CWI, I spent considerable time and effort realising some of the embeddings appearing in this table.

Ad a. An example where the conjugacy class question is more subtle than suggested by the table is provided by  $L(2, 13)$ . By [CW93], it is isomorphic to



a subgroup of  $F_4(\mathbf{C})$  whose normalizer is a finite maximal closed Lie subgroup of  $F_4(\mathbf{C})$ , whereas Table 1 only hints at the existence of embeddings via a closed Lie subgroup of  $F_4(\mathbf{C})$  of type  $G_2$ .

Ad b. For instance, the simple group  $L(2, 37)$  listed embeds into a group of type  $E_7$  but not in a group of type  $E_8$  because each embedding in an adjoint group of type  $E_7$  lifts to an embedding of  $SL(2, 37)$  into the universal covering group  $2 \cdot E_7(\mathbf{C})$ . Of course, the double cover  $SL(2, 37)$  of  $L(2, 37)$  embeds in the universal Lie group of type  $E_7$ , whence in a Lie group of type  $E_8$ .

Another warning concerning Table 1 is perhaps in order: The main theorems in [CW92] and [CoG] only concern subgroups not contained in closed Lie subgroups of positive dimension whereas Table 1 lists all finite simple subgroups (whether in a closed Lie subgroup of positive dimension or not).

- i. The choice of central extensions of simple groups rather than just simple groups is important because they are the ones needed for the generalized Fitting subgroup.
- ii. The table does not account for all groups that are involved in  $E_8(\mathbf{C})$ . For instance, no central extension of  $L(5, 2)$  is embeddable in  $E_8(\mathbf{C})$ , but a nonsplit extension  $2^{\{5+10\}} \cdot L(5, 2)$  does embed (cf. [A]).
- iii. The group  $L(2, 29)$  appears in a Lie group of type  $B_7$ , whence in one of type  $E_8$ . So, if the question whether a central cover of  $L(2, 29)$  embeds in  $E_7(\mathbf{C})$  has a negative answer, the group should appear at the bottom line of Table 1.
- iv. Unlike the  $GL(n, \cdot)$  case, knowledge of the classes of the individual elements of an embedded group  $L$  does not suffice to determine the conjugacy class of  $L$  in  $G$ . This has been observed by Borovik for the alternating group  $Alt_6$  in  $E_8(\mathbf{C})$ . The problem of how many conjugacy classes of embeddings of  $L$  exist only has a partial solution. See [Gr] for the full solution concerning  $G_2$ .
- v. The groups  $L(2, 41)$ ,  $L(2, 49)$  and  $Sz(8)$  do not appear as possible subgroups of  $E_8(\mathbf{C})$  in [CoG]; the arguments ruling them out given there are erroneous.
- vi. Another error in [loc. cit.] concerns the character given for  $L(2, 31)$ . The restriction of the adjoint character for  $E_8(\mathbf{C})$  to the subgroup isomorphic to  $L(2, 31)$  constructed by Serre (see below) has a different character.

One of the more spectacular results is the embedding of  $L(2, 61)$  in  $E_8(\mathbf{C})$ , the biggest of all five exceptional Lie groups. Using more refined versions of the techniques described in §§2, 3, Griess, Lisser and I have been able to prove that the suggested embedding exists and is unique up to conjugacy. In this case, the algebraic group can be seen as the subgroup of  $GL(248, \mathbf{C})$  stabilizing

a particular alternating trilinear form. Because our computations ran out of hand, we did all computations over a finite field ( $\mathbf{Z}/1831$ ) and argued that, if  $G$  embeds in a modular form of  $E_8$  over  $\mathbf{Z}/1831$ , it would also embed in  $E_8(\mathbf{C})$ . A key point in this argument was that  $G$  has order prime to 1831. This made it possible to deduce that any extension of  $G$  by a normal (profinite) subgroup of order a power of 1831, would split, that is, actually contain a subgroup isomorphic to  $G$ .

Very recently, Serre ([Se]) realised that this condition is not always needed. He started from a reasonable well-known embedding of  $L(2, 61)$  in  $E_8(61)$ . Then, the lifting technique gives a subgroup  $L$  of  $E_8(\mathbf{C})$  that has a normal profinite 61-subgroup  $N$  with quotient isomorphic to  $L(2, 61)$ . The important step is to show that, as an extension of  $L(2, 61)$  by  $N$ , the group  $L$  splits. For the  $L(2, 61)$  case, Serre needed a rather intricate argument; in the same sweep he also dealt with some other cases, like the embedding of  $L(2, 31)$  in  $E_8(\mathbf{C})$ , where the argument is rather succinct.

The algebraic group setting is also the right one for reconsidering the minimal splitting field question raised at the end of §3. Recall that, for the quaternion group, there is no unique minimal field realising an embedding in  $GL(2, k)$ . However, if we look at representations somewhat differently, it turns out that there does exist a minimal field for each conjugacy class of representations. To this end, we need to allow for all  $k$ -forms of  $GL(V)$ , that is all algebraic groups whose complex points form the group  $GL(2, \mathbf{C})$ .

In the above quaternion case, we have the following  $\mathbf{Q}$ -form of  $GL(2, \mathbf{C})$ :

$$H(k) = \{\alpha + \beta\mathbf{i} + \gamma\mathbf{j} + \delta\mathbf{k} \mid \alpha, \beta, \gamma, \delta \in k, \alpha^2 + \beta^2 + \gamma^2 + \delta^2 \neq 0\}.$$

This set forms a group, the basis elements of which multiply as the elements in  $Q$ . In particular,  $Q$  is a subgroup of  $H(\mathbf{Q})$ . To see that it is a  $\mathbf{Q}$ -form of  $GL(2, \mathbf{C})$ , consider the injective morphism  $H(\mathbf{Q}) \rightarrow GL(2, \mathbf{Q}(i))$ :

$$\alpha + \beta\mathbf{i} + \gamma\mathbf{j} + \delta\mathbf{k} \mapsto \begin{pmatrix} \alpha + \beta i & \gamma + \delta i \\ -\gamma + \delta i & \alpha - \beta i \end{pmatrix}.$$

When extended to  $\mathbf{Q}(i)$ , and so certainly, when extended to  $\mathbf{C}$ , this maps becomes an isomorphism.

Thus, we have obtained a unique minimal field  $k$ , namely  $\mathbf{Q}$ , for which there exists a  $k$ -form of  $GL(V)$  containing  $Q$ . This illustrates a result due to Springer [Spr] that for each group morphism  $\rho : G \rightarrow H(\mathbf{C})$  from  $G$  to an algebraic group  $H(\cdot)$ , there is a minimal field extension  $k$  of  $\mathbf{Q}$  such that  $G$  embeds into a  $k$ -form of  $H$ .

Coming back to this problem for the subgroup  $L(2, 61)$ , the minimal splitting field is probably  $Q(\sqrt{61})$ ; but, to the best of my knowledge, this has not yet been established. The next question is then, if  $k$  is the minimal splitting field, which  $k$ -form is it that the subgroup embeds in? The various  $\mathbf{Q}(\sqrt{61})$ -forms of  $E_8(\mathbf{C})$  are known by Cernousov's work (there are 9).

Together with Tiep ([CT]), I have found the minimal splitting fields for some other remarkable subgroups of the exceptional algebraic groups, namely the Jordan subgroups.

## 6 The reflection representation

As promised earlier, we now come to another way of constructing a 3-dimensional representation for the icosahedral group  $W$ . Tits has shown that, for the so-called Coxeter groups, one can always find a faithful “reflection representation.” The icosahedral group is a Coxeter group, whose reflection representation is equivalent to the one found above.

We shall describe the construction of the reflection representation of the icosahedral group, thereby following the general construction for Coxeter groups. Put  $\sigma = \zeta^2 + \zeta^3$  and  $\tau = \zeta + \zeta^4$ , where  $\zeta = e^{2\pi i/5}$ .

Starting point is a 3-dimensional space  $V$  (one dimension for each generator of  $W$ ), supplied with the symmetric bilinear form given by the following matrix:

$$\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & \sigma \\ 0 & \sigma & 2 \end{pmatrix}.$$

Note that, if the rows and columns are labeled with  $x, y$  and  $z$ , respectively, the off-diagonal entries are  $-2 \cos(\pi/m)$ , where  $m$  is the order of the product of the generators corresponding to row and to column. (This hints toward the general case for those who know what a Coxeter group is.) Denote the bilinear form by  $(\cdot, \cdot)$ . It is positive-definite, so the 3-dimensional space, supplied with this form is Euclidean. Now, for  $\alpha \in V$  with  $(\alpha, \alpha) = 2$ , the reflection with “root”  $\alpha$  is given by

$$s_\alpha : w \mapsto w - (w, \alpha)\alpha.$$

The reflection representation is determined by the images of  $x, y, z$ . These images will be the reflections  $s_\alpha$  for  $\alpha$  the standard basis vectors:  $\alpha = e_1, e_2, e_3$ . These roots are called the *fundamental roots* of  $W$ . Thus, we obtain the following matrices:

$$\begin{aligned} x &= \begin{pmatrix} -1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \\ y &= \begin{pmatrix} 1 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & -\sigma & 1 \end{pmatrix}, \\ z &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -\sigma \\ 0 & 0 & -1 \end{pmatrix}. \end{aligned}$$

By what we have seen above, this representation must be equivalent to one of the two (algebraically conjugate ones) constructed using Gröbner bases in §5.

Using the faithfulness of the reflection representation, it is easy to derive that the icosahedral group  $W$  is finite and to find a permutation representation. For instance, consider the set  $\Phi$  of all roots of reflections in  $W$ . This set can be built up from the fundamental roots. Up to signs, they are:

$$\begin{array}{lll}
 (1, 0, 0) & (0, 1, 0) & (0, 0, 1) \\
 (1, 1, 0) & (0, 1, -\sigma) & (0, -\sigma, 1) \\
 (1, 1, -\sigma) & (0, -\sigma, -\sigma) & (-\sigma, -\sigma, 1) \\
 (1, -\tau - 2\sigma, -\sigma) & (-\sigma, -\sigma, -\sigma) & (1, -\tau - 2\sigma, -\tau - 2\sigma) \\
 (-\sigma, -\tau - 2\sigma, -\sigma) & (-\sigma, -\tau - 2\sigma, -\tau - 2\sigma) & (-\sigma, -2\sigma, -\tau - 2\sigma)
 \end{array}$$

Thus, we have a set  $\Phi$  of  $2 \times 15 = 30$  roots. Clearly, if  $\alpha \in \Phi$ , then also  $-\alpha = s_\alpha \alpha \in \Phi$ . If the 15 pairs  $\pm\alpha$  are numbered according to their occurrence, the generators  $x$ ,  $y$  and  $z$  induce the following permutations:

$$x = (2, 4)(5, 7)(6, 9)(8, 11)(10, 13)(12, 14),$$

$$y = (1, 4)(3, 6)(5, 8)(7, 10)(11, 13)(14, 15),$$

$$z = (2, 5)(4, 7)(6, 8)(9, 11)(10, 12)(13, 14).$$

The kernel of this permutation representation is readily seen to be  $\{\pm I_2\}$ . Since only a finite number of roots are being permuted, and the reflection representation of  $W$  is faithful, we see again that  $W$  is finite.

A remarkable property, true of arbitrary Coxeter groups, is that one of  $\pm\alpha$  has all coefficients with respect to the fundamental root basis non-negative. These roots are called the positive roots. The set of all positive roots is denoted by  $\Phi^+$ , so that  $\Phi = \Phi^+ \cup \Phi^-$ , where  $\Phi^- = -\Phi^+$ . In Figure 2 we have pictured  $\Phi^+$  and the way it is built up using the generators  $x$ ,  $y$ ,  $z$ , with the same conventions as for Figure 1 regarding the edges. The dashed line at the bottom indicates where the action of the generators crosses over to negative roots.

## 7 Presentation by generators and relations

We now go back to presentations of groups by means of generators and relations. For the icosahedral group  $W$  we have already given such a presentation:  $W = \langle X \mid Y \rangle$ , with  $X = \{x, y, z\}$  and

$$Y = \{x^2, y^2, z^2, (xy)^3, (yz)^5, (xz)^2\}.$$

Of course, for a given group, such a presentation is far from unique.

Computations using the presentation of a group by generators and relations are based on the idea that it is easy to present a free group over a given alphabet  $X$ . Or maybe, even simpler, start with the free monoid  $X^*$  over  $X$ . This is the set of all strings (also called words) we can form with the symbols (also called letters) from  $X$ . Such a monoid has the great advantage that every element corresponds to a unique expression for it.

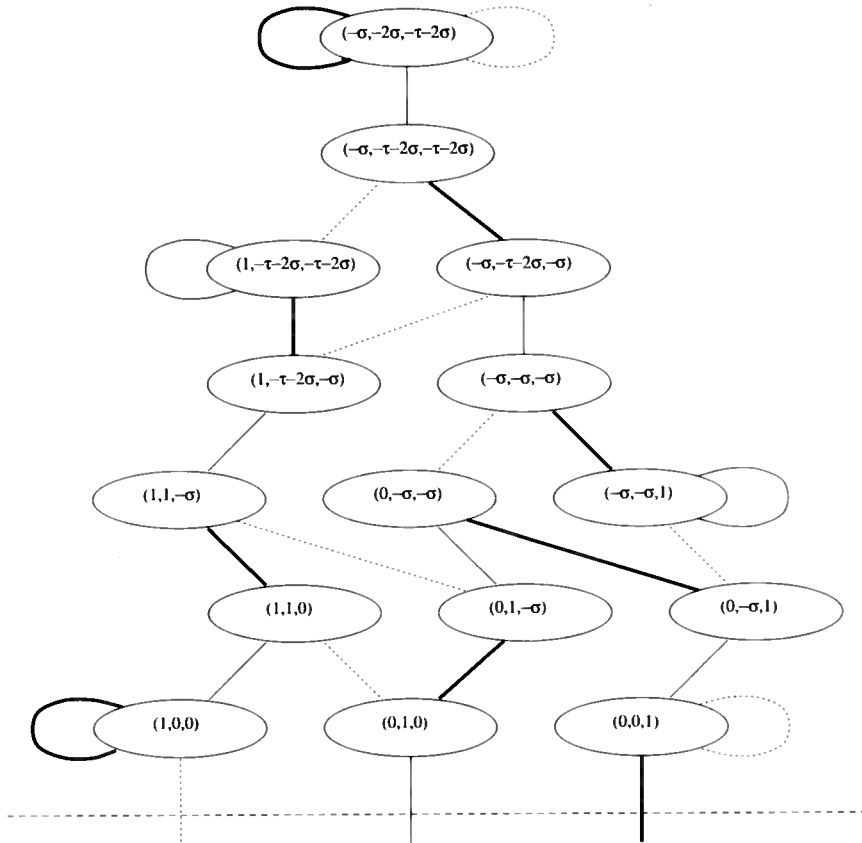


Figure 2. The positive roots with action of  $W$

This phenomenon is no longer true of the free group on  $X$ . We can define it as a quotient of the free monoid on

$$A = X \cup X^{-1} = \{x, x^{-1}, y, y^{-1}, z, z^{-1}\}$$

with respect to the relations

$$xx^{-1} = x^{-1}x = 1$$

$$yy^{-1} = y^{-1}y = 1$$

$$zz^{-1} = z^{-1}z = 1.$$

Although now it is no longer true that every element of the free group on  $X$  corresponds to a unique word in  $A^*$ , we still have a very good way of handling this: the group elements correspond bijectively to the reduced words in the monoid, i.e., those with no occurrences of

$$xx^{-1}, x^{-1}x, yy^{-1}, y^{-1}y, zz^{-1}, z^{-1}z.$$

The icosahedral group  $W$  is obtained as a quotient by dividing out with respect to the normal subgroup generated by the relators

$$x^2, y^2, z^2, (xy)^3, (yz)^5, (xz)^2.$$

The question now arises how to find a set of words  $\widetilde{W}$  in  $A^*$  such that every element of  $W$  corresponds to a unique element of  $\widetilde{W}$ . Another way of saying this is that we want to find a section  $\sigma$  of the natural map  $\phi : A^* \rightarrow W$ . The set  $\widetilde{W}$  is then the image of  $\sigma$ . Yet another way of expressing the wish for unique representatives in  $A^*$  of elements of  $W$  is more algorithmic: for each element  $w \in A^*$ , we want to be able to find a “canonical” element in the fibre  $\phi^{-1}(\phi(w))$ .

A very successful approach is based on rewriting techniques. It uses a total well-ordering on  $A^*$ . More precisely, a well-founded total ordering  $<$  is called a *reduction ordering* if

$$\begin{aligned} \forall l, r, m_1, m_2 \in A^* \\ m_1 < m_2 \Rightarrow lm_1r < lm_2r \end{aligned}$$

and  $1 = \min A^*$ . We need a reduction ordering  $<$  on  $A^*$ . There are plenty such orderings, but we will content ourselves with the total degree lexicographic one, that is the one where  $v < w$  if either the length of  $v$  (as a string of symbols from  $A$ ) is less than the length of  $w$ , or these lengths are equal and  $v$  comes prior to  $w$  in the usual lexicographic ordering (where  $x < y < z < x^{-1} < y^{-1} < z^{-1}$ ). Thus,

$$1 < x < y < z < xx < xy < xz < yx < yy < \dots$$

The canonical element for an arbitrary  $m \in A^*$  can then be taken to be

$$\min \phi^{-1}\phi(m).$$

Now the purpose is to rewrite an arbitrary word  $m \in A^*$  to the canonical word  $\min \phi^{-1}\phi(m)$  by stepwise finding smaller representatives of  $\phi(m)$ . First of all, for involutions such as the generators in  $X$  for  $W$ , we may rid ourselves of inverses by use of the rewriting rules

$$x^{-1} \Rightarrow x, \quad y^{-1} \Rightarrow y, \quad z^{-1} \Rightarrow z.$$

For  $W$  with the above presentation, the obvious rewriting rules

$$\begin{aligned} xx &\Rightarrow 1, \quad yy \Rightarrow 1, \quad zz \Rightarrow 1, \\ zx &\Rightarrow xz, \\ yxy &\Rightarrow xyx, \\ zyzyz &\Rightarrow yzyzy. \end{aligned}$$

do not suffice. For instance,  $(xyz)^{10}$  cannot be reduced to the trivial element; but, for instance, writing out the permutation of the roots corresponding to  $xyz$  (by use of the permutations given for  $x$ ,  $y$  and  $z$  in §6), we find cycles of length 5 only, so the fifth power is in the kernel of the permutation representation, whence  $(xyz)^{10} = 1$ .

## 8 A rewriting system for the icosahedral group

Recent techniques for Coxeter groups have given insight in how to produce a proper set of rewriting rules. By “proper” we mean what is usually called “confluent”; it has the effect that each input word can be successfully rewritten to the corresponding canonical word by use of the rewriting rules. Rather than presenting the rewriting rules explicitly, we give an algorithm for rewriting an input word  $\mathbf{w} \in X^*$ , where  $X = \{x, y, z\}$  to the corresponding canonical element in  $\widetilde{W}$ . The present treatment comes from [BH], with a variation due to DuCloux and Casselman.

Consider the set  $\Phi^+$  of 15 positive roots of  $W$  again. The algorithm works with induction. Let us assume that, for  $w = r_1 \cdots r_{k-1} r_k \cdots r_q$ , where  $r_i \in X$  for  $i = 1, \dots, q$ , we have already established that  $r_1 \cdots r_{k-1}$  is in canonical form.

Then, for  $i = k - 1, k - 2, \dots$  we consider the action of  $r_i \cdots r_{k-1}$  on the fundamental root  $\alpha \in \Pi = \{e_1, e_2, e_3\}$  corresponding to  $r_k$ . That is, we subsequently compute  $r_{k-1}\alpha$ ,  $r_{k-2}r_{k-1}\alpha$ , and so on, until we reach a fundamental root again.

Say this happens the first time for  $i \in \{1, \dots, k - 1\}$  and fundamental root  $\beta$ :

$$r_i r_{i+1} \cdots r_{k-1} \alpha = \beta.$$

Write  $s = s_\beta$ . Then, since  $s_\gamma r_\gamma = g s_\gamma g^{-1}$  for any  $\gamma \in \Phi$  and  $g \in GL(\mathbf{R}^3)$ , we have

$$r_i r_{i+1} \cdots r_{k-1} r_k = s r_i r_{i+1} \cdots r_{k-1}.$$

If the right hand side represents a (lexicographically) smaller word, we substitute it for  $r_i r_{i+1} \cdots r_{k-1} r_k$  and continue determining the canonical word for the first part  $r_1 \cdots r_{i-1} s$  of  $\mathbf{w}$ . Otherwise, we leave things as they are ... except that we do not want to move to negative roots. This can only happen, if a fundamental root  $e_j$  occurs to which the corresponding reflection is applied (sending it to  $-e_j$ ). This remarkable property is clearly visible from Figure 2, where only three edges make a root sink through the bottom line.

For further details, it is useful to write  $\alpha_r$  for the positive root corresponding to a reflection  $r$  of  $W$ . Recall  $\Pi = \{e_1, e_2, e_3\}$ . Here is a full description of the canonical word algorithm:

**At initialization:**  $\mathbf{w} = [r_1, \dots, r_q] \in X^*$ , representing  $w = r_1 \cdots r_q \in W$ ; and an index  $k := 1$ .

**At termination:**  $\mathbf{w}$  is the canonical word for  $w$ .

**Invariants:**  $w \in W$  will be fixed throughout, and  $\mathbf{w}$  will always be an expression for  $w$ . The first part of length  $k - 1$  of  $\mathbf{w}$  is in canonical form.

```

while  $k \leq \ell(\mathbf{w})$  do
   $i := k - 1$ ;  $\alpha := \alpha_{r_k}$ ;
  while  $i > 0$  do
     $\alpha := r_i \alpha$ ;

```



```

case  $\alpha \in \Phi^-$ :
   $\mathbf{w} := [r_1, \dots, r_{i-1}, r_{i+2}, \dots, r_q]$ ;
   $k := k - 1$ ;  $i := i - 1$ ;
case  $\alpha \in \Pi$ :
  if  $[r_i, r_{i+1}, \dots, r_k] > [s_\alpha, r_i, \dots, r_{k-1}]$ 
  then  $\mathbf{w} := [s_\alpha, r_j, \dots, r_{k-1}]$ ;
  fi;
   $k := i$ ;  $i := k - 1$ ;
otherwise:  $i := i - 1$ ;
od;
 $k := k + 1$ ;
od

```

For given  $i$  and  $k$ , the root  $\alpha = r_i \cdots r_{k-1} \alpha_{r_k}$  is being considered. In case  $\alpha \in \Phi^-$ , we must have  $i = k - 1$ ; a fundamental root is reached by its corresponding reflection, we have  $r_i = r_{i+1} = r_k$  and we can reduce length.

If a positive fundamental root  $\alpha = r_i r_{i+1} \cdots r_{k-1} \alpha_{r_k}$  is hit, then we have seen above that the new expression generated by the algorithm represents the same element of  $W$ .

It may seem to be a computational difficulty that the root system is needed. But, in fact, the full action, in terms of images of roots under fundamental reflections, has already been stored in Figure 2. The roots there are pictured with respect to “depth”: the number of fundamental reflections needed to turn them into negative roots: the fundamental roots have depth 1, the next layer up consists of  $(1, 1, 0)$ ,  $(0, 1, -\sigma)$  and  $(0, -\sigma, 1)$  (of depth 2), and so on, until we reach the unique one of depth 7:  $(-\sigma, -2\sigma, -\tau - 2\sigma)$ .

A new rewriting rule that we obtain by applying the algorithm to the left hand side is  $zyzyxz \Rightarrow yzyzyx$ . Cor Baayen is encouraged to try and prove that  $(xyz)^{10} = 1$  using the algorithm. (Hint: the rewriting rule  $(yxz)^5 \Rightarrow (xzy)^5$  is crucial.)

We have seen that the positive root system, with its “depth” structure, and, above all, its  $W$ -action, is an excellent automaton for the “icosahedral” word problem. For a finite group like  $W$ , it may not be much of a surprise that we can find a solution to the word problem. The surprise however is that the technique described works for all Coxeter groups, including the infinite ones, once a little variation has been made that we shall now describe.

If we take  $W$  to be an arbitrary Coxeter group, the same algorithm may work again, but then the set of all positive roots may be infinite and so cannot be fully constructed in advance. The merit of Brink and Howlett is that they showed that in that case one can “truncate” the root system, and work with a finite part only. It runs as follows: define, for  $\alpha$  and  $\beta$  positive roots,  $\alpha \succ \beta$  if  $(\alpha, \beta) \geq 1$  and  $\alpha - \beta$  has non-negative coefficients (when written as a linear combination of fundamental roots). We then say that  $\alpha$  dominates  $\beta$ . The domination relation is a partial ordering with (and this is the non-trivial result:) finitely many minimal roots. The “automaton” can then be restricted to the minimal roots, and a single additional element, denoted by  $*$ , replacing all non-minimal

elements. Whenever a minimal root is mapped onto a non-minimal root, the acceptance state  $*$  is reached: this means that the word that is being rewritten is canonical (in the inner loop of the above algorithm), so that one can move up to the next value of  $k$ , without having to process the word any further to the left, lowering the parameter  $i$ ).

## 9 Synthesis of Todd-Coxeter and Buchberger

It is well known that Buchberger's Gröbner basis algorithm can be seen as a particular case of the Knuth-Bendix procedure, in which confluent rewriting is guaranteed due to the successful completion in the context of polynomial algebras. More and more, I am convinced that the classical Todd-Coxeter coset enumeration procedure can also be seen as such. In particular, the success here is guaranteed by Mendelssohn's result described in §3. This is a line of research that I have only recently started to pursue, and I will only vaguely indicate what I have in mind.

Given a monoid  $M$  and a field  $k$ , we can define the monoid algebra  $k\langle M \rangle$  (if  $M$  is a group, this comes down to the group algebra).

We study quotients of  $k\langle M \rangle$  with respect to ideals  $I$ . Again a reduction ordering  $<$  on  $M$  is useful. Not every monoid affords a reduction ordering, but the most important examples, the free monoid and the free abelian monoid (in which case  $M$  is a polynomial algebra!) on a finite alphabet do.

For

$$f = \sum_{m \in M} f_m m \in k\langle M \rangle,$$

with  $f_m \in k$  (finitely many nonzero), we set

$$lt(f) = \max\{m \in M \mid f_m \neq 0\}.$$

Moreover, for any subset  $X$  of  $k\langle M \rangle$ , set:

$$M(X) = \{lt(f) \mid f \in X\} \quad \text{and} \quad O(X) = M \setminus M(X).$$

**Theorem.** *Let  $M$  be a monoid with a reduction ordering  $<$ , and suppose  $I$  is an ideal in  $k\langle M \rangle$ . Then the following statements hold.*

- (i)  $k\langle M \rangle = I \oplus k \cdot O(I)$ .
- (ii)  $k\langle M \rangle / I \cong k \cdot O(I)$  as vector spaces over  $k$ .
- (iii)  $\forall f \in k\langle M \rangle \exists! g \in k \cdot O(I) : f - g \in I$ .

In this setting, we write  $g := Can(f, I)$ , and refer to it as the canonical element corresponding to  $f$ . Observe that

$$Can(f, I) = Can(g, I) \Leftrightarrow f - g \in I;$$

A subset  $G$  of  $I$  is called a Gröbner basis if  $(M(G)) = M(I)$ , where  $(N)$ , for a subset of  $M$ , denotes the semigroup ideal generated by  $N$  in  $M$ .

This approach can be found in [M].

**Proposition.** *Let  $M$  be finitely generated (Noetherian) and supplied with a reduction ordering. For each ideal  $I$  of  $k\langle M \rangle$ , there is a unique subset  $B$  of  $I$  satisfying:*

- (i)  $M(B)$  is a minimal generating set of  $M(I)$ ;
- (ii) the coefficient of  $lt(b)$  in  $b$  is 1 for each  $b \in B$ ;
- (iii)  $b = lt(b) - Can(lt(b), I)$  for each  $b \in B$ .

This set  $G$  is the so-called reduced Gröbner basis of  $I$ . The polynomial case occurs for  $M = \mathbf{N}^n$ . Then the already classical Buchberger algorithm finds a Gröbner basis for  $M = \mathbf{N}^n$ .

Thus, quotients of polynomial rings can be determined algorithmically. But this is inconceivable for the general case, since the word problem for groups is known to be unsolvable.

To see the connection with group presentations, start with a finitely presented group  $G = \langle X \mid Y \rangle$ . Take  $M$  to be the free monoid generated by  $A = X \cup X^{-1}$  and total degree lexicographic ordering  $<$  such that  $x < y^{-1}$  for all  $x, y \in X$ . Now let  $I$  be the ideal of all  $v - w \in k\langle M \rangle$  with  $v, w \in M$  such that  $vw^{-1} \in Y$ . Here, we assume that  $xx^{-1}$  and  $x^{-1}x$  are relators (i.e., belong to  $Y$ ). Then  $k\langle M \rangle/I$  is the group algebra of  $G$  over  $k$ . The set  $O(I)$  of the above theorem coincides with the collection  $\tilde{G}$  of words in  $A^*$  which are minimal in the inverse image under  $A^* \rightarrow G$  of an element in  $G$ .

It is a very useful fact that binomials are transformed into binomials under all operations involved in the Knuth-Bendix procedure, and also under the transformations obtained from a translation of the Todd-Coxeter enumeration to this setting. If a Gröbner basis is found for the ideal  $I$ , then, by the above proposition, and the “binomial invariance,” a solution to the word problem for  $G$  has been found.

Let us return once more to the icosahedral group  $W$ . The algorithm of §8 uses only finitely many rewriting rules; they can be read off from Figure 2. A simple example is  $[y, x, y] \Rightarrow [x, y, x]$ , which corresponds to the element  $yxxy - xyx \in k\langle A \rangle$ . The collection of all rules thus obtained, together with  $x - x^{-1}$ ,  $xx - 1$ ,  $y - y^{-1}$ ,  $yy - 1$ ,  $z - z^{-1}$ ,  $zz - 1$  will lead to a Gröbner basis for the ideal  $I$ , thus presenting a model to compute with the group algebra  $k[W]$  in terms of  $kO(I)$ .

As remarked at the end of §8, such results are (at least theoretically) no surprise for finite groups like  $W$  (although the automaton is efficient). But, due to the results of Brink and Howlett, we have similar Gröbner bases for arbitrary (infinite) Coxeter groups.

## 10 Acknowledgments

I am grateful to Hans Sterk and Remko Riebeek for reading a preliminary version of this paper. I am greatly indebted to Hans Cuyppers for making the figures.

Part of §4 is from joint preparations of a course with Hans Cuypers and Remko Riebeek. The bulk of §5 is from [CW94].

#### References

- [A] A.V. Alekseevskii 1974, *Finite commutative Jordan subgroups of complex simple Lie groups*, *Funct. Anal. and its Appl.* 8 (1974), 277 – 279.
- [BCN] A.E. Brouwer, A.M. Cohen, A. Neumaier, *Distance-Regular Graphs*, Springer-Verlag, Berlin, 1989.
- [BH] B. Brink and R. Howlett, *A finiteness property and an automatic structure for Coxeter groups*, *Math. Ann.* 296 (1993), 179 – 190.
- [CoG] A.M. Cohen and R.L. Griess, Jr., *On finite simple subgroups of the complex Lie group of type  $E_8$* , *Proc. Sympos. Pure Math.* 47 (1987), Pt. 2, 367 – 405.
- [CT] A.M. Cohen, P.H. Tiep, *Integral forms for Jordan subgroups*, preliminary preprint, Eindhoven, 1994.
- [CW92] A. M. Cohen and D. B. Wales, *On finite subgroups of  $E_6(C)$  and  $F_4(C)$* , preprint, Eindhoven, 1992.
- [CW93] A. M. Cohen and D. B. Wales, *Embedding of the group  $L(2, 13)$  in the groups of Lie type  $E_6$* , *Israel J. Math.* 82 (1993), 45 – 86.
- [CW94] A. M. Cohen and D. B. Wales, *Finite simple subgroups of semisimple complex Lie groups – a survey*, Como 1993 Proceedings.
- [Gr] R.L. Griess, Jr., *Basic conjugacy theorems for  $G_2$* , report, Ann Arbor, University of Michigan, 1994.
- [K] F. Klein, *Vorlesungen über da Ikosaeder und die Auflösung der Gleichungen vom fünften Grade*, Teubner, Leipzig, 1884.
- [M] T. Mora, *An introduction to commutative and non-commutative Gröbner bases*, preprint, Genua, 1994.
- [Se] J-P. Serre, *Les sous-groupes  $PSL(2, p)$* , preliminary report, Paris, 1994.
- [Spr] T.A. Springer, Private communication, Utrecht, 1994.
- [Suz] M. Suzuki, *Group Theory I*, Chapter 2.5, Springer, Berlin, 1982.



# Research in Computational Fluid Dynamics, Stimulated by ERCIM

Jean-Antoine Désidéri  
*INRIA Sophia-Antipolis*

Pieter W. Hemker  
Barry Koren  
*CWI*

Marie-Hélène Lallemand  
*INRIA Rocquencourt*

## 1 INTRODUCTION

During the last six years, by his spoken and written words in support of ERCIM<sup>1 2 3 4</sup>, Prof. Baayen has stimulated cooperation between fellow researchers at European sister institutes. Baayen's words were not just pie in the sky. Already in 1988, financial and logistic support became available for mutual working visits of ERCIM researchers. Based on earlier contacts, the authors could soon take advantage of these newly created ERCIM opportunities. Without encountering any red-tape, the authors could start joint ERCIM work in the field of computational fluid dynamics. This led to several papers in the international scientific literature. This contribution gives a survey of some of this research, which is described in more detail in two SIAM articles<sup>5 6</sup>. It is our tribute to Prof. Baayen's inspiring role as the first ERCIM president.

---

<sup>1</sup>P.C. Baayen, A. Bensoussan, G. Seegmüller, European computer science market, *CWI GMD INRIA Newsletter*, **1**, p. 1, 1989.

<sup>2</sup>P.C. Baayen, ERCIM's joint action programme is taking shape, *CWI GMD INRIA Newsletter*, **3**, p. 1, 1990.

<sup>3</sup>P.C. Baayen, Strengthening ERCIM, *ERCIM News*, **11**, p. 2, 1992.

<sup>4</sup>P.C. Baayen, Editorial, *ERCIM News*, **15**, p. 1, 1993.

<sup>5</sup>M.-H. Lallemand and B. Koren, Iterative defect correction and multigrid accelerated explicit time stepping schemes for the steady Euler equations, *SIAM Journal on Scientific Computing*, **14**, p. 953-970, 1993.

<sup>6</sup>J.-A. Désidéri and P.W. Hemker, Analysis of the convergence of iterative implicit and defect correction algorithms for hyperbolic problems, *SIAM Journal on Scientific Computing* (to appear, Jan. 1995).

## 2 ITERATIVE DEFECT CORRECTION AND MULTIGRID ACCELERATED EXPLICIT TIME STEPPING FOR THE STEADY EULER EQUATIONS

Convergence results are presented for a new pseudo-unsteady solution method for higher-order accurate upwind discretisations of the steady Euler equations. Comparisons are made with an existing pseudo-unsteady solution method. Both methods make use of nonlinear multigrid for acceleration and nested iteration for the fine-grid initialisation. The new method uses iterative defect correction (ItDeC). This section is based on the paper [9].

### 2.1 Equations

The equations considered are the steady, two-dimensional, compressible Euler equations

$$\frac{\partial F(W)}{\partial x} + \frac{\partial G(W)}{\partial y} = 0, \quad (2.1)$$

where

$$W = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{pmatrix}, \quad (2.2a)$$

$$F(W) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho u(e + \frac{p}{\rho}) \end{pmatrix}, \quad G(W) = \begin{pmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho v(e + \frac{p}{\rho}) \end{pmatrix}. \quad (2.2b)$$

Assuming a perfect gas, the total energy  $e$  satisfies:  $e = \frac{1}{\gamma-1} \frac{p}{\rho} + \frac{1}{2}(u^2 + v^2)$ . The ratio of specific heats  $\gamma$  is assumed to be constant.

### 2.2 Spatial discretisation

The computational grid is obtained by a hybrid finite element - finite volume partition. A (possibly unstructured) finite-element triangularisation is used as the basic partition. A cell-centered finite-volume partition is derived from the finite-element partition by connecting the centers of the triangle sides in the manner illustrated in Figure 1.1. The finite-volume grid gives us the easy possibility of grouping together the nodes associated with contiguous finite volumes. If we take unions of control volumes this results in a new coarser mesh. Repetition of this operation gives coarser and coarser meshes. For details about the coarsening process (multilevel gridding) we refer to [8].

On the finest grid, for all finite volumes  $C_i$ ,  $i = 1, 2, \dots, N$ , we consider the integral form

$$\oint_{\partial C_i} (F(W)n_x + G(W)n_y) ds = 0, \quad i = 1, 2, \dots, N, \quad (2.3)$$



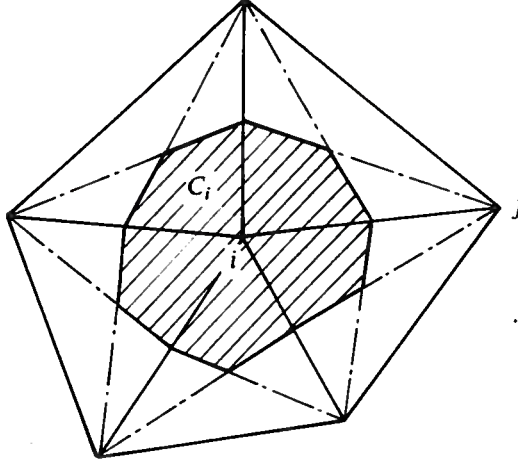


FIGURE 2.1. Finite volume  $C_i$

with  $n_x$  and  $n_y$  the  $x$ - and  $y$ -component of the outward unit normal on the volume boundary  $\partial C_i$ . For the Euler equations, because of their rotational invariance, (1.3) may be rewritten as

$$\oint_{\partial C_i} T^{-1}(n_x, n_y) F(T(n_x, n_y) W) ds = 0, \quad i = 1, 2, \dots, N, \quad (2.4)$$

where  $T(n_x, n_y)$  is the rotation matrix

$$T(n_x, n_y) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & n_x & n_y & 0 \\ 0 & -n_y & n_x & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.5)$$

For simplicity, we assume the flux to be constant across each bi-segment  $\partial C_{ij}$  of the boundary  $\partial C_i$ , where  $\partial C_{ij} = \partial C_i \cap \partial C_j$  is the common boundary between the neighbouring volumes  $C_i$  and  $C_j$  (Figure 1.2a). Hence,  $\partial C_i = \cup \partial C_{ij}$ ,  $j = 1, 2, \dots, n_i$ , with  $n_i$  the number of neighbouring volumes  $C_j$ . (In the example of Figure 1.1:  $n_i = 5$ .) Since we have assumed that the flux is constant along  $\partial C_{ij}$ , it is equal to the flux across the straight segment  $\bar{\partial} C_{ij}$  connecting the two extreme points of  $\partial C_{ij}$  (Figure 1.2b). If we introduce the outward unit normal  $\bar{n}_{ij} = ((\bar{n}_x)_{ij}, (\bar{n}_y)_{ij})^T$  along each  $\bar{\partial} C_{ij}$ ,  $j = 1, 2, \dots, n_i$ , with the assumption of a constant flux, the contour integral (1.4) can be rewritten as the sum

$$\sum_{j=1}^{n_i} \bar{T}_{ij}^{-1} F(\bar{T}_{ij} W_{ij}) l_{ij} = 0, \quad i = 1, 2, \dots, N, \quad (2.6)$$

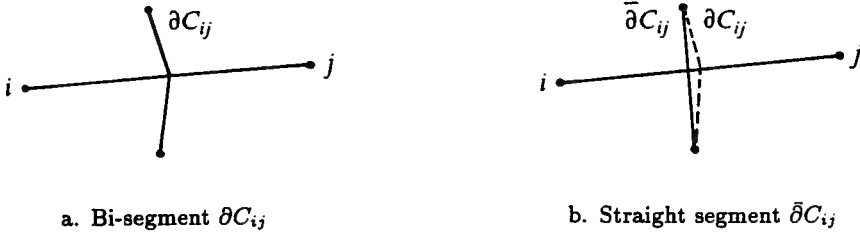


FIGURE 2.2. Segments in between finite volumes  $C_i$  and  $C_j$

where  $\bar{T}_{ij} = T((\bar{n}_x)_{ij}, (\bar{n}_y)_{ij})$ , where  $W_{ij}$  is some value of  $W$  depending on for instance  $W_i$  and  $W_j$ , and where  $l_{ij}$  is the length of the segment  $\partial C_{ij}$ .

Crucial in (1.6) is the way in which the cell-face flux  $F(\bar{T}_{ij}W_{ij})$  is evaluated. For this we use an upwind scheme which follows the Godunov principle [3], which assumes that the constant flux vector along each segment  $\partial C_{ij}$  is determined only by a uniformly constant left and right cell-face state ( $W_{ij}^l$  and  $W_{ij}^r$ ). The 1D Riemann problem which then arises at each cell face is solved in an approximate way. With this, (1.6) can be further rewritten as

$$\sum_{j=1}^{n_i} \bar{T}_{ij}^{-1} \Phi(\bar{T}_{ij}W_{ij}^l, \bar{T}_{ij}W_{ij}^r) l_{ij} = 0, \quad i = 1, 2, \dots, N, \quad (2.7)$$

where  $\Phi$  denotes the approximate Riemann solver. Several approximate Riemann solvers exist. In the present paper we apply that of Osher and Solomon [11].

The flux evaluation, and so the space discretisation, may be either first- or higher-order accurate. First-order accuracy is obtained in the standard way; at each finite-volume wall, the left and right cell-face state which have to be inserted in the numerical flux function are taken equal to those in the corresponding adjacent volumes:

$$W_{ij}^l = W_i, \quad W_{ij}^r = W_j. \quad (2.8)$$

Whereas the first-order accurate discretisation is applied at all levels, the higher-order discretisation is applied at the finest grid only, using the finite-element partition existing there. Higher-order accuracy is obtained with a MUSCL-approach [10]. Here,  $W_{ij}^l$  and  $W_{ij}^r$  are derived from linear interpolations. On each volume  $C_i$  around the triangle-vertex  $i$  an approximate gradient, denoted by  $(\bar{\nabla}W)_i$ , is derived by integrating the gradient of the linear inter-

polant of  $W$  over all the triangles which have  $i$  as a vertex:

$$(\bar{\nabla}W)_i = \left( \left( \frac{\bar{\partial}W}{\partial x} \right)_i, \left( \frac{\bar{\partial}W}{\partial y} \right)_i \right)^T, \quad \text{with} \quad (2.9a)$$

$$\left( \frac{\bar{\partial}W}{\partial x} \right)_i = \frac{\int_{\text{supp}(i)} \frac{\partial W}{\partial x} dx dy}{\int_{\text{supp}(i)} dx dy}, \quad \left( \frac{\bar{\partial}W}{\partial y} \right)_i = \frac{\int_{\text{supp}(i)} \frac{\partial W}{\partial y} dx dy}{\int_{\text{supp}(i)} dx dy}. \quad (2.9b)$$

In here,  $\text{supp}(i)$  denotes the union of triangles which have  $i$  as a vertex. Then for each pair of neighbouring vertices  $(i, j)$  we compute the extrapolated values

$$W_{ij}^l = W_i + \frac{1}{2}(\bar{\nabla}W)_i \cdot \bar{i}j, \quad W_{ij}^r = W_j - \frac{1}{2}(\bar{\nabla}W)_j \cdot \bar{i}j. \quad (2.10)$$

On equidistant grids, this higher-order accurate discretisation can be formally proved to be second-order accurate. The proof is still valid for nearly equidistant grids.

In order to ensure monotonicity, while preserving the higher-order accuracy in smooth flow regions, the higher-order values  $W_{ij}^l$  and  $W_{ij}^r$  according to (1.10) can be replaced by limited values which do not affect the order of accuracy.

### 2.3 Existing solution method

To solve the steady discretised system (1.7), we consider the unsteady, semi-discrete system of ordinary differential equations

$$\frac{dW_i}{dt} = R_i, \quad i = 1, 2, \dots, N. \quad (2.11)$$

The natural choice for  $R_i$  is

$$R_i = \frac{-1}{A_i} \sum_{j=1}^{n_i} \bar{T}_{ij}^{-1} \Phi(\bar{T}_{ij} W_{ij}^l, \bar{T}_{ij} W_{ij}^r) l_{ij}, \quad (2.12)$$

where  $A_i$  is the area of finite volume  $C_i$ .

As an upwind analogue to Jameson's central method (Jameson 1983), in [8] an explicit four-stage Runge-Kutta (RK4-) scheme is applied for the temporal integration of (1.11)-(1.12). The benefits of the upwind analogue are evident: better shock capturing, greater robustness and no tuning of explicitly added artificial viscosity. Similarly, just as in [6], in [8] multigrid is applied for accelerating the solution process. Furthermore, just as in [6], time accuracy is not pursued and optimal Runge-Kutta coefficients are applied to get good stability as well as good smoothing properties. It seems that the solution method presented in [8] is already competitive with Jameson's method, without the introduction of a further acceleration technique such as for example residual averaging.

It is of interest that the upwind analogue allows a further efficiency improvement by exploitation of the direct availability of the corresponding first-order

upwind discretisation, with its better stability and smoothing properties. Since a first-order central discretisation is not readily available, a standard central method does not easily allow this improvement.

#### 2.4 Improved solution method

Compared with the existing solution method, the new solution method only uses a more extensive right-hand side in the explicit time-stepping scheme. The extension consists of two first-order upwind defects, one which is evaluated at each stage of the multistage scheme, and another which is kept frozen during a fixed number of  $\nu_t$  RK4-time-steps ( $\nu_t \geq 1$ ) and which compensates for the other first-order defect by its opposite sign. Further - which is important - the higher-order defect is kept frozen as well during  $\nu_t$  RK4-steps. The four-stage time-stepping scheme is given in Table 1. In here,  $\nu$  is the time-step number,  $k$  the stage number,  $\Delta t_i$  the local time step and  $\alpha_k$  the  $k$ -th Runge-Kutta coefficient. In the existing higher-order method the right-hand side  $R_i^{\nu,k-1}$  is

$$R_i^{\nu,k-1} = \frac{-1}{A_i} \sum_{j=1}^{n_i} \bar{T}_{ij}^{-1} \Phi(\bar{T}_{ij}(W_{ij}^l)^{\nu,k-1}, \bar{T}_{ij}(W_{ij}^r)^{\nu,k-1}) l_{ij}, \quad (2.13)$$

with  $(W_{ij}^l)^{\nu,k-1}$  and  $(W_{ij}^r)^{\nu,k-1}$  higher-order accurate. So nothing is kept frozen in the existing method's right-hand side. For the improved method we take

$$R_i^{\nu,k-1} = \frac{-1}{A_i} \sum_{j=1}^{n_i} \bar{T}_{ij}^{-1} \left[ \Phi(\bar{T}_{ij} W_i^{\nu,k-1}, \bar{T}_{ij} W_j^{\nu,k-1}) - \Phi(\bar{T}_{ij} W_i^{0,0}, \bar{T}_{ij} W_j^{0,0}) + \Phi(\bar{T}_{ij} (W_{ij}^l)^{0,0}, \bar{T}_{ij} (W_{ij}^r)^{0,0}) \right] l_{ij}, \quad (2.14)$$

where only  $(W_{ij}^l)^{0,0}$  and  $(W_{ij}^r)^{0,0}$  are higher-order accurate. The frozen first-order cell-face states  $(W_i^{0,0}$  and  $W_j^{0,0})$  and the frozen higher-order cell-face states  $((W_{ij}^l)^{0,0}$  and  $(W_{ij}^r)^{0,0})$  are updated in an additional outer iteration, a

TABLE 1. Explicit RK4-scheme

---

```

W_i^{0,4} := W_i^{0,0},   i = 1, 2, ..., N
for  $\nu$  from 1 to  $\nu_t$  do
  W_i^{\nu,0} := W_i^{\nu-1,4},   i = 1, 2, ..., N
  for  $k$  from 1 to 4 do
    W_i^{\nu,k} := W_i^{\nu,0} + \Delta t_i \alpha_k R_i^{\nu,k-1},   i = 1, 2, ..., N
  enddo
enddo

```

---

defect correction iteration. For general information on defect correction processes we refer to [1]. For explanation and analysis of the present defect correction iteration we refer to [9]. Here, we directly proceed with an illustration of the performance of the present new method.

### 2.5 Numerical results

In [9], by analysis we found that the new higher-order method has better stability and smoothing properties than the existing higher-order method. In order to verify these predicted better stability and convergence properties, we compute the standard transonic channel flow from [12] with the two-dimensional Euler equations. Three finest grids are considered: a 161-vertices grid, an about twice as fine 585-vertices grid and an about four times as fine 2225-vertices grid. (See [8] for more grid details.) The corresponding solution schedules applied are a 4-, 5- and 6-levels schedule ( $L = 4, 5, 6$ ), respectively, all with  $\nu_{\text{pre}} = \nu_{\text{post}} = 1, \forall l$ . (For the definition of symbols we refer to [9].)

In Figure 1.3a we present various convergence histories as obtained for  $L = 4, 5, 6$ , respectively. The convergence results presented are:

- those of the first-order discretised Euler equations solved by means of the nonlinear multigrid iteration (dotted lines),
- those of higher-order discretised Euler equations solved by means of the existing higher-order method (dashed lines), and
- those of higher-order discretised Euler equations solved by means of the improved higher-order method (solid lines).

In all three graphs in Figure 1.3a, the residual considered is the  $L_2$ -norm of the error in the conservation of mass over all the finest-grid cells. Further, in all three graphs, the number of cycles indicated along the horizontal axis is:

- the number of FAS-cycles in case of both the first-order method and the existing higher-order method, and
- the number of ItDeC-cycles in case of the new higher-order method.

Note that with the new higher-order method, for  $\nu_{\text{FAS}} = 2, 5, 10$  the number of inner FAS-cycles is respectively 2, 5 and 10 times larger than the number of indicated ItDeC-cycles. (Only for  $\nu_{\text{FAS}} = 1$ , the number of FAS-cycles equals the number of ItDeC-cycles.) All convergence histories start at the end of the FMG-stage ([9]). In agreement with the theoretical results presented in [9], for all four values of  $\nu_{\text{FAS}}$  (so also for  $\nu_{\text{FAS}} = 1$ ), the new method does indeed give a better convergence than the existing higher-order method. For decreasing mesh width, the convergence of the new higher-order method becomes even relatively better than that of the first-order method. (For all four values of  $\nu_{\text{FAS}}$  under consideration, the corresponding convergence histories in Figure 1.3a show a better grid-independency than those of the multigrid method applied to the

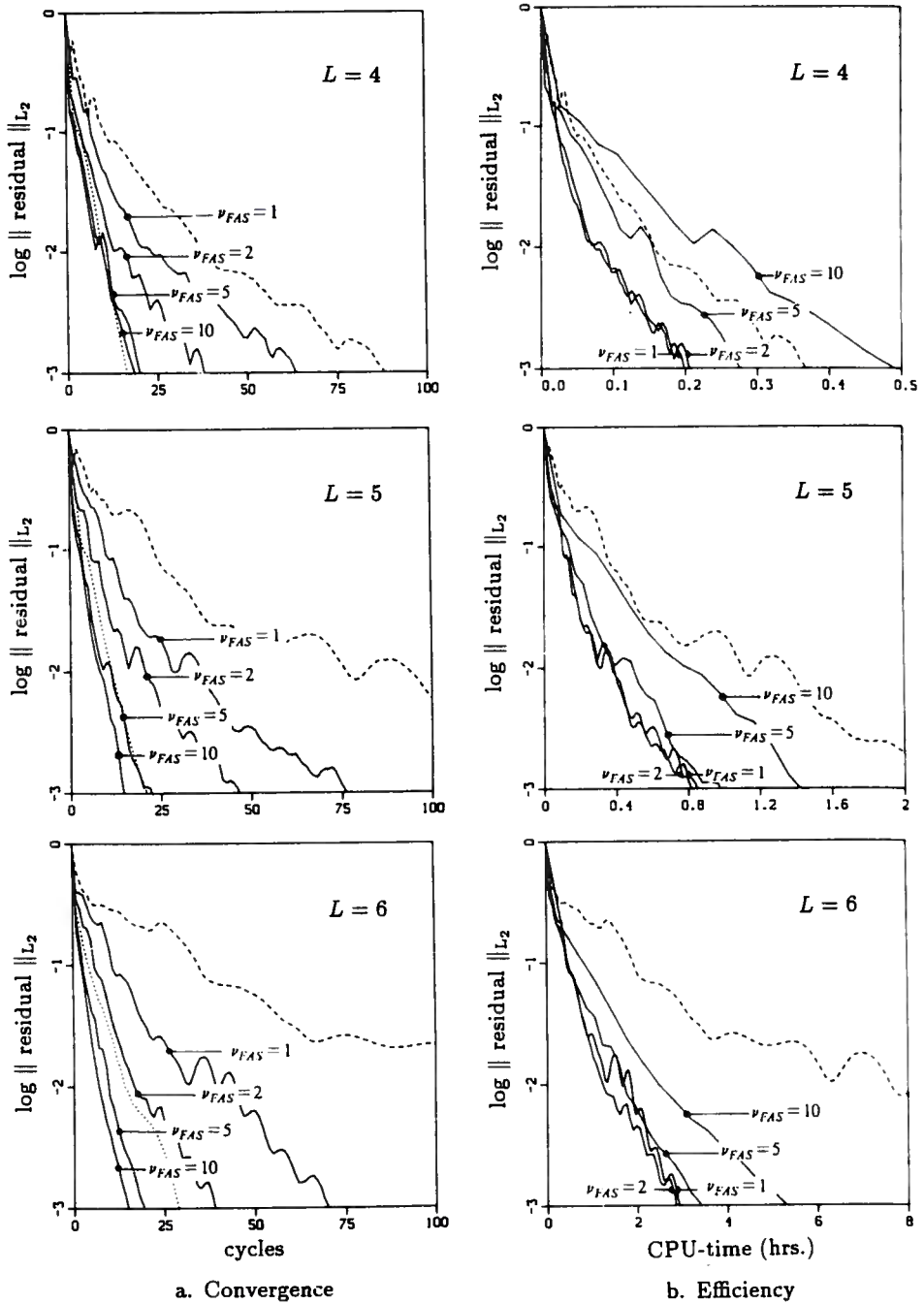


FIGURE 2.3. Convergence and efficiency histories (first-order method:  $\cdots$ , existing higher-order method:  $---$ , improved higher-order method:  $—$ )

first-order discretised equations.) This better performance is probably due to the predicted better smoothing in the new method.

As for the actual order of accuracy, if we took the converged higher-order accurate solution obtained on the 2225-vertices grid as the reference solution, we measured local orders of accuracy in the range  $[O(h^{1.4}), O(h^{2.3})]$  for the solutions on the coarser grids (the 585-vertices grid and the 161-vertices grid). The global order of accuracy appears to be almost  $O(h^2)$ .

Finally, the important question still remains which of the various higher-order methods is the most efficient. To answer this question, we give the higher-order efficiency histories in Figure 1.3b. (The indicated computing times have been obtained on a Sequent.) Since the sizes of the three grids considered are related to each other by approximately a factor 4, we have related the scales along the horizontal axes accordingly. Concerning the relative efficiency of the improved higher-order method, for the four values of  $\nu_{\text{FAS}}$  considered, it appears that for all three grids the best efficiency is obtained with  $\nu_{\text{FAS}} = 1$  (just as in [7], for the schedule with only a single FAS-cycle per ItDeC-cycle.) Further it appears - and this is important - that the improved method with  $\nu_{\text{FAS}} = 1$  is also more efficient than the existing higher-order method. Due to the better grid-independency of the improved method, this relatively better efficiency becomes even increasingly better with decreasing mesh width.

## 2.6 Conclusions

Fully implicit solution methods for higher-order discretised equations may strongly benefit from iterative defect correction when these systems of discretised equations are not easily invertible, which often is the case with higher-order accurate discretisations. Fully explicit solution methods may also profit from iterative defect correction. Here the profits are faster convergence and higher efficiency. The defect correction method appears to lead to greater stability (and hence to greater robustness) than the existing (standard) explicit method. Compared to the existing explicit method it possesses remarkably good smoothing properties, in fact even better than the first-order method. Last but not least its convergence rate appears to be grid-independent. For upwind discretisations, the 'price' which has to be paid for using defect correction iteration - a slightly more complex algorithm - is negligible, because of the direct availability of an appropriate approximate operator: the first-order upwind operator.

## 3 CONVERGENCE BEHAVIOUR OF DEFECT CORRECTION FOR HYPERBOLIC EQUATIONS

This section is based on the paper [2]. The nonlinear multigrid method is efficient for the solution of the compressible Navier-Stokes equations with a large Reynolds number, or for the Euler equations [5, 7]. The relaxation procedure being the workhorse of the multigrid method, the existence of a relaxation routine suited for fast reduction of the high frequency error components in the

solution of the discrete equations is essential for this success [5]. A good relaxation routine is found in point- or line-wise nonlinear (collective) Gauss-Seidel relaxation, assumed that we solve the first order accurate discrete equations.

For the second order discretisation the relaxation procedures are significantly less efficient. This is the reason why an additional iteration procedure is introduced as an outer loop: iterative defect correction (ItDeC [1]). The second order accurate approximation is now computed by the iteration

$$N_h^1(q_h^{(1)}) = 0, \quad (3.1)$$

$$N_h^1(q_h^{(i+1)}) = N_h^1(q_h^{(i)}) - N_h^2(q_h^{(i)}), \quad i = 1, 2, \dots. \quad (3.2)$$

Here  $N_h^1$  and  $N_h^2$  denote the first and second order (nonlinear) discrete operators. Only systems for first order accurate discrete equations are solved, but the fixed point of the iteration is the solution of the second order discrete system

$$N_h^2(q_h) = 0. \quad (3.3)$$

For the approximate solution of each iterate  $q_h^{(i+1)}$ ,  $i = 0, 1, \dots$ , a small number of multigrid iteration steps (and in many cases only a single step) is sufficient.

It is a classical result that, under easily satisfied conditions, the second iterate  $q_h^{(2)}$  is already second order accurate [4, Sect.14.2.2]. This result describes the convergence behaviour for the low-frequency difference between the first and second order discrete approximations. It explains why the convergence is fast for smooth solutions and fine grids. However, for the Navier-Stokes equations with high Reynolds number and for the Euler equations, sharp layers or discontinuities may exist in the solution. Therefore, it is of interest to study the total convergence behaviour for defect correction.

### 3.1 Linear model problem

In this contribution we restrict ourselves to the Euler equations. These equations form a hyperbolic system of conservation laws. To analyze the convergence for these equations, we first study the linear model problem in two dimensions

$$\frac{\partial}{\partial t} q + a \frac{\partial q}{\partial x} + b \frac{\partial q}{\partial y} = 0. \quad (3.4)$$

Although we are mainly interested in the steady state, we consider here the time-dependent problem in order to introduce a ‘flow direction’ so that inflow and outflow boundaries can be identified. The vector  $(a, b)^T$  determines the flow direction, and with  $a > 0$  the flow is in the positive  $x$ -direction.

For the first order discretisation, the simple upwind scheme is used. This scheme is described by its stencil

$$L_h^1 \sim \begin{bmatrix} & & 0 & \\ -a & a+b & 0 & \\ & & -b & \end{bmatrix}. \quad (3.5)$$



For the second order discretisation, various alternatives are available. Obvious possibilities are the second order upwind scheme and the central scheme, with the stencils

$$L_h^{2U} \sim \begin{bmatrix} 0 & & & & \\ 0 & & & & \\ \frac{a}{2} & -2a & \frac{3(a+b)}{2} & 0 & 0 \\ -2b & & & & \\ \frac{b}{2} & & & & \end{bmatrix}, \quad \text{and} \quad L_h^{2C} \sim \begin{bmatrix} & & & & \\ & & -\frac{b}{2} & & \\ -\frac{a}{2} & 0 & & \frac{a}{2} & \\ & -\frac{b}{2} & & & \\ & & & & \end{bmatrix}. \quad (3.6)$$

The corresponding linear operators are denoted by  $L_h^1$ ,  $L_h^{2C}$  and  $L_h^{2U}$ , for the first order and the second order central and upwind scheme respectively. By linear combination of  $L_h^{2C}$  and  $L_h^{2U}$  a scale of second order schemes is obtained, the so-called  $\kappa$ -schemes

$$L_h^{2\kappa} = \frac{1+\kappa}{2} L_h^{2C} + \frac{1-\kappa}{2} L_h^{2U}. \quad (3.7)$$

Here  $\kappa \in [-1, 1]$  is a free parameter that determines the particular scheme;  $\kappa = 0$  corresponds with Fromm's scheme. Being interested in the convergence of ItDeC, we study the amplification operator of the error,

$$M_h^\kappa = (L_h^1)^{-1} (L_h^1 - L_h^{2\kappa}). \quad (3.8)$$

### 3.2 One-dimensional analysis

We first study the operator  $M_h^\kappa$  in the one-dimensional case. Then, without loss of generality, we have

$$L_h^1 \sim [-1, 1, 0], \quad \text{and} \quad (3.9)$$

$$L_h^{2\kappa} \sim \frac{1+\kappa}{4} [-1, 0, 1] + \frac{1-\kappa}{4} [1, -4, 3, 0, 0]. \quad (3.10)$$

For an infinite, regular grid with mesh width  $h$ , eigenfunctions for these operators are  $u_\omega$ ,  $\omega \in [-\pi/h, \pi/h]$ , where  $u_\omega(jh) = e^{i\omega hj}$ . Corresponding eigenvalues of the operator  $M_h^\kappa$  are

$$\widehat{M}_h^\kappa(\omega) = i \sin(\omega h/2) \cos(\omega h/2) + \kappa \sin^2(\omega h/2). \quad (3.11)$$

This shows that the eigenvalues are located in the complex plane on an ellipse with axes  $x \in [0, \kappa]$ ,  $y \in [-1/2, 1/2]$ . From (3.11) we see that the upper bound for the convergence factor is

$$\sup_{\omega \in [-\pi/h, \pi/h]} |\widehat{M}_h^\kappa(\omega)| = \sup_{t \in [0, 1]} \sqrt{\kappa^2 t^2 + t(1-t)}.$$

Thus, as upper bounds we find

$$\sup_{\omega \in [-\pi/h, \pi/h]} |\widehat{M}_h^\kappa(\omega)| = \frac{1}{2} \frac{1}{\sqrt{1-\kappa^2}} \quad \text{for} \quad \kappa^2 \leq 1/2, \quad (3.12)$$

and

$$\sup_{\omega \in [-\pi/h, \pi/h]} |\widehat{M}_h^\kappa(\omega)| = |\kappa| \quad \text{for} \quad 1/2 \leq \kappa^2 \leq 1. \quad (3.13)$$

These expressions describe the convergence if no boundaries are present in the domain. To obtain an impression of the influence of the inflow Dirichlet boundary, we consider grid functions on a uniform partition  $\{x_i = ih; i = 0, 1, 2, \dots\}$  of the half-line  $[0, \infty)$  and we restrict ourselves to error components that vanish for large  $x_j$ . The operators  $L_h^1$  and  $L_h^{2\kappa}$  are again described by (3.9), (3.10), except for the first two equations in the system, that are determined by the boundary discretisation.

The eigenfunctions  $u_\lambda$  of  $M_h^\kappa$  and the corresponding eigenvalues  $\lambda$  satisfy the relation  $L_h^{2\kappa} u_\lambda = (1 - \lambda) L_h^1 u_\lambda$ , and from (3.10) it follows that  $u_\lambda$  has the form  $u_\lambda(jh) = A_0 + A_1 \mu_1^j + A_2 \mu_2^j$ , where  $\mu_1$  and  $\mu_2$  are roots of the equation

$$\frac{1 + \kappa}{2} \mu^2 + (2\lambda - \kappa) \mu - \frac{1 - \kappa}{2} = 0.$$

A straightforward computation [2] shows

$$\lambda = \frac{\kappa \pm i\sqrt{1 - \kappa^2} \cos \theta}{2}, \quad \theta \neq 0 \pmod{\pi}. \quad (3.14)$$

This shows that all eigenvalues are located on a line segment in the complex plane at a distance  $\kappa/2$  from the imaginary axis and that all eigenvalues satisfy  $|\lambda| \leq \frac{1}{2}$ .

In the case  $\kappa = \pm 1$ , we still have  $\rho = \max |\lambda| = 1/2$ , but the eigenvalues coalesce and the eigenvectors are no longer independent. Consequently, in the operator decomposition Jordan blocks  $J$  arise. In the one-dimensional case, on a finite interval, the size of these blocks is  $N - 1$ , where  $N$  is the number of mesh points. Then the convergence behaviour after  $n$  iterations of ItDeC is described by  $\tau_n = \|J^n\|_\infty$ , where

$$J^n = \begin{pmatrix} \rho^n & & & \\ \xi_j^n & \rho^n & & \\ \vdots & \ddots & \ddots & \\ & & \xi_j^n & \rho^n \end{pmatrix}, \quad \text{with} \quad \xi_j^n = \binom{n}{j} \rho^{n-j}.$$

It follows that  $\tau_n \geq \max_{j=0,1,2,\dots,N} |\xi_j^n|$ , and hence

- it is possible that  $\tau \geq 1$  if  $n < N$ ;
- $\tau_n \approx n^{N-1} \rho^n$  for  $n \rightarrow \infty$ , and hence the asymptotic convergence rate of the iteration is  $\rho \log |n|$ ;
- the sequence  $\{\tau_\nu\}_{\nu \leq n}$  is guaranteed to be decreasing only for  $n > N/(1 - \rho)$ . In our case  $\rho = 0.5$ . This implies that the iteration may show no convergence for the first  $2N$  iteration steps.

These phenomena are seen in practice indeed, as is shown in Figure 3.2.a-d.

If  $\kappa \neq \pm 1$  but  $1 \leq |\kappa| \leq \frac{1}{2}\sqrt{2}$ , the convergence during the first  $2N$  iteration steps is dominated by the behaviour as described by the Fourier analysis (3.13), i.e. a convergence rate of  $|\kappa|$  is seen. For all  $\kappa \in [-1, +1]$  the convergence rate has the lower bound  $\rho = 1/2$ .

In summary, for the one-dimensional problem we distinguish different phases in the convergence of the iterated defect correction. In most cases we first observe an impulsive start, where all components corresponding with small eigenvalues are damped. For the regular schemes (i.e.  $|\kappa|$  different from 1) soon an asymptotic rate of  $1/2$  is obtained. For the (near) pathological cases (i.e.  $|\kappa|$  close to 1), after the impulsive start, we distinguish first a Fourier (or pseudo-convection) phase for about  $2N$  iterations, in which the convergence is described by the Fourier analysis. After  $2N$  iterations the asymptotic rate  $1/2$  is found. In the truly degenerate cases ( $|\kappa| = 1$ ) we recognise a Fourier (pseudo-convection) phase, where the error does not decrease for  $2N$  iterations, and a logarithmic asymptotic rate due to the large Jordan block in the eigenvalue decomposition.

### 3.3 Two-dimensional analysis

In principle, the Fourier analysis for the two-dimensional difference operators (3.5,3.6) is completely analogous to the one-dimensional case. With the Fourier modes defined by  $u_\omega(hj) = e^{i(\omega_1 h_1 j_1 + \omega_2 h_2 j_2)}$ , where the subscripts refer to the  $x$ - and the  $y$ -directions respectively, we find

$$\widehat{L}_h^1(\omega) = 2ia e^{-i\omega_1 h_1/2} \sin(\omega_1 h_1/2) + 2ib e^{-i\omega_2 h_2/2} \sin(\omega_2 h_2/2), \quad (3.15)$$

and

$$\widehat{L}_h^{2\kappa}(\omega) = \begin{aligned} & 2ia e^{-i\omega_1 h_1/2} S_1(C_1^2 + iS_1 C_1 + (1 - \kappa)S_1^2) + \\ & 2ib e^{-i\omega_2 h_2/2} S_2(C_2^2 + iS_2 C_2 + (1 - \kappa)S_2^2), \end{aligned} \quad (3.16)$$

where  $S_1 = \sin(\omega_1 h_1/2)$ ,  $S_2 = \sin(\omega_2 h_2/2)$ ,  $C_1 = \cos(\omega_1 h_1/2)$  and  $C_2 = \cos(\omega_2 h_2/2)$ .

As the amplification factor we find

$$\begin{aligned} g(\omega) &= \left\| \widehat{M}_h^\kappa(\omega) \right\| = \left\| (\widehat{L}_h^1)^{-1} (\widehat{L}_h^1 - \widehat{L}_h^{2\kappa}) \right\| \\ &= \sqrt{\frac{(a_1 S_1^2 (1 - (1 - \kappa) S_1^2) + a_2 S_2^2 (1 - (1 - \kappa) S_2^2))^2 + (1 - \kappa)^2 (a_1 S_1^3 C_1 + a_2 S_2^3 C_2)^2}{(a_1 S_1^2 + a_2 S_2^2)^2 + (a_1 S_1 C_1 + a_2 S_2 C_2)^2}}}. \end{aligned} \quad (3.17)$$

This expression can be used to determine the convergence rate for the separate modes on an infinite domain. It shows that, for a given  $\kappa$ , we can never expect a better convergence rate in the two-dimensional case than in the one-dimensional case.

For the analysis of the two-dimensional case on a finite domain, we refer to [2]. Essentially, the results for two space dimensions can be seen as a perturbation

of the results for one dimension. Analogous to the one-dimensional domain, the location of the eigenvalues is shown in Figure 3.1. We now find the eigenvalues not on a line segment in the complex plane, but in a cloud near that line segment. The real part of the eigenvalues is generally larger than is the case in one dimension (for the same  $\kappa$ ). This means that the cloud is shifted to the right of the corresponding line segment. For the case of large  $\kappa$  ( $\kappa \approx +1$ ), the cloud is larger than for small  $\kappa$  ( $\kappa \approx -1$ ).

For different values of  $\kappa$  and for different values of  $N$  the location of the eigenvalues in the complex plane is shown in Figure 3.1. In this figure the ratio  $a/b$  is  $2/3$ .

In Figure 3.2 the convergence behaviour is shown for the model problem on a  $40 \times 40$ -mesh. For  $\kappa \leq 0$  the cloud of eigenvalues is still contained in the circle  $|z| \leq 0.5$ , so  $\rho(M_h^\kappa) \leq 0.5$  if  $\kappa \leq 0$ . However, for  $0 < \kappa \leq 1$  we find possibly  $\rho(M_h^\kappa) > 0.5$ , and for large  $\kappa$  we have  $\lim_{\kappa \rightarrow +1} \lim_{h \rightarrow 0} \rho(M_h^\kappa) = 1$ . This explains why a convergence rate  $\rho(M_h^\kappa) > 0.5$  is found for  $\kappa = 1/3$  in Figure 3.2.e whereas  $\rho(M_h^\kappa) = 0.5$  for  $\kappa = 0$  (Figure 3.2.e). For more details we refer to [2].

### 3.4 Euler equations

A similar behaviour, depending on  $\kappa$ , as for the linear model problem, is found for the nonlinear Euler equations. In Figure 3.3 we show the convergence behaviour for a problem that describes subsonic flow around a standard NACA0012 airfoil. This is a smooth flow where the problem is described by a complex nonlinear system of equations and the domain is not simply connected. The mesh is  $20 \times 32$  and results are shown for different values of  $\kappa$ . We see that the iteration doesn't converge for  $\kappa = 1$ , as it doesn't for  $\kappa = -1$  (not shown). We obtain slow convergence for  $\kappa = 0.8$  and  $\kappa = -0.8$ . Good convergence with a rate of approximately 0.5 per iteration step is obtained for  $\kappa = 1/3, 0$  and  $-1/3$ . Probably the asymptotic rate cannot be observed because rounding error accuracy is obtained after approximately 40 iterations. For  $\kappa = 1/3$  and  $\kappa = -1/3$  we see that after an initial phase with  $\rho \approx 0.5$ , we obtain another phase with a slightly slower convergence rate. Such effect is not (yet) seen for  $\kappa = 0$ .

The first order discrete equations are solved by a nonlinear multigrid method [5]. It employs a nonlinear symmetric point-Gauss-Seidel relaxation as a smoother and a nested sequence of Galerkin discretisations for the coarse grid corrections. Experience has shown that a small number of iteration cycles of this multigrid method solves the discrete system to a high degree of accuracy. In the experiments shown, 3 FAS V-cycles were applied for each single defect correction step. It was shown by experiments that the same results were obtained for multigrid iteration with 2 through 5 FAS V-cycles. All initial estimates were obtained by interpolation from a first order accurate solution on a coarser grid.

For this flow subsonic flow around the airfoil  $\kappa = 1$  gives an almost diverging

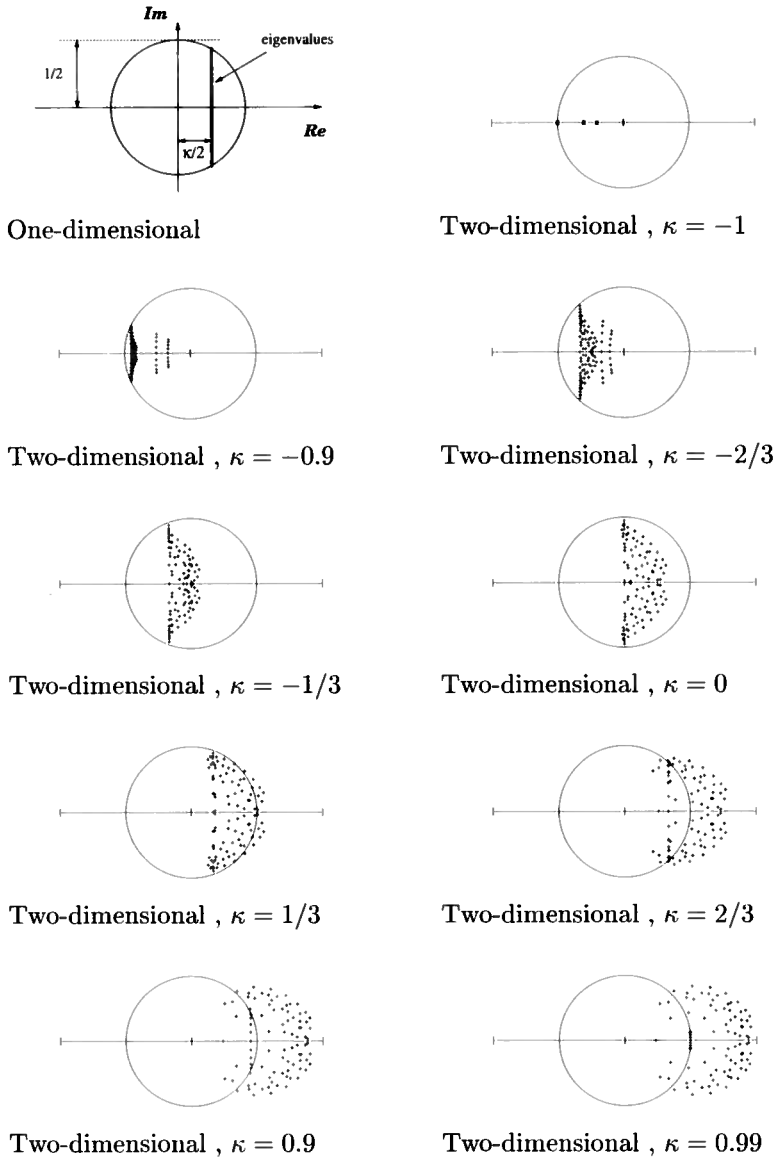


FIGURE 3.1. Location of the eigenvalues of the amplification matrix  $M_h^\kappa$  in the complex plane, relatively to the circle of radius  $1/2$ , for the one and the two-dimensional model problem. Except for the first 1-dim. figure, the mesh is  $10 \times 10$  and the ratio  $a/b = 2/3$ .

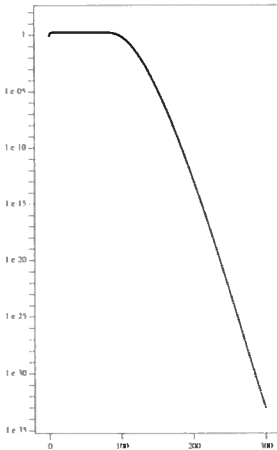


Figure 2.a, 1-dim  
 $N = 50; \kappa = 1$

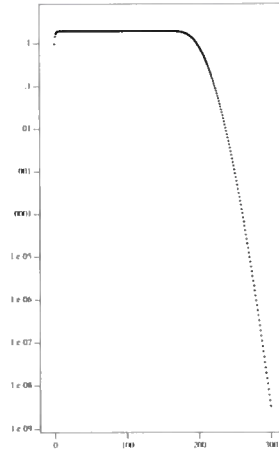


Figure 2.b, 1-dim  
 $N = 100; \kappa = 1$

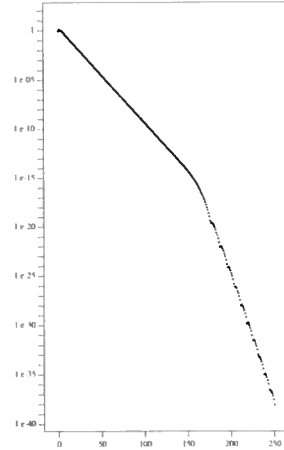


Figure 2.c, 1-dim  
 $N = 100; \kappa = 0.8$

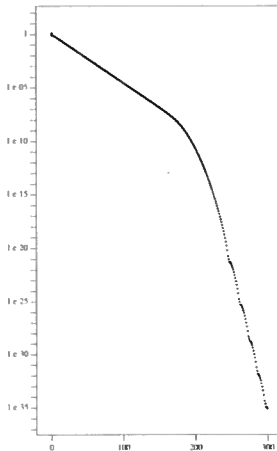


Figure 2.d, 1-dim  
 $N = 100; \kappa = -0.9$

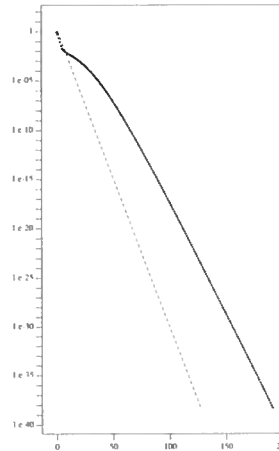


Figure 2.e, 2-dim  
 $40 \times 40$  mesh;  $\kappa = 1/3$

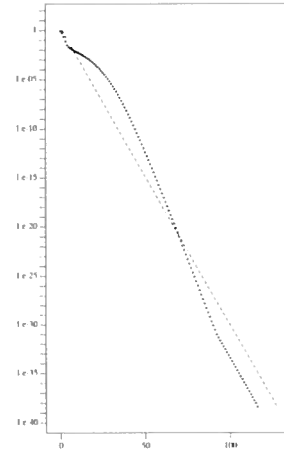


Figure 2.f, 2-dim  
 $40 \times 40$  mesh;  $\kappa = 0$

The sup-norm of the error is plotted against the number of iterations.  
 The dashed line corresponds with a convergence rate  $\rho = 1/2$ .

FIGURE 3.2. Convergence of ItDeC for the one- or two-dimensional linear test problem.

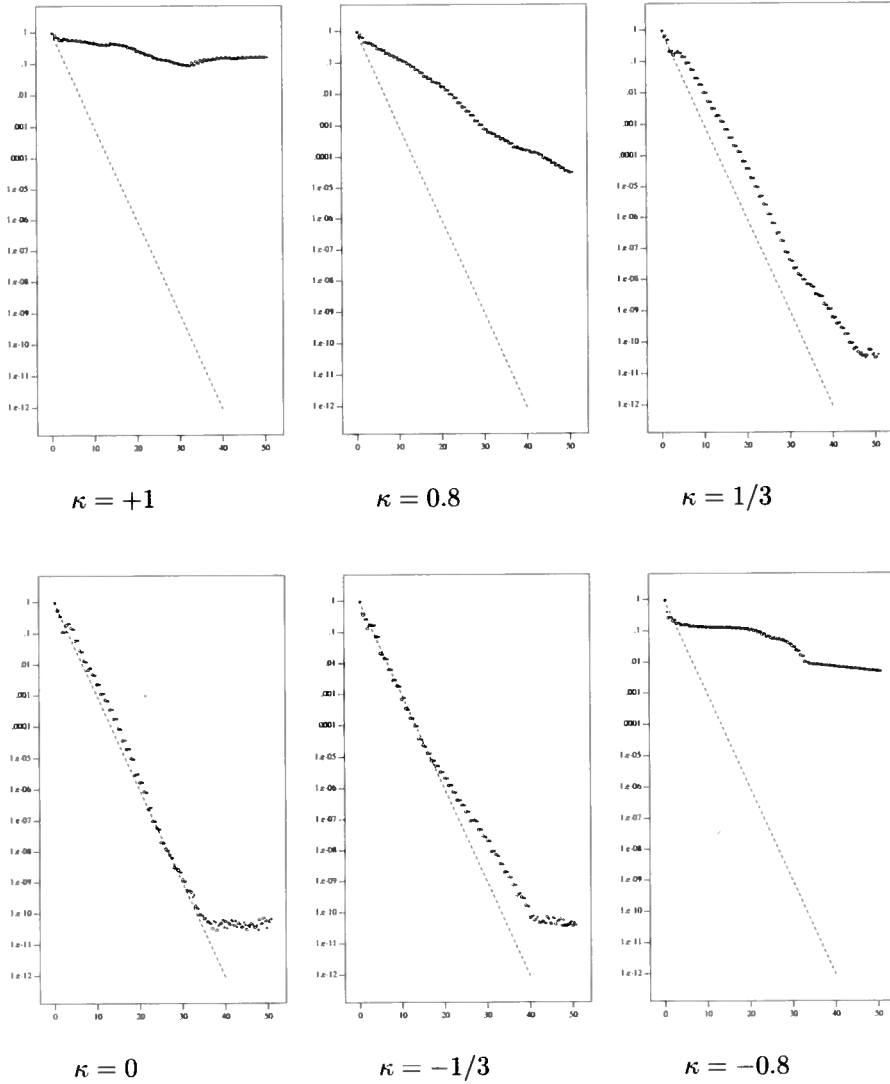


FIGURE 3.3. Subsonic Flow over a NACA0012 Airfoil  
 Convergence of the defect-correction method, on a  $20 \times 32$  mesh.  
 Mach number at infinity,  $M_\infty = 0.63$ , and the angle of attack  $\alpha = 2.0^\circ$ .  
 The dashed line corresponds to a convergence rate  $1/2$ .

process and  $\kappa = 0.8$  and  $\kappa = -0.8$  shows bad convergence. The asymmetry in the convergence behaviour with respect to  $\kappa > 0$  (worse) and  $\kappa < 0$  (better convergence) might be understood by the location of the eigenvalues in the complex plane (as shown in Figure 3.1). There we see that more eigenvalues are located in the neighbourhood of the origin for  $\kappa < 0$  than for  $\kappa > 0$ . This may be of greater importance for the nonlinear equations, where the corresponding eigenvectors are excited again and again, than for the linear problems, where the effect of these eigenvalues is no longer seen after a sufficient number of iterations.

#### REFERENCES

1. K. Böhmer, P.W. Hemker and H.J. Stetter, The defect correction approach, *Computing Suppl.*, **5**, 1-32, 1984.
2. J.-A. Désidéri and P.W. Hemker, Analysis of the convergence of iterative implicit and defect correction algorithms for hyperbolic problems, *SIAM J. Sci. Comput.*, (to appear, Jan. 1995).
3. S.K. Godunov, Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics (Cornell Aeronautical Lab. Transl. from Russian), *Mat. Sbornik*, **47**, 271-306, 1959.
4. W. Hackbusch, *Multi-Grid Methods and Applications*, Springer, Berlin, 1985.
5. P.W. Hemker and S.P. Spekreijse, Multiple grid and Osher's scheme for the efficient solution of the steady Euler equations, *Appl. Numer. Math.*, **2**, 475-493, 1986.
6. A. Jameson, Solution of the Euler equations for two dimensional transonic flow by a multigrid method, *Appl. Math. Comput.*, **13**, 327-355, 1983.
7. B. Koren, Defect correction and multigrid for an efficient and accurate computation of airfoil flows, *J. Comput. Phys.*, **77**, 183-206, 1988.
8. M.-H. Lallemand and A. Dervieux, A multigrid finite element method for solving the two-dimensional Euler equations, *Lecture Notes in Pure and Applied Mathematics*, **110**, 337-363 (S.F. McCormick, ed.), Marcel Dekker, New York, 1988.
9. M.-H. Lallemand and B. Koren, Iterative defect correction and multigrid accelerated explicit time stepping schemes for the steady Euler equations, *SIAM J. Sci. Comput.*, **14**, 953-970, 1993.
10. B. van Leer, Towards the ultimate conservative difference scheme V. A second-order sequel to Godunov's method, *J. Comput. Phys.*, **32**, 101-136, 1979.
11. S. Osher and F. Solomon, Upwind-difference schemes for hyperbolic systems of conservation laws, *Math. Comput.*, **38**, 339-374, 1982.
12. A. Rizzi and H. Viviand (eds.), Numerical Methods for the Computation of Inviscid Transonic Flows with Shock Waves, *Notes on Numerical Fluid Mechanics*, **3**. Vieweg, Braunschweig, 1981.



# A Natural Term Language

Jan van Eijck

This paper proposes a natural term language, investigates some of its properties, and discusses some of the advantages of natural term logic (NTL) as a medium for natural language semantics over its rivals and ancestors.

## 1 INTRODUCTION

In 1989 Cor Baayen was the prime mover behind the decision to start long-term work on the logic of natural language at CWI. Work in this area had found an occasional refuge at the centre before, witness Janssen [13], but the seed of a full scale research group in 'Logic and Language' was sown in the Autumn of 1989. Right now, five years later, the group consists of five researchers (six if we count a distinguished longtime guest), all but one supported by external funds. Fortunately for the rest of CWI we anticipate that this rate of growth will not be sustained in the future.

The main focus of current CWI research in 'Logic and Language' is on connections between programming language semantics and natural language semantics and on the design and analysis of suitable representation languages for natural language meaning. The connection with programming is explained by the fact that natural language representation should account for incrementality of processing, i.e., for the fact that we tend to understand each natural language utterance in the context of our understanding of what we have heard before. The semantics of a natural language text  $T$  consisting of  $T_1$  followed by  $T_2$  will specify that  $T_1$  sets up a context which is passed on as input to  $T_2$ , and that the meaning of  $T$  can be described as an increment of the meaning of  $T_1$ . This has a straightforward parallel in the analysis of computation: the semantics of a computer program  $P$  consisting of two parts  $P_1$  and  $P_2$ , in that order, will specify that the result of the computation to which  $P_1$  refers is passed on as input to  $P_2$ , and that the output of  $P_2$  for this input is the final output of  $P$ .

The paper starts with listing some desiderata for natural language representation, and then makes a new proposal for an incremental language for meaning representation.

## 2 WHAT MAKES AN NL REPRESENTATION LANGUAGE ‘NATURAL’?

If we assume that the meaning of (descriptive uses of) language should reveal itself in the conclusions we can draw from the truth of natural language utterances, the following requirement is possibly the most important:

**Suitability of Representation for Reasoning** The representation language should come with a sound and complete calculus for reasoning, and preferably with decidable and efficient sound reasoning systems for useful fragments of it.

First order logic meets this requirement quite well, as we know. More esoteric higher order representation languages such as Montague’s [18] Intensional Logic and its derivatives score lower in this dimension, as it is not always obvious how such logics should be axiomatized in the first place.

Another natural requirement on NL representation is the following:

**Structural Similarity of Representation** The structure of the logical representation language should bear a reasonable amount of similarity to that of the ‘source’ natural language.

At first sight, first order predicate logic does not meet this requirement at all. Consider (1), with its first order representation (2) (disregarding tense for simplicity). In the logical translation the subject–predicate structure of the natural language source seems to have got lost.

1 *A man walked in.*

2  $\exists x(Mx \wedge Wx)$ .

But here the appearance of the representation is misleading. If one thinks of the representation as the result of combining, by functional application, the meaning of the subject,  $\lambda P \cdot \exists x(Mx \wedge Px)$ , with that of the predicate,  $\lambda y \cdot Wy$ , then the structure of the source text reveals itself in the meaning representation of (1) before lambda reduction:

3  $(\lambda P \cdot \exists x(Mx \wedge Px))(\lambda y \cdot Wy)$ .

Still, the end result (2) of normalizing (3) does not have the same subject–predicate structure as the original. A representation where noun phrases reveal themselves in normal form as terms would satisfy the requirement better.

In the representation of the meaning of a very simple natural language example like (4), an extension of (1), we want to capture the fact that the first sentence of the example makes an indefinite reference to a man, while the second sentence picks up the reference to that same individual.

4 *A man walked in. He looked happy.*

The reason why ordinary first order predicate logic is letting us down here is that we also want our representation language to satisfy the following principle of incremental representation (already hinted at in the introduction above):

**Incrementality of Representation** The representation of a text  $T$  consisting of a subtext  $T_1$  followed by a subtext  $T_2$  should be an increment of the representation of  $T_1$ .

This principle is closely connected to, although not identical with, the principle of compositional interpretation which is the main preoccupation of Janssen's [13] investigations in Montague grammar.

In ordinary predicate logic, the natural representation of the first sentence of (4) is (2). This is not a suitable basis to construct a representation of the whole text (4). A natural representation of the pronoun *he* would use the variable  $x$ , but this choice runs into the problem that the scope of  $\exists x$  in (2) has been closed off.

The theory of discourse representation proposed in Kamp [14] tried to remedy this problem by assuming that every indefinite description gives rise to a so-called *discourse marker*, which can be picked up later on by an anaphoric link (*anaphora* is the standard linguistic name for the connection between the pronoun *he* and its antecedent *a man* in example (4)). Discourse representations à la Kamp essentially consist of sets or lists of discourse markers followed by lists of conditions. A discourse representation for the first sentence of (4) is given in (5)

$$5 \quad \{x\}, \{Mx, Wx\}.$$

In an analysis à la Kamp, the representation for the second sentence of the example can introduce a new marker  $y$  for *he*, and specify that the markers are to be linked:

$$6 \quad \{y\}, \{y = x, Hy\}.$$

The representation of the complete example text (4) is the result of an obvious process of 'merging' the two representations:

$$7 \quad \{x, y\}, \{Mx, Wx, y = x, Hy\}.$$

Later on, Groenendijk and Stokhof [8] observed that the essence of Kamp's proposal is already captured by a very simple modification of ordinary predicate logic. Replace Tarski's truth definition for first order logic by a dynamic variant which interprets a first order formula as a two-place relation on the set of variable assignments. The meaning of  $\varphi$  is then given as  $s[\varphi]s'$ , where  $s$  denotes the input assignment and  $s'$  the output assignment. All semantic clauses are tests, in the sense of imperative programming (where a test which gets memory state  $s$  as input indicates success by returning  $s$  as output and failure by giving no output at all), with the exception of  $\exists x$ , which has the clause  $s[\exists x]s'$  iff  $s' = s(x|d)$ , for some arbitrary  $d$  in the domain of the model under consideration.

If the predicate logical meaning of the first part of (1) is read dynamically in the manner indicated, and the pronoun in the second part of (1) is translated with the same variable, then in the end result this 'dangling' variable turns out to be bound after all, due to the continuing dynamic effect of the 'existential switch':

8  $\exists x(Mx \wedge Wx) \wedge Hx$ .

It is clear that the requirement of incremental representation leads in a natural way to a representation language with a dynamic semantics, and we can expect such representation languages to be similar to programming languages in interesting ways. For instance, it turned out that the dynamic version of predicate logic can be analysed with the standard tools from the study of imperative programming, such as Hoare logic (Van Eijck and De Vries [4]). Also, it became clear that dynamic predicate logic and its derivatives suffer from the problem of destructive assignment (see Dekker [1], Vermeulen [24] and Visser [25] for discussion and for possible remedies): because  $\exists x$  has been effectively replaced by the assignment statement  $x := ?$ , an existential quantification destroys the old value of its variable, with the result that anaphoric reference to that value by means of the variable (or a pronoun which has that variable as its translation) becomes impossible. The present proposal adds one more item to the long list of possible solutions for this problem.

### 3 THE BASIC IDEA

The basic idea of this paper is to design a language with complex ‘indefinite’ terms, with a dynamic semantics based on term valuations rather than variable assignments. This representation language is structurally more similar to natural language than languages which adopt the term structure of predicate logic, it caters for the needs of incremental representation by its dynamic nature, and it also looks like a promising tool for reasoning, due to its link to Hilbert’s epsilon calculus [9]. An earlier application of epsilon logic to the concerns of natural language representation is Meyer Viol [16].

The Natural Term Logic (NTL) to be defined in the next section is intended to achieve several goals at once:

- to give an account of the dynamics of left to right processing by means of a relational semantics (an idea from dynamic predicate logic [8], update logic [23], and similar proposals)
- to use intensional choice functions from epsilon logic [9] and instantial logic [6, 17] for the representation of indefinites,
- to account for the existential and universal quantifier in term of choices (friendly for existentials, unfriendly for universals), thus incorporating a key idea from Game Theoretical Semantics [11],
- to link pronouns to descriptions of their antecedents (the key idea of the so-called e-type analysis of pronouns proposed by Evans [5]),
- to treat universal and existential NPs as terms (one half of this idea incorporated in file change semantics and DRT; the full idea plays a role in traditional syllogistics and natural logic (Purdy [19], Sanchez [21],

Sommers [22]) and was all but killed off by Frege's Begriffsschrift analysis of quantification [7]).

#### 4 SEMANTICS OF NATURAL TERM LOGIC

We start with the non-logical vocabulary of a predicate logical language  $L$ . This consists of a set

$$C = \{c_0, c_1, c_2, \dots\}$$

of *names* (or *individual constants*), for each  $n > 0$  a set

$$P^n = \{P_0^n, P_1^n, P_2^n, \dots\}$$

of *n-place predicate constants* and for each  $n > 0$  a set

$$f^n = \{f_0^n, f_1^n, f_2^n, \dots\}$$

of *n-place function constants*.

It is also useful to have  $\perp$  for absurdity,  $=$  for identity, and  $\bar{\phantom{x}}$  for predicate negation. The further logical vocabulary we add to this consists of parentheses, the  $\epsilon$  term operator (borrowed from Hilbert and Bernays [9]), the colon  $:$ , an infinitely denumerable set  $V$  of individual variables, the sequential composition connective  $;$  and the connective  $\Rightarrow$  for dynamic implication.

Terms and formulas are defined by mutual recursion, as follows (assume  $c \in C$ ,  $v \in V$ ,  $f \in f^n$ ,  $P \in P^n$ ):

**terms**  $t ::= c \mid v \mid ft_1 \cdots t_n \mid (\epsilon v : \varphi)$ .

**formulas**  $\varphi ::= \perp \mid Pt_1 \cdots t_n \mid \bar{P}t_1 \cdots t_n \mid t_1 = t_2 \mid (\varphi_1 ; \varphi_2) \mid (\varphi_1 \Rightarrow \varphi_2)$ .

The translation in this language of Example (4) becomes:

$$9 \ W(\epsilon x : Mx); H(\epsilon x : Mx).$$

Note that in this translation the reference to the previously mentioned individual *a man* gets picked up by just repeating the term which was used to refer to that individual in the first place: the translation of *he* is the same as that of its antecedent.

An occurrence of  $v$  is bound in  $\varphi$  if  $v$  occurs inside a subformula  $\psi$  of the form  $(\epsilon v : \psi)$ , otherwise it is free in  $\varphi$ . I will write  $\varphi(v_1, \dots, v_n)$  to indicate that the free variables of  $\varphi$  are among  $v_1, \dots, v_n$ . Just as in standard predicate logic one has to take some care with substitution. If one wants to substitute  $t$  for free occurrences of  $v$  in  $\varphi$ , one should check that  $t$  is free for  $v$  in  $\varphi$ , i.e., that no free variable inside  $t$  is in danger of becoming bound in the result. Substituting  $(\epsilon x : Pxy)$  for  $x$  in  $R(\epsilon y : Sxy)x$ , would run into this problem, for instance. The problem can always be remedied by switching to an alphabetic variant. In the example case, the result would be  $R(\epsilon z : S(\epsilon x : Pxy)z)(\epsilon x : Pxy)$ . I will use  $\varphi(t/v)$  for the result of substituting  $t$  for all free occurrences of  $v$  in  $\varphi$ , with a switch to an alphabetic variant if the need arises. The result

of simultaneous substitution of  $t_1, \dots, t_n$  for free occurrences of  $v_1, \dots, v_n$ , respectively, in  $\varphi$ , with renaming of bound variables as the need arises, will be written as  $\varphi(t_1/v_1, \dots, t_n/v_n)$ .

Let  $M = \langle \text{dom}(M), \text{int}(M) \rangle$  be a first order model for the vocabulary of  $L$ . I will use  $c^M$ ,  $f^M$ ,  $P^M$  as shorthand for  $\text{int}(M)(c)$ ,  $\text{int}(M)(f)$  and  $\text{int}(M)(P)$ , respectively.

Let  $A$  be the set of variable assignments for  $L$  in  $M$ , i.e., let  $A$  be the set of functions  $\text{dom}(M)^V$ . We will use  $a, a'$  for members of  $A$ , and  $a(v|d)$  for the assignment  $a'$  with  $a'(t) = t$  for  $t \neq v$  and  $a'(t) = d$  for  $t = v$ .

Let  $T$  be the set of terms of  $L$ . We consider the set of partial functions

$$\text{dom}(M)^{[A \times T]}$$

as total functions in

$$B = (\text{dom}(M) \cup \{\uparrow\})^{A \times T}.$$

For  $T' \subseteq T$  and  $s \in B$ , let  $s \upharpoonright T'$  be the function  $s' \in B$  given by:

$$s'(a, t) = s(a, t) \text{ if } t \in T', \text{ and } s'(a, t) = \uparrow \text{ otherwise.}$$

Define  $\text{dom}(s)$  as:

$$\{\langle a, t \rangle \in A \times T \mid s(a, t) \neq \uparrow\}.$$

The relation  $\leq$  on  $B$  is defined as  $s \leq s'$  iff  $s' \upharpoonright \text{dom}(s) = s \upharpoonright \text{dom}(s)$ .

The set  $S \subseteq B$  of states for  $L$  in  $M$  is the set of those  $s \in B$  satisfying the following:

- $s(a, v) = a(v)$ ,
- $s(a, c) = c^M$ ,
- $s(a, ft_1 \cdots t_n) = \begin{cases} f^M(s(a, t_1), \dots, s(a, t_n)) \\ \text{if } s(a, t_1) \neq \uparrow, \dots, s(a, t_n) \neq \uparrow, \\ \uparrow \text{ otherwise.} \end{cases}$
- $s(a, \epsilon v : \varphi) = \begin{cases} d & \text{for some } d \in \llbracket \varphi \rrbracket_{s,a}^v \text{ if } \llbracket \varphi \rrbracket_{s,a}^v \neq \emptyset, \\ \uparrow & \text{otherwise.} \end{cases}$

where  $\llbracket \varphi \rrbracket_{s,a}^v$  is

$$\{d \in \text{dom}(M) \mid s, a(v|d)[\varphi]\},$$

with  $s, a(v|d)[\varphi]$  given by the following clauses (where we assume  $s, s', s'' \in S$  and  $a \in A$ ):

- |   |     |  |
|---|-----|--|
| $s, a[\varphi]$                           | iff | $\exists s'$ with $s \leq s'$ and $s, a[\varphi]s'$ ,  |
| $s, a[\perp]s'$                           | iff | never,   |
| $s, a[Pt_1 \cdots t_n]s'$                 | iff | $s \leq s', s'(a, t_1) \neq \uparrow, \dots, s'(a, t_n) \neq \uparrow,$<br>$\langle s'(a, t_1), \dots, s'(a, t_n) \rangle \in P^M,$    |
| $s, a[\bar{P}t_1 \cdots t_n]s'$           | iff | $s \leq s', s'(a, t_1) \neq \uparrow, \dots, s'(a, t_n) \neq \uparrow,$<br>$\langle s'(a, t_1), \dots, s'(a, t_n) \rangle \notin P^M,$ |
| $s, a[t_1 = t_2]s'$                       | iff | $s \leq s', s'(a, t_1) \neq \uparrow, s'(a, t_2) \neq \uparrow, s'(a, t_1) = s'(a, t_2),$  |
| $s, a[\varphi_1; \varphi_2]s'$            | iff | $\exists s''$ with $s, a[\varphi_1]s''$ and $s'', a[\varphi_2]s'$ ,  |
| $s, a[\varphi_1 \Rightarrow \varphi_2]s'$ | iff | $s = s'$ and $\forall s''$ with $s, a[\varphi_1]s''$ it holds that $s'', a[\varphi_2]$ .   |

## 5 ADEQUACY OF THE SEMANTIC DEFINITION

Note that the definition of the state set  $S$  for  $L$  in  $M$  is phrased in terms of  $S$  itself, a potentially dangerous situation. The next proposition shows that for every  $M$  for  $L$ , the set of states for  $L$  in  $M$  is non-empty.

**PROPOSITION 1** *If  $S$  is the set of states for  $L$  in  $M$ , then  $S \neq \emptyset$ .*

**Proof:** The proof uses a variation on a standard Skolem expansion argument (see e.g. Hodges [12]).

Start out with the following language  $L_0$ :

**terms**  $t ::= c \mid v \mid ft_1 \cdots t_n$ .

**formulas**  $\varphi ::= \perp \mid Pt_1 \cdots t_n \mid \bar{P}t_1 \cdots t_n \mid t_1 = t_2 \mid (\varphi_1; \varphi_2) \mid (\varphi_1 \Rightarrow \varphi_2)$ .

Let  $T_0$  be the set of terms of  $L_0$ . Surely, states for  $L_0$  exist, for a state for  $L_0$  is just a mapping from assignments to classical first order term valuations. Let  $S_0$  be the set of states for  $L_0$ . Note that  $\llbracket \varphi \rrbracket_{s,a}^v$  is well-defined for  $\varphi \in L_0$ .

Next, expand the language in layers. Assume  $T_k$ , the set of terms for layer  $k$ , and  $L_k$ , the set of formulas for layer  $k$ , are given. Then  $T_{k+1}$  and  $L_{k+1}$  are given by the following clauses:

**terms**  $t ::= c \mid v \mid ft_1 \cdots t_n \mid (\epsilon v : \varphi)$  with  $\varphi \in L_k$ ,

**formulas**  $\varphi ::= \perp \mid Pt_1 \cdots t_n \mid \bar{P}t_1 \cdots t_n \mid t_1 = t_2 \mid (\varphi_1; \varphi_2) \mid (\varphi_1 \Rightarrow \varphi_2)$   
with  $t \in T_{k+1}$ .

We may assume that  $S_k$ , the set of states for  $L_k$ , is non-empty. Also, we may assume that  $\llbracket \varphi \rrbracket_{s,a}^v$  is well-defined for  $\varphi \in L_k$ ,  $s \in S_k$ .

Take some member  $s_k \in S_k$  and use it to construct a member  $s$  of  $S_{k+1}$  as follows.

- if  $t \in T_k$ , then  $s(a, t) := s_k(a, t)$ ,
- if  $t \in T_{k+1} - T_k$ , then  $t$  has the form  $(\epsilon v : \varphi)$ , with  $\varphi \in L_k$ , and we set
 
$$s(a, \epsilon v : \varphi) := \begin{cases} d & \text{for some } d \in \llbracket \varphi \rrbracket_{s_k, a}^v \text{ if } \llbracket \varphi \rrbracket_{s_k, a}^v \neq \emptyset \\ \uparrow & \text{otherwise.} \end{cases}$$

Obviously, this can always be done, so we have shown that  $S_{k+1} \neq \emptyset$ , and moreover, that every  $s_k \in S_k$  can be extended to an  $s_{k+1} \in S_{k+1}$ . Also, if  $s \in S_{k+1}$ ,  $\llbracket \varphi \rrbracket_{s,a}^v$  will be well-defined for  $\varphi \in L_{k+1}$ .

The full language  $L$  is  $\bigcup_{k=0}^{\infty} L_k$ , the full set of terms  $T$  is  $\bigcup_{k=0}^{\infty} T_k$ . The set of states  $S$  for  $L$  in  $M$  is given by:

$$\{s \in B \mid s \upharpoonright T_k \in S_k, 0 \leq k < \infty\}.$$

As each  $S_k$  is non-empty and each  $s_k \in S_k$  has an extension  $s_{k+1} \in S_{k+1}$ , this proves that  $S \neq \emptyset$ . ■

## 6 TRUTH, VALIDITY AND ENTAILMENT

The following definitions of truth, validity and entailment round off the presentation of the semantics of  $L$ .

**DEFINITION 1 (TRUTH)**  $\varphi$  is true in  $L$ -model  $M$  if  $\exists s \in S, \exists a \in A$  with  $s, a[\varphi]$ , where  $S$  is the set of states for  $L$  in  $M$  and  $A = \text{dom}(M)^V$ .

Here are some examples of first order equivalents of NTL formulas to illustrate the definition (where  $\models$  denotes NTL truth, and  $\models_c$  the classical first order notion of truth).

- $M \models B(\epsilon x : Ax)$  iff  $M \models_c \exists x(Ax \wedge Bx)$
- $M \models R(\epsilon x : Ax)(\epsilon x : Bx)$  iff  $M \models_c \exists x \exists y(Ax \wedge By \wedge Rxy)$ .
- $M \models R(\epsilon x : Ax)(\epsilon x : Ax)$  iff  $M \models_c \exists x(Ax \wedge Rxx)$ .
- $M \models \bar{R}(\epsilon x : Ax)(\epsilon x : Ax)$  iff  $M \models_c \exists x(Ax \wedge \neg Rxx)$ .
- $M \models A(\epsilon x : Ax) \Rightarrow B(\epsilon x : Ax)$  iff  $M \models_c \forall x(Ax \rightarrow Bx)$ .

**DEFINITION 2 (VALIDITY)**  $\varphi$  is valid if  $\varphi$  is true in every  $L$ -model  $M$ .

Here is an example validity (with  $\models \varphi$  for ‘ $\varphi$  is valid’):

$$\models A(\epsilon x : Bx) \Rightarrow B(\epsilon x : Ax).$$

**DEFINITION 3 (ENTAILMENT)**  $\varphi$  entails  $\psi$  if the truth of  $\varphi$  in  $L$ -model  $M$  entails the truth of  $\psi$  in  $L$ -model  $M$ .

This may sound slightly non-standard. The reason for looking at the conclusion ‘in the context of the premise’ is of course that the conclusion may contain translations of pronouns that find an antecedent in the premise.

Here is an example entailment (with  $\models$  for the entailment relation):

$$\begin{aligned} & (P(\epsilon x : Ax) \Rightarrow P(\epsilon x : Bx)); (P(\epsilon x : Bx) \Rightarrow P(\epsilon x : Cx)) \\ & \models P(\epsilon x : Ax) \Rightarrow P(\epsilon x : Cx). \end{aligned}$$

The term language  $L$  is a dynamic variant of Hilbert and Bernays’ epsilon logic (see [9]). The dynamic epsilon terms are meant to represent the process of referring indefinitely to individual entities (by means of indefinite descriptions) in natural language.

Moreover, it is an intensional version, for two formulas  $\varphi$  and  $\psi$  which are logically equivalent (i.e., which entail one another) can give rise to different ‘epsilon choices’ in the sense that for some state  $s$ ,  $s(\epsilon v : \varphi) \neq s(\epsilon v : \psi)$ . In extensional epsilon logic (cf. Leisenring [15]) this situation cannot occur. For our purposes the intensionality of choice is indispensable, for we want to be able to use logically equivalent indefinite descriptions for indefinite reference to different individuals.



Some extra notation is useful for that. Note that according to the semantic clauses,  $(\perp \Rightarrow \perp)$  is valid. Let  $(\epsilon v : \varphi)_n$  abbreviate the following:

$$(\epsilon v : (\varphi; \underbrace{((\perp \Rightarrow \perp); ((\perp \Rightarrow \perp); \dots))}_n))$$

Then we can use  $(\epsilon x : Ax)_0$ ,  $(\epsilon x : Ax)_1$ ,  $(\epsilon x : Ax)_2$ , and so on, to translate different occurrences of an indefinite description in a text.

10 *A beer for her, a beer for him, and an orange juice for me.*

In ordering a round of drinks for three, as in (10), a repetition of the same indefinite description should not entail that the same glass is to be shared by two of your friends, so the translation should use  $(\epsilon x : Bx)_0$  and  $(\epsilon x : Bx)_1$ , for the two different glasses of beer.

## 7 AN UPDATE FORMULATION OF THE SEMANTICS

If  $I \subseteq S$ , where  $S$  is the state set for  $L$  in some given  $M$ , let  $I[\varphi]$  be the set of states given by:

$$\{s \in S \mid \exists s' \in I \exists a \in A : s', a[\varphi]s\}.$$

We can use this notion to define a global index elimination procedure for NTL. An index for  $L$  is a pair  $\langle M, I \rangle$ , where  $M$  is a model for  $L$  and  $I \subseteq S, I \neq \emptyset$ , with  $S$  the state set for  $L$  in  $M$ .

If  $U$  is a set of indices, then define:

$$U|\varphi| = \{\langle M, I[\varphi] \rangle \mid \langle M, I \rangle \in U \text{ and } I[\varphi] \neq \emptyset\}.$$

Let  $W$  be the class of all pairs  $\langle M, S \rangle$ , with  $M$  a model for  $L$  and  $S$  the full state set for  $L$  in  $M$ . Then  $\varphi$  is valid iff  $(W|\varphi|)_0$  equals the class of all models for  $L$ ; here  $( )_0$  denotes the operation of taking the first projection.

Let  $\mathcal{U}$  be the power set of the class of all indices for  $L$ . A natural information ordering on  $\mathcal{U}$  can now be given in terms of the local ordering  $\leq$  on states for a given model, which we first extend to state sets, as follows:

$$I \leq J \text{ iff for all } s \in J \text{ there is an } s' \in I \text{ with } s' \leq s.$$

Next, we set, for  $U_1, U_2 \in \mathcal{U}$ :

$$U_1 \leq U_2 \text{ iff for all } \langle M, J \rangle \in U_2 \text{ there is a } I \leq J \text{ with } \langle M, I \rangle \in U_1.$$

This distinction between a global and a local perspective on the semantics should be compared to a similar distinction made for dynamic modal predicate logic, in Van Eijck and Cepparello [3]. The distinction is the key to extending the present proposal with epistemic operators such as *maybe*, an extension which is beyond the scope of the present paper, however.

## 8 SOME EXAMPLE MEANING REPRESENTATIONS

We will now illustrate the potential of the language by a brief discussion of examples, some of them famous from the literature.

11 *Some farmer owns a donkey. He beats it.*

Natural translation:

12  $O(\epsilon x : Fx)(\epsilon y : Dy); B(\epsilon x : Fx)(\epsilon y : Dy)$ .

This does have the expected meaning, for it is equivalent to the following first order sentence:

13  $\exists x \exists y (Fx \wedge Dy \wedge Oxy \wedge Bxy)$ .

The advantage of the NTL version is the fact that the translation of the second part is an increment of that of the first.

14 *If a farmer owns a donkey, he beats it.*

The translation of this key motivating example for Discourse Representation Theory:

15  $O(\epsilon x : Fx)(\epsilon y : Dy) \Rightarrow B(\epsilon x : Fx)(\epsilon y : Dy)$ .

The first order equivalent of this:

16  $\forall x \forall y ((Fx \wedge Dy \wedge Oxy) \rightarrow Bxy)$ .

This example derives its fame from the fact that its first order translation is so hard to get in a compositional way. The NTL version does not face such a problem.

17 *Every farmer who owns a donkey beats it.*

To treat the example it is useful to have a notation of universal terms. Let  $P(\dots(\tau v : \varphi)\dots)$  be shorthand for:

$$(\epsilon v : \varphi = \epsilon v : \varphi) \Rightarrow P(\dots(\epsilon v : \varphi)\dots).$$

Then (17) gets as natural translation:

18  $B(\tau x : Fx; Ox(\epsilon y : Dy))(\epsilon y : Dy)$ .

This is shorthand for:

19  $(\epsilon x : Fx; Ox(\epsilon y : Dy)) = (\epsilon x : Fx; Ox(\epsilon y : Dy))$   
 $\Rightarrow B(\epsilon x : Fx; Ox(\epsilon y : Dy))(\epsilon y : Dy)$ ,

which has the same first order equivalent as (15).

20 *Every farmer owns a donkey. He beats it (regularly).*

The discourse representation literature [14] claims that the example is ill-formed, a nice illustration of the fact that linguistic observation, like all observation in science, is biased by theory. Unlike discourse representation theory, which cannot handle it, we can afford to assume that this example is linguistically acceptable. Here is the translation:

$$21 \ O(\tau x : Fx)(\epsilon y : Dy); B(\tau x : Fx)(\epsilon y : Dy).$$

Its first order equivalent:

$$22 \ \forall x(Fx \rightarrow \exists y(Dy \wedge Oxy)) \wedge \forall x(Fx \rightarrow \exists y(Dy \wedge Bxy)).$$

If this isn't close enough, we can relax our regime of pronoun translation which says that pronouns are to be translated by repetition of the term translation of their antecedent.

$$23 \ O(\tau x : Fx)(\epsilon y : Dy).$$

In fact, from the truth of (23) we get that in every setting the term  $(\tau x : Fx)$  can be replaced *salva veritatis* by  $(\tau x : Fx; Ox(\epsilon y : Dy))$ . Using this as pronoun translation we get:

$$24 \ O(\tau x : Fx)(\epsilon y : Dy); B(\tau x : Fx; Ox(\epsilon y : Dy))(\epsilon y : Dy).$$

The first order equivalent of (24):

$$25 \ \forall x(Fx \rightarrow \exists y(Dy \wedge Oxy)) \wedge \forall x(Fx \rightarrow \forall y((Dy \wedge Oxy) \rightarrow Bxy)).$$

$$26 \ \textit{Every farmer owns a donkey. Some farmer beats it.}$$

Like the previous example, this one is beyond the scope of most current semantic theories. Outside of the mainstream of natural language semantics, Game Theoretical Semantics [10] does sketch an account, however. NTL now incorporates this treatment in standard dynamic semantics. Here is a translation:

$$27 \ O(\tau x : Fx)(\epsilon y : Dy); B(\epsilon x : Fx)(\epsilon y : Dy).$$

Its first order equivalent:

$$28 \ \forall x(Fx \rightarrow \exists y(Dy \wedge Oxy)) \wedge \exists x(Fx \wedge \exists y(Dy \wedge Bxy)).$$

Again, if this isn't close enough, we can relax the pronoun translation regime and observe that the truth of the first half of (27) guarantees that we can replace the term  $(\epsilon x : Fx)$  by  $(\epsilon x : Fx \wedge Ox(\epsilon y : Dy))$  without changing truth conditions. This gives the following alternative translation:

$$29 \ O(\tau x : Fx)(\epsilon y : Dy); B(\epsilon x : Fx; Ox(\epsilon y : Dy))(\epsilon y : Dy),$$

with first order equivalent:

$$30 \ \forall x(Fx \rightarrow \exists y(Dy \wedge Oxy)) \wedge \exists x(Fx \wedge \exists y(Dy \wedge Oxy \wedge Bxy)).$$

Of course, all first order equivalents in this section were given *ad hoc*. In the next section the issue of reasoning about and in NTL will be addressed in a more systematic way.

## 9 ASSERTION REASONING FOR NATURAL TERM LOGIC

One approach to developing a calculus for a dynamic logic is by using assertions, in the style of Hoare logic or quantified dynamic logic. The statements from the dynamic language to be analyzed then become modalities, and we interpret  $\langle \varphi \rangle X$  as: there is some state  $s'$  reachable from the current state  $s$  with  $s, a[\varphi]s'$  and  $X$  holds at  $s'$ , and its dual  $[\varphi]X$  as: for all states  $s'$  reachable from the current state  $s$  with  $s, a[\varphi]s'$ ,  $X$  holds at  $s'$ .

Here are some axioms for an assertion calculus along these lines (we use  $X$  as metavariable over assertion statements, and  $\top$  as abbreviation of some arbitrary tautology).

$$\text{A 1 } \langle \varphi_1; \varphi_2 \rangle X \leftrightarrow \langle \varphi_1 \rangle \langle \varphi_2 \rangle X.$$

$$\text{A 2 } \langle \varphi_1 \Rightarrow \varphi_2 \rangle X \leftrightarrow (X \wedge [\varphi_1] \langle \varphi_2 \rangle \top).$$

$$\text{A 3 } \langle P(\dots (\epsilon v : \varphi) \dots) \rangle X \leftrightarrow \exists x(\langle \varphi \rangle \top \wedge \langle P(\dots x \dots) \rangle X(v/(\epsilon v : \varphi))).$$

$$\text{A 4 } [P(\dots (\epsilon v : \varphi) \dots)] X \leftrightarrow \forall x(\langle \varphi \rangle \top \rightarrow [P(\dots x \dots)] X(v/(\epsilon v : \varphi))).$$

$$\text{A 5 } \langle P t_1 \dots t_n \rangle X \leftrightarrow (P t_1 \dots t_n \wedge X).$$

Condition on A-5: none of the  $t_i$  is of the form  $(\epsilon v : \varphi)$ .

$$\text{A 6 } [P t_1 \dots t_n] X \leftrightarrow (P t_1 \dots t_n \rightarrow X).$$

Condition on A-6: none of the  $t_i$  is of the form  $(\epsilon v : \varphi)$ .

Further discussion of these axioms is beyond the present scope (see Van Eijck [2] for a similar calculus for dynamic predicate logic).

Instead, we confine ourselves to illustrating their use by means of the following example.

$$\begin{aligned} & \langle O(\epsilon x : Fx)(\epsilon y : Dy) \Rightarrow B(\epsilon x : Fx)(\epsilon y : Dy) \rangle \top \\ & \leftrightarrow [O(\epsilon x : Fx)(\epsilon y : Dy)] \langle B(\epsilon x : Fx)(\epsilon y : Dy) \rangle \top \\ & \leftrightarrow \forall x(\langle Fx \rangle \top \rightarrow [Ox(\epsilon y : Dy)] \langle Bx(\epsilon y : Dy) \rangle \top) \\ & \leftrightarrow \forall x(Fx \rightarrow [Ox(\epsilon y : Dy)] \langle Bx(\epsilon y : Dy) \rangle \top) \\ & \leftrightarrow \forall x(Fx \rightarrow \forall y(\langle Dy \rangle \top \rightarrow [Oxy] \langle Bxy \rangle \top)) \\ & \leftrightarrow \forall x(Fx \rightarrow \forall y(Dy \rightarrow [Oxy] \langle Bxy \rangle \top)) \\ & \leftrightarrow \forall x(Fx \rightarrow \forall y(Dy \rightarrow (Oxy \rightarrow Bxy))). \end{aligned}$$

## 10 NATURAL DEDUCTION FOR NATURAL TERM LOGIC

A different approach to reasoning with term logic is given by the following example rules from a natural deduction calculus with ordered premises.

$$\text{A 1 } \frac{\varphi; \psi}{\varphi}$$

$$\text{A 2 } \frac{\varphi; \psi(\epsilon v : \chi/v)}{\psi(\epsilon v : (\chi \wedge \varphi)/v)}$$

Condition on A-2:  $\varphi$  should not contain occurrences of epsilon terms. An example application of the second rule is:

$$\frac{W(\epsilon x : Mx); H(\epsilon x : Mx)}{H(\epsilon x : Mx \wedge Wx)}$$

These rules are for purposes of illustration only. Axiom A-2 needs a more complex formulation to deal with cases where the first member  $\varphi$  of the sequence  $\varphi; \psi$  contains more than one epsilon term.

Further work on natural deduction for NTL should establish a connection with natural deduction for standard epsilon logic (see Meyer Viol [17] for a treatment).

## 11 CONCLUSION AND FURTHER DIRECTIONS

We have sketched a representation for natural language meaning which treats indefinite descriptions as terms. An obvious first extension is definite descriptions, for which standard logic has a term treatment using the  $\iota$  term operator (see e.g. Reichenbach [20] for an illuminating discussion). Further extensions of the representation language that seem interesting are epistemic modalities and, in a different direction, plural terms.

## REFERENCES

1. P. Dekker. *Transsentential Meditations*. PhD thesis, Department of Philosophy, University of Amsterdam, 1993.
2. J. van Eijck. Axiomatizing dynamic predicate logic with quantified dynamic logic. In J. van Eijck and A. Visser, editors, *Logic and Information Flow*, pages 30–48. MIT Press, Cambridge Mass, 1994.
3. J. van Eijck and G. Cepparello. Dynamic modal predicate logic. In M. Kanazawa and C.J. Pinón, editors, *Dynamics, Polarity and Quantification*, pages 251–276. CSLI, Stanford, 1994.
4. J. van Eijck and F.J. de Vries. Dynamic interpretation and Hoare deduction. *Journal of Logic, Language, and Information*, 1:1–44, 1992.
5. G. Evans. Pronouns. *Linguistic Inquiry*, 11:337–362, 1980.
6. K. Fine. *Reasoning With Arbitrary Objects*. Blackwell, Oxford, 1985.
7. G. Frege. *Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens*. Verlag Nebert, Halle, 1879.
8. J. Groenendijk and M. Stokhof. Dynamic predicate logic. *Linguistics and Philosophy*, 14:39–100, 1991.
9. D. Hilbert and P. Bernays. *Grundlagen der Mathematik*. Springer, Berlin, 1939. Second edition: Berlin 1970.
10. J. Hintikka. Quantifiers in natural languages: Some logical problems. In I. Niiniluoto and E.E. Saarinen, editors, *Essays in Mathematical and Philosophical Logic*, pages 295–314. Reidel, Dordrecht, 1979.
11. J. Hintikka and J. Kulas. *Anaphora and Definite Descriptions: Two Applications of Game-Theoretical Semantics*. Reidel, Dordrecht, 1985.

12. W. Hodges. *Model Theory*. Cambridge University Press, 1993.
13. T.M.V. Janssen. *Foundations and Applications of Montague Grammar*. CWI Tract 19. CWI, Amsterdam, 1986.
14. H. Kamp. A theory of truth and semantic representation. In J. Groenendijk et al., editors, *Formal Methods in the Study of Language*. Mathematisch Centrum, Amsterdam, 1981.
15. A.C. Leisenring. *Mathematical Logic and Hilbert's  $\epsilon$ -symbol*. MacDonald, London, 1969.
16. W. Meyer Viol. Partial objects and DRT. In P. Dekker and M. Stokhof, editors, *Proceedings of the Eighth Amsterdam Colloquium*, pages 381–402. ILLC, Amsterdam, 1992.
17. W. Meyer Viol. *Instantial Logic*. PhD thesis, OTS, Utrecht, to appear.
18. R. Montague. The proper treatment of quantification in ordinary english. In J. Hintikka e.a., editor, *Approaches to Natural Language*, pages 221–242. Reidel, 1973.
19. W.C. Purdy. A variable-free logic for anaphora. Manuscript, Syracuse University, 1992.
20. H. Reichenbach. *Elements of Symbolic Logic*. Macmillan, London, 1947.
21. V. Sánchez. *Studies on Natural Logic and Categorical Grammar*. PhD thesis, University of Amsterdam, 1991.
22. F. Sommers. *The Logic of Natural Language*. Cambridge University Press, 1982.
23. F. Veltman. Defaults in update semantics. Technical report, Department of Philosophy, University of Amsterdam, 1991. To appear in the *Journal of Philosophical Logic*.
24. C.F.M. Vermeulen. *Explorations of the Dynamic Environment*. PhD thesis, OTS, Utrecht, 1994.
25. A. Visser. The design of dynamic discourse denotations. Lecture notes, Utrecht University, February, 1994.

# The full non-renameability result; a lost tale

Dedicated to Cor Baayen at the occasion of his retirement from the CWI

Peter van Emde Boas

*ILLC, FWI, UvA; Plantage Muidergracht 24, 1018 TV Amsterdam; peter@fwi.uva.nl*  
*supported in part by HCM program COLORET (Complexity, logic and recursion theory, nr.*  
*CHRX-CT93-0415 (DG 12 COMA))*

The *naming theorem*, one of the classical results in Abstract Complexity Theory states that the entire hierarchy of complexity classes under an arbitrary complexity measure can be renamed using an effective measured transformation by a honest collection of names preserving the extension of the classes. The *non-renameability result* which was proven by the author two decades ago states the opposite to be the case for the hierarchy of honesty classes: every attempted measured transformation must destroy the extension of at least one honesty class. However, the published version of the theorem uses the fact that in the theory partial functions are first class citizens; a version involving total functions only is proven under restrictions on the names of the classes. In this note we present the full version of the theorem; this result was obtained and announced twenty years ago but has remained unpublished since.

## 1 INTRODUCTION

Abstract Complexity Theory is a research subject which connects Recursion theory and Theoretical Computer Science. It finds its origin in the seminal paper by Manuel Blum [2], was intensively studied during the early seventies, but it has become obsolete and forgotten by 1980. The subject can be found in several textbooks from that period, but an almost complete survey will also be included in the second volume of Odifreddi's textbook on recursion theory [6].

At the time I was completing my thesis on this subject [8] under supervision of A. van Wijngaarden and Cor Baayen with Juris Hartmanis serving as referee, interest in complexity theory already had shifted to the study of the fundamental complexity classes based on standard computational devices and to the study of the fundamental questions about the power of nondeterminism and the relation between time and space which are unsolved until today. Actually I know only of two ph.d. projects which have been completed on Abstract Complexity Theory since 1974 [1, 3]. So by the end of the seventies the subject

had quite well become stable. It was my intent to transform my thesis into a two-volume textbook on Abstract Complexity Theory which has in fact been announced for many years and which has been consuming an open slot in the Mathematical Centre Tract series until this series was replaced by the CWI tracts series. This book however was never completed.

Some central results from my thesis have been published in two papers [9, 10]. Other results from the research only have appeared in the thesis itself or were never published at all. The oldest of these two papers presents the so-called *non-renameability result*. This theorem establishes the most visible distinction between the structure of the hierarchy of complexity classes and the hierarchy of honesty classes: whereas according to the *naming theorem* of Mc Creight and Meyer [5] the entire collection of complexity classes can be renamed by means of a measured transformation preserving the extension of all classes, it is shown that every attempt to rename honesty classes in a similar way must destroy the extension of at least one class.

The result in the present note is an improvement of the results in [9]; it was obtained during my residence at Cornell after my ph.d. defense. When the galley proofs of this paper were sent to the printer I inserted a *note added in proof* announcing the full version of theorem 7 in that paper. Since the commercial edition of the thesis (where the promised improvement was stated to be appearing) was never completed the result only exists as a manuscript. Evidently after these many years the result might legally have been claimed by an independent researcher but this has not happened either. I am therefore grateful to be offered the opportunity to use the invitation to contribute to this volume dedicated to Cor Baayen to retrieve it from my archives in order to preserve it for prosperity.

## 2 PRELIMINARIES

By a function in this paper we mean a *partial recursive function* from the set of integers  $\mathbf{N}$  into itself. Functions which are defined for all arguments are called *total*. The symbol  $\mathcal{P}(\mathcal{R})$  denotes the set of all partial (total) functions. The set of arguments  $x$  for which  $f(x)$  is defined is denoted  $\mathcal{D}f$ . We write  $f(x) \leq \infty (f(x) = \infty)$  for  $x \in \mathcal{D}f (x \notin \mathcal{D}f)$ .

The inequality  $f \leq g$  means that  $\mathcal{D}f \subseteq \mathcal{D}g$  and  $g(x) \geq f(x)$  for  $x \in \mathcal{D}g$ . Strict inequality  $f < g$  means that  $\mathcal{D}f \subseteq \mathcal{D}g$  and  $g(x) > f(x)$  for  $x \in \mathcal{D}g$ . If  $\mathcal{D}f \subseteq \mathcal{D}g$  and  $g(x) = f(x)$  for  $x \in \mathcal{D}g$  then we write  $g \subseteq f$ . The range of  $f$  is denoted  $\mathfrak{R}f$ .

For finite  $k$  the inequalities  $k \leq \infty$  and  $\infty \leq \infty$  are taken to be true whereas  $\infty \leq k$  is false. Beside inequalities on all arguments we also have inequalities holding *almost everywhere*. If  $P(x)$  is some predicate we write  $\overset{\infty}{\forall}_x [P(x)]$  for “ $P(x)$  holds for all  $x$  except finitely many” and  $\overset{\infty}{\exists}_x [P(x)]$  for “there exist infinitely many  $x$  such that  $P(x)$ ”. Using these notations we let  $f(x) \leq g(x)$  denote  $\overset{\infty}{\forall}_x [f(x) \leq g(x)]$ . This later notation can be relativized moreover to a



subset  $A \subseteq \mathbf{N}$ : we let  $f(x) \preceq g(x)(A)$  denote  $\forall_x^\infty [x \in A \rightarrow f(x) \leq g(x)]$ . We let  $\mu z[P(z)]$  denote "the least  $z$  such that  $P(z)$ ".

We use a fixed recursive *pairing function*  $\langle x, y \rangle$  with coordinate projection functions  $\pi_1$  and  $\pi_2$ . We have  $\pi_1(\langle x, y \rangle) = x, \pi_2(\langle x, y \rangle) = y$  and  $\langle \pi_1 z, \pi_2 z \rangle = z$ . Moreover  $\langle x, y \rangle$  is increasing in both arguments and consequently  $\langle 0, 0 \rangle = 0$ . We let  $\epsilon(\mathbf{zero})$  denote the function which is everywhere undefined (zero). According to our convention  $\epsilon^2(\mathbf{zero}^2)$  denote the two argument function which is everywhere undefined (zero). Using this pairing function we can interpret one-variable functions as being many-variable functions; an occasional superindex like for example in  $\varphi_i^2(x, y) = \varphi_i(\langle x, y \rangle)$  indicates the use of this interpretation.

By  $(\varphi_i)_i$  we denote a fixed *Gödel numbering* of partial recursive functions [7]. The *universal function*  $u(i, x) = \varphi_i(x)$  is recursive and there exists a total function  $s$ , called the *s-n-m function* satisfying  $\varphi_i^2(x, y) = \varphi_{s(i, x)}(y)$ . Using the interpretation  $\varphi_i^2(x, y) = \varphi_i(\langle x, y \rangle)$  many variable functions are included in our enumeration. The functions  $\varphi_i$  are also called *programs*.

We extend the enumeration  $(\varphi_i)_i$  to a *Complexity measure* by means of a sequence of *step counting functions*  $(\Phi_i)_i$ ; this sequence satisfies the two *Blum axioms*: for each  $i$ ,  $\mathcal{D}\varphi_i = \mathcal{D}\Phi_i$  and the relation  $\Phi_i(x) = y$  is decidable. Again we write  $\Phi_i^2(x, y)$  for  $\Phi_i(\langle x, y \rangle)$ .

A *transformation of programs*  $\sigma$  is a total recursive function operating on the indices of programs. In general these transformations are defined intensionally by implicit invocation of the s-n-m axiom and the universal machine axiom (possibly in combination with the recursion theorem) by writing a formula like

$$\varphi_{s(i)}(x) \Leftarrow P(i, x)$$

where  $P(i, x)$  denotes some expression recursive in  $i$  and  $x$ .

A *measured set* is a sequence of functions  $(\gamma_i)_i$  such that the predicate  $\gamma_i(x) = y$  is decidable. The sequence of run-times  $(\Phi_i)_i$  is an example. A transformation  $\tau$  such that  $(\varphi_{\tau(i)})_i$  is measured is called a *measured transformation of programs*.

For a (partial) function  $t$  we define:

$$F_t = \{\varphi_i \mid \forall_x^\infty [x \in \mathcal{D}t \Leftarrow \Phi_i(x) \leq t(x)]\}$$

$$C_t = \{f \mid \exists_i [f = \varphi_i \wedge \varphi_i \in F_t]\}$$

The *complexity class of functions*  $C_t$  contains all functions computed by programs in the *complexity class of programs*  $F_t$ . Note that in our definition both  $F_t$  and  $C_t$  contain partial functions even if the *name* of the class  $t$  is total.

In this definition complexity is measured in terms of the running time of the machines only. If we take into consideration that larger function values may require longer running times for being evaluated we arrive at the concept of *honesty*. *Honesty classes* have two-argument functions as names:

$$G_R = \{\varphi_i \mid \forall x [\varphi_i(x) \leq \infty \wedge \langle x, \varphi_i(x) \rangle \in \mathcal{DR} \Leftarrow \Phi_i(x) \leq R(x, \varphi_i(x))]\}$$

$$H_R = \{f \mid \exists i [f = \varphi_i \wedge \varphi_i \in G_R]\}$$

$G_R(H_R)$  is called a *Honesty class of programs (functions)*. Note that the condition enforced in the definition of  $G_R$  holds vacuously if  $\varphi_i(x)$  diverges; consequently each honesty class contains all functions with a finite domain, whereas it can be shown that no complexity class except for the trivial class  $C_\epsilon = \mathcal{P}$  contains any such function.

Special honesty classes with single variable names are obtained by considering honesty bounds  $R$  of the form  $R(x, y) = t(x)$ ; the resulting classes are called *weak complexity classes*:  $F_t^W = G_R$  and  $C_t^W = H_R$ . Note that  $C_t^W$  and  $C_t$  contain the same total functions. An alternative special type of honesty classes with single variable names is obtained by taking names  $R$  of the form  $R(x, y) = t(\max(x, y))$ ; these classes are called *modified honesty classes* in [9].

There exists a close connection between the notions of a measured set and a honesty class. By a well known theorem Mc Creight [4] every measured set is included in some honesty class with a total name; conversely every honesty class with a total name can be enumerated in such a way that the enumerating sequence represents a measured set. More formally:

**THEOREM 1 (MC CREIGHT & MEYER)** *If  $(\gamma_i)_i$  is a measured set then there exists a total function  $R$ , such that as a set of functions  $(\gamma_i)_i \subseteq H_R$ ; moreover an index for  $R$  can be obtained uniformly from an index for the decision predicate for  $\gamma_i(x) = y$ . Conversely, if  $R$  is a total function then  $H_R$  is enumerated by some measured set  $(\gamma_i)_i$  and indices for both the enumerating sequence and the decision predicate for  $\gamma_i(x) = y$  are obtained uniformly in the index of  $R$ .*

The above theorem has led to the feeling that the two concepts are more or less equivalent. This is certainly not the whole truth. The above equivalence is lost as soon as the name of the honesty class is partial. Moreover, it is not hard to construct a presentation of a honesty class with a total name such that this presentation as a sequence is not a measured set.

We now formulate the *naming theorem* of Mc Creight and Meyer [5] and our full *non-renameability result*:

**THEOREM 2 (NAMING THEOREM)** *There exists a measured transformation of programs  $\sigma$  such that for each  $i$  the classes  $F_{\varphi_i}$  and  $F_{\varphi_{\sigma(i)}}$  are equal (and consequently  $C_{\varphi_i} = C_{\varphi_{\sigma(i)}}$  as well).*

**THEOREM 3 (NON-RENAMEABILITY THEOREM)** *For every measured transformation  $\sigma$  there exists an index  $i$  of a total function such that  $H_{\varphi_i^2} \cap \mathcal{R} \neq H_{\varphi_{\sigma(i)}^2} \cap \mathcal{R}$ .*

This result resembles the results proven in [9]; however if inspected in more details all the results published in this paper are weaker: in theorem 6 the result

claimed reads  $H_{\varphi_i^2} \neq H_{\varphi_{\sigma(i)}^2}$ , i.e., the classes may turn out to be different due to the presence of partial functions in the classes; in theorem 7 there is shown a difference on the subclasses of total functions within the honesty classes, but the result is proven for the modified honesty classes only, i.e. the names are of a special form.

Expressions describing functions and/or transformations of programs in this paper are defined in terms of the hybrid language introduced in my thesis which combines elements from standard recursion theory and the (by now archaic) programming language ALGOL68. The resulting expressions may have in general several plausible computational interpretations which may differ with respect to convergence; the intended computational meaning is uniquely determined according to the guidelines as indicated in [8], section 1. The reader should keep in mind that according to this intended interpretation inequalities involving either a step-counting function or an element of some other measured set are evaluated using the decision predicate instead of a brute-force evaluation.

### 3 PROOF OF THE NON-RENAMEABILITY RESULT

The proof of the improved result uses the same technique used in the earlier results: we obtain a suitable version of the *mirror lemma*, which shows that a measured transformation  $\sigma$  eventually will “reflect” some name  $\varphi_e^2(x, y)$  with respect to some suitably large function  $R(x, y)$  in the sense that  $\varphi_e^2(x, y)$  is large compared to  $R(x, y)$  if and only if  $\varphi_{\sigma(e)}^2(x, y)$  is small; subsequently we show that the set of arguments where the reflected name is small supports the graph of a total diagonal function which is included in the honesty class with the original name but not in the transformed class. This diagonal then separates the original class from its renamed version.

We start with a function  $R$  which is sufficiently large in order that there exists an  $R$ -honest *odd-valued* function which is not **zero**<sup>2</sup>-honest. We define the transformation  $\alpha$  by:

$$\varphi_{\alpha(i,j)}^2(x, y) \Leftarrow \begin{array}{l} \text{if even } y \text{ then } \varphi_j^2(x, y) + R(x, y) + 1 \\ \quad \text{elif } \varphi_{\sigma(i)}^2(x, y) \leq R(x, y) \text{ then } \varphi_j^2(x, y) + R(x, y) + 1 \\ \quad \text{else } 0 \text{ fi} \end{array}$$

By the recursion theorem there exists a transformation  $\rho$  satisfying

CLAIM 1 (MIRROR LEMMA)

$$\varphi_{\rho(j)}^2(x, y) = \begin{array}{l} \text{if even } y \text{ then } \varphi_j^2(x, y) + R(x, y) + 1 \\ \quad \text{elif } \varphi_{\sigma(\rho(j))}^2(x, y) \leq R(x, y) \text{ then } \varphi_j^2(x, y) + R(x, y) + 1 \\ \quad \text{else } 0 \text{ fi} \end{array}$$

We next define (implicitly using the recursion theorem once again) the transformation  $\kappa$ :

$$\begin{aligned} \varphi_{\kappa(j)}(n) = & \text{if } n = 0 \text{ then } \mu m[\varphi_{\sigma(\rho(j))}^2(\pi_1 m, \pi_2 m) \leq R(\pi_1 m, \pi_2 m) \\ & \text{and odd } \pi_2 m] \\ & \text{else } \mu m[\text{odd } \pi_2 m \text{ and } \pi_1 m > \pi_1 \varphi_{\kappa(j)}(n-1) \text{ and} \\ & \varphi_{\sigma(\rho(j))}^2(\pi_1 m, \pi_2 m) \leq R(\pi_1 m, \pi_2 m)] \mathbf{f} \end{aligned}$$

Hence  $\varphi_{\kappa(j)}$  enumerates pairs  $\langle x, y \rangle$  with  $x$  increasing and  $y$  odd such that  $\varphi_{\sigma(\rho(j))}(\leq)R(x, y)$ . A “partial inverse” of  $\varphi_{\kappa(j)}$  is obtained by the transformation:

$$\varphi_{\beta(j)}(x) \Leftarrow \begin{aligned} & \text{if } \pi_1 \varphi_{\kappa(j)}(\mu n[\pi_1 \varphi_{\kappa(j)}(n) \geq x]) = x \\ & \text{then } \mu n[\pi_1 \varphi_{\kappa(j)}(n)] \text{ else false } \mathbf{f} \end{aligned}$$

The function  $\varphi_{\beta(j)}$  computes in fact a partial inverse to  $\pi_1 \varphi_{\kappa(j)}$ . If for some input  $x$  some pair  $\langle x, y \rangle$  is enumerated then  $\varphi_{\beta(j)}(x)$  yields the index of this pair in this enumeration; if no such pair is enumerated but if eventually some pair  $\langle x', y' \rangle$  is enumerated with  $x' > x$  the the value is **false**. Otherwise  $\varphi_{\beta(j)}$  is undefined.

Finally we define a diagonal transformation  $\varphi_{\delta(j)}$  by:

$$\begin{aligned} \varphi_{\delta(j)}(x) \Leftarrow & \text{if } \varphi_{\beta(j)}(x) = \text{false then } 2\Phi_{\beta(j)}(x) \\ & \text{elif } \Phi_{\pi_1 \varphi_{\beta(j)}(x)}(x) \leq R(x, \pi_2 \varphi_{\kappa(j)}(\varphi_{\beta(j)}(x))) \\ & \text{and } \varphi_{\pi_1 \varphi_{\beta(j)}(x)}(x) = \pi_2 \varphi_{\kappa(j)}(\varphi_{\beta(j)}(x)) \\ & \text{then } \pi_2 \varphi_{\kappa(j)}(\varphi_{\beta(j)}(x)) - 1 \text{ else } \pi_2 \varphi_{\kappa(j)}(\varphi_{\beta(j)}(x)) \mathbf{f} \end{aligned}$$

Informally, in order to evaluate  $\varphi_{\delta(j)}(x)$  one first must evaluate  $\varphi_{\beta(j)}(x)$ . If this computation diverges then  $\varphi_{\delta(j)}(x)$  is undefined. If the computation converges but yields the value **false** then output twice the time it has taken to compute this value **false**. Otherwise we diagonalize: we know that for some value  $y$  a pair  $\langle x, y \rangle$  is enumerated by  $\varphi_{\kappa(j)}$ , say  $\varphi_{\kappa(j)}(m) = \langle x, y \rangle$ . Test whether  $\Phi_{\pi_1 m}(x) \leq R(x, y)$  and if so whether  $\varphi_{\pi_1 m}(x) = y$ ; if both conditions are satisfied then output  $y - 1$  and output  $y$  otherwise.

Note that this computation diverges when  $\varphi_{\beta(j)}(x)$  diverges, and this will only happen if no pair  $\langle x', y' \rangle$  with  $x' > x$  is enumerated by  $\varphi_{\kappa(j)}$ , i.e., when  $\varphi_{\kappa(j)}$  is partial. Hence in case  $\varphi_{\kappa(j)}$  is total then so is  $\varphi_{\delta(j)}$ .

**CLAIM 2** *The sequence  $(\varphi_{\delta(j)})_j$  is a measured set.*

This can be seen as follows.

For a given pair  $\langle x, y \rangle$  it can be decided whether  $\langle x, y \rangle \in \mathfrak{R}\varphi_{\kappa(j)}$ : if  $y$  is even or if  $\varphi_{\sigma(\rho(j))}^2(x, y) > R(x, y)$  then  $\langle x, y \rangle$  is no candidate for being enumerated so we can answer “no”. Otherwise we know that some pair  $\langle x', y' \rangle$  with  $x' \geq x$  will eventually be enumerated and we can wait and see whether  $\langle x, y \rangle$  is enumerated by that time.

Using this observation we can describe the following decision procedure for  $\varphi_{\delta(j)}(x) = y$ ?

If  $y$  is even then test whether  $\Phi_{\beta(j)}(x) = y/2$  and  $\varphi_{\beta(j)}(x) = \mathbf{false}$ ; if so the answer is “yes”. Otherwise test whether  $\langle x, y + 1 \rangle \in \mathfrak{R}\varphi_{\kappa(j)}$ ; if not then the answer is “no”. If  $\langle x, y + 1 \rangle = \varphi_{\kappa(j)}(m)$  test whether  $\Phi_{\pi_1 m}(x) \leq R(x, y + 1)$  and  $\varphi_{\pi_1 m}(x) = y + 1$ ; if so the answer is “yes” and otherwise the answer is “no”.

If  $y$  is odd then test directly whether  $\langle x, y \rangle \in \mathfrak{R}\varphi_{\kappa(j)}$ . If not the answer is “no”. Otherwise let  $m$  be the argument such that  $\langle x, y \rangle = \varphi_{\kappa(j)}(m)$ , and test whether  $\Phi_{\pi_1 m}(x) \leq R(x, y)$  and  $\varphi_{\pi_1 m}(x) = y$ ; if so the answer is “no” and otherwise the answer is “yes”.

The correctness proof for this decision procedure is left to the reader.

Our next claim holds only in the case that  $\varphi_{\kappa(j)}$  is a total function, i.e.,  
 $\exists x \exists y [\varphi_{\sigma(\rho(j))}^2(x, y) \leq R(x, y)]:$

**CLAIM 3** *If  $\varphi_{\kappa(j)}$  is total then  $\varphi_{\delta(j)} \notin H_{\varphi_{\sigma(\rho(j))}^2}$ .*

Consider an index  $i$  for  $\varphi_{\delta(j)}$  and a value  $m$  with  $\pi_1 m = i$ . Let  $\varphi_{\kappa(j)}(m) = \langle x, y \rangle$  then we have for this particular argument  $x$ :

$$\begin{aligned} \varphi_i(x) = \varphi_{\delta(j)}(x) = & \mathbf{if } m = \mathbf{false} \mathbf{ then } 2\Phi_{\beta(j)}(x) \\ & \mathbf{elif } \Phi_i(x) \leq R(x, y) \mathbf{ and } \varphi_i(x) = y \mathbf{ then } y - 1 \\ & \mathbf{else } y \mathbf{ fi} \end{aligned}$$

The first condition is evidently false; since the then-part for the second condition is contradictory we conclude that  $\varphi_i(x) = y$  and  $\Phi_i(x) > R(x, y)$ ; since also for this pair  $\langle x, y \rangle$  it holds that  $\varphi_{\sigma(\rho(j))}^2(x, y) \leq R(x, y)$  this shows that  $\varphi_i$  violates the honesty condition at  $\langle x, y \rangle$ . From the fact that  $\varphi_{\kappa(j)}$  is total we infer that there exist infinitely many violations of this type; since also  $i$  was an arbitrary index for  $\varphi_{\delta(j)}$  this proves our claim.

**CLAIM 4** *For every pair  $\langle x, y \rangle$  such that  $\varphi_{\delta(j)}(x) = y$  one has  $\varphi_{\rho(j)}^2(x, y) \geq \varphi_j^2(x, y)$ .*

For even  $y$  this claim is a direct consequence of the definition of  $\rho$ , whereas for odd  $y$  the definition of  $\delta$  implies that  $\langle x, y \rangle$  is a pair enumerated by  $\varphi_{\kappa(j)}$  and therefore the condition  $\varphi_{\sigma(\rho(j))}^2(x, y) \leq R(x, y)$  is satisfied. However, according to our use of the mirror lemma this means that  $\varphi_{\rho(j)}^2(x, y) = R(x, y) + \varphi_j^2(x, y) + 1 \geq \varphi_j^2(x, y)$ .

The theorem now can be derived using the above claims.

Since  $(\varphi_{\delta(j)})_j$  is a measured set there exists an index  $j_0$  of some total function  $\varphi_{j_0}^2$  such that  $(\varphi_{\delta(j)})_j \subseteq H_{\varphi_{j_0}^2}$ .

If for this index  $j_0$  the function  $\varphi_{\kappa(j_0)}^2$  is total then  $\varphi_{\delta(j_0)}^2$  is a total function in  $H_{\varphi_{\rho(j_0)}^2} \setminus H_{\varphi_{\sigma(\rho(j_0))}^2}$ .

In the alternative case that  $\varphi_{\delta(j_0)}^2$  is a partial function then for almost all  $x$  it holds that  $\varphi_{\rho(j_0)}^2(x, y) = 0$  for all odd values of  $y$ . So the odd-valued functions in  $H_{\varphi_{\rho(j_0)}^2}$  are **zero**<sup>2</sup>-honest functions. By the mirror lemma it follows that for almost all  $x$  one has  $\varphi_{\sigma(\rho(j_0))}^2(x, y) \geq R(x, y)$  for all odd values of  $y$ . Since there exists by assumption an odd-valued  $R$ -honest function  $f$  which is not **zero**<sup>2</sup>-honest, one concludes that  $f \in H_{\varphi_{\sigma(\rho(j_0))}^2} \setminus H_{\varphi_{\rho(j_0)}^2}$ .

Having shown that in both cases the honesty classes are different, the proof is complete.

#### 4 LOOKING BACKWARDS

With hindsight one may ask why this result is not included in the earlier presentations of the non-renameability theorem. There is just one additional technique involved in the proof which was not present in the proofs in [9]: the use of parity. The problem in the earlier proofs is how to obtain “escape values” for the diagonalization, in such a way that the choice for this escape value won’t lead to a violation against the original honesty bound. The earliest proof of the non-renameability uses the undefined escape value, since this choice will never violate any honesty condition. The consequence is that the diagonal function becomes partial.

The question whether the non-renameability result extends to the case that only the total functions in a honesty class are considered originates with Albert Meyer. Evidently, considering the simple case of the weak complexity classes (which are non-renameable if partial functions are considered; see theorem 4 in [9]), will yield no answer since the weak and the strong complexity classes contain the same total functions, and the strong classes can be renamed. Thus the need for finite escape values arose.

In order that the choice of the escape value  $y$  does not lead to a violation of a honesty condition at argument  $x$ , the pair  $\langle x, y \rangle$  should be located at a place where the original bound  $S$  is large. If the transformed bound  $S'$  is obtained using the mirror lemma, then these places can be detected by deciding whether  $S'(x, y)$  is small; however, since existence of such a value  $y$  is in general undecidable finding one may be too hard. Only because of the special structure of the names for the modified complexity classes this hurdle could be overcome.

Using the parity of the  $y$  value as a dividing condition our new proof in fact constructs bounds  $S$  and  $S'$  where the mirror effect only is enforced on half of the plane (i.e., for odd values of  $y$  only). The diagonal tries to produce violations against the honesty bound  $S'(x, y)$  for odd values of  $y$  for which  $S(x, y)$  is large and  $S'(x, y)$  is consequently small. The escape value is chosen to be even. By a standard combining lemma argument the complexity of this diagonalization can be estimated, and it suffices to choose  $S(x, y)$  being sufficiently large for even  $y$  and pairs  $\langle x, y \rangle$  where  $S$  should be large.

Ultimately there are two cases; either the diagonalization succeeds and a member of  $H_S \setminus H_{S'}$  is obtained or  $S'$  becomes so small that some odd-valued

member of  $H_S$  gets excluded. Evidently this idea does not reach far beyond the original techniques, so the result could have been obtained already in 1973 with the others.

A more interesting question is whether the whole field of Abstract Complexity Theory should be looked at at all at this stage in the development of theoretical computer science. The subject disappeared from the battlefields of theoretical computer science since the axioms of the theory failed to put any constraint of naturalness on the models; all sort of pathologies were possible, and any attempt to further constrain the theory by enforcing naturalness conditions was doomed to failure [3]. Also the theory failed to provide any insight in the core problems of the field: the relation between time and space and the power of nondeterminism.

I claim however that some sort of a positive revival today is possible; the gap between recursion theory and complexity theory is being narrowed these years, both because of the nowadays frequent use of recursion theoretical techniques in structural complexity theory, but also since researchers in recursion theory once more become interested in complexity issues. So there still might be a market for the lost textbook on Abstract Complexity Theory.

#### REFERENCES

1. Bemmison, V.L., *On the computational complexity of recursively enumerable sets*, ph.d. thesis, Univ. of Chicago, 1976.
2. Blum, M., *A machine-independent theory of the complexity of recursive functions*, J. Assoc. Comput. Mach. 14 (1967) 322–336.
3. Lischke, G., *Erhaltungssätze in der Theorie der Blumschen Komplexitätsmaße*, ph.d. thesis, Fr. Schiller Univ. Jena, 1976.
4. Mc Creight, E.M., *Classes of computable functions defined by bounds on computation*, ph.d. thesis, Carnegie Mellon Univ., 1969.
5. Mc Creight, E.M. & Meyer, A., *Classes of computable functions defined by bounds on computation*, Proc. SIGACT STOC 1, 1969, 79–88.
6. Odifreddi, P., *Classical Recursion Theory, part 1*, North-Holland, Studies in Logic and the Foundations of Mathematics, vol. 125, 1989; part 2, to appear.
7. Rogers, H., jr., *Gödel numbering of partial recursive functions*, J.S.L. 23 (1958) 331–341.
8. van Emde Boas, P., *Abstract Resource Bound Classes*, ph.d. thesis, Univ. of Amsterdam, 1974.
9. van Emde Boas, P., *The non-renameability of Honesty Classes*, Computing 14 (1975) 183–193.
10. van Emde Boas, P., *Some applications of the Mc Creight-Meyer algorithms in Abstract Complexity Theory*, Theor. Computer Science 7 (1978) 79–98.





# Mathematics as the Paradigm for Metaphysics

Louk Fleischhacker<sup>1</sup>

Twente University Enschede The Netherlands

Dedicated to Prof. Cor Baayen, who retires as the scientific director of CWI, and who can therefore revive his interest in philosophy.

The ideal of mathematical exactness is strongly paradigmatic for modern science, for which mathematics practically functions as a metaphysical foundation. This strongly influenced philosophy. In our century, however, critical voices arise, even from the ranks of scientists. Reflection on the foundations of mathematics has produced a deeper insight into its nature. As a result the tendency to judge content by structure has become less pre dominant. Metaphysics, however, is still often rejected as only producing constructions with unjustified claims to necessity. Clear recognition of the role of the mathematical paradigm shows that this rejection is unnecessary.

## MATHEMATICS AND METAPHYSICS

It is sometimes claimed as an advantage, and sometimes regretted, that modern natural science has no metaphysical foundation. The unconventional thesis might, however, be defended that *mathematics* has effectively functioned as the metaphysical foundation of the modern scientific tradition. The still living fundamental principle of science, from Galileo onward, is the reduction of qualitative phenomena to measurable quantities and structures. Many underlying forms of thought in which this principle has been active apart from actual mathematisation, such as the mechanistic view, determinism, and positivism, have been superseded by others such as complementarity, probabilism, and chaos-theory in science itself, and critical rationalism and even sociologism in the philosophy of science. But the idea that knowledge is scientific in the complete sense of the word only if it is expressible in mathematical structures and equations seems to be unchallenged. Seemingly extreme reactions to the mathematical perspective, such as holism, implicitly presuppose the same mathematical models as their more positivistic counterparts. This probably accounts for their apparent extremeness. Even if real mathematisation lies far behind the horizon, as it does in the cognitive sciences, it is nevertheless taken as a standard, e.g. in the form of computational models. In logic and linguistics, and even in ethics, the mathematical perspective is prevailing now. What is often called 'formalisation' or 'formal methods' by analogy to mathematical logic, is in fact the construction of mathematical models, as it originally was in mathematical logic too.<sup>2</sup>

---

1) This article contains parts of the introduction and Chapter 5 from: L.E.Fleischhacker, *Beyond Structure*, Peter Lang, Frankfurt 1995

2) For a coherent and convincing criticism of this trend, see: Sören Stenlund, *Language and Philosophical Problems*, Routledge London, 1990.

The paradigm of mathematical thought has also thoroughly invaded philosophy. Not only by attempts to systemize the discipline *more geometrico*, such as Spinoza's *Ethica*, but also by the ideal of demarcating a domain of pure rationality from the ambiguities and prejudices of common sense. From Descartes to Wittgenstein this ideal has exerted a strong influence on philosophy, and the consequence has been an estrangement of philosophy from its most fundamental discipline: metaphysics.

#### THE PROBLEM OF MATHEMATICAL THOUGHT

From the perspective of the philosophy of mathematical thought, the relationship of mathematical structure to observable reality has remained extremely problematic. Plato formulated the question where in the world to look for numbers and geometrical figures,<sup>3</sup> and he concluded that the visible world is not the only possible mode of being. Mathematics cannot be about the world of human experience, for example, because this world resists reduction to purely mathematical structure. The reality of change especially is a hard nut to crack, as was noticed already by Aristotle. But he found a way different from Plato's for dealing with mathematical objects. He regarded them as the results of abstraction, the actualization *in thought* of a principle we find in the world of experience. He called this principle 'ὄλη νοητή: intelligible matter.

In antiquity the main philosophical problem with mathematical objectivity was to *separate* it from experience, without making the applicability of mathematics impossible to understand. Modern times, however, begin with the idea of the *identity* of mathematics and physics. Nature herself is thought to be structural, and thus accessible to mathematical investigation, not only by her external geometrical shapes - as Archimedes had already discovered - but also in her inner laws.

In philosophy this had a very strong impact. Descartes characterises the world of experience as *res extensa*, taking what in Aristotelianism had only been an outer property of material things to be their essence. The externality of nature becomes its inner principle. In the nineteenth century Hegel formulated the essence of nature as 'the Idea in the form of externality to itself'.

For philosophy this meant that the problem was no longer one of the relation of the mathematical to the physical, but of the relationship of a knowing subject, Descartes' *res cogitans*, to an objective world which is mathematical and physical at the same time. This produced strongly mathematically coloured, but never really mathematical, metaphysics.

Spinoza's attempt to construct metaphysics '*more geometrico*' has led to points of view which actually went beyond mathematical reasoning, but remained strongly influenced by it. Even when in modern philosophy the paradigm of geometry, or mathematics in general, is explicitly rejected, as in the case of Hegel's system, the lure of structural rigour is still present as can be seen in his rigorously systematic approach.

In the nineteenth century the identification of mathematical and physical objectivity became less and less obvious. Mathematics, liberated from its close connec-

---

3) Plato, Republic 526a

tion to physics and technology, began to develop highly speculative theories such as complex number theory, abstract algebra, Fourier analysis, non-Euclidean geometry and projective geometry. It started looking for a foundation of its own, independently of physics, and thereby more and more overtly showed its ideal character.

In philosophy, on the eve of the twentieth century, two - apparently opposite - impulses emerge, which may eventually undermine the ideal of mathematical rigour: Husserl's phenomenology, which introduces another ideal of philosophical rigour, and Frege's mathematical logic, which *objectifies* mathematical reasoning.

Gödel's results teach us that, as a consequence of this objectification,<sup>4</sup> the foundation of mathematics cannot be formulated explicitly as a mathematical theory. Mathematical thought as such cannot be free from intuitive presuppositions demanding investigation by a discipline other than mathematics itself.

For a 'working mathematician' this is no problem at all. She or he is perfectly happy with Hilary Putnam's '*Yes, we have no foundations*'; but the philosopher experiences a change of problem-field again. Now *both* the subject-object relationship *and* the relationship of structure and reality have become problematic. The mathematical point of view appears to be based on an intuitive insight, constituting a certain perspective - which I call *structural* - on the world of experience.

But if that is true, mathematical structure is not necessarily the only or even the most adequate form in which scientific knowledge can be expressed. Perhaps the success of measuring-science has blinded us to metaphysical perspectives, whether or not they justify or radicalise the mathematical approach. Even philosophies which are generally considered to be anti-mathematical, such as Hegel's speculative dialectics or Heidegger's existential philosophy, when inspected more closely, appear to share certain essential presuppositions with the mathematical approach, e.g. the denial of real potentialities. In fact the ideal of 'exactitude' - the possibility of making all presuppositions explicit and developing a body of thought consistently from them - seems to be all-pervading in our culture. Wittgenstein's *Philosophical Investigations*, because of its anti-systematic tendency, may be regarded as an outstanding exception; but this work also shows clearly the kind of trouble that arises if one tries to leave the mathematical paradigm behind. For what other method than allusion remains, if an explicit development of ideas is forbidden? The very different ways in which

---

4) The words 'subject' and 'object' are used in different senses, but the tendency always is that they are correlatives in the performance of some (theoretical or practical) action - as the linguistic use suggests. The subject is the active pole, the object not necessarily passive, but the action is always directed towards it. Subjective is what belongs to the subject as such (i.e. in its function of being the active pole), objective what belongs to the object as such (i.e. in its function of being the 'aim' of the activity), which does not necessarily mean that it exists independently of the subject. Mathematical objects for instance, need not be conceived of as existing independently of mathematical thought. Subjectivity and objectivity are the properties of being subjective, respectively objective. Objectification is the act of giving objectivity to some content, either by theory - conceiving of it in the form of objectivity - or by practice - bringing about a state of affairs which may be understood as representing the said content in an objective form.

Wittgenstein's philosophy has been interpreted make this clear, for if one cannot express one's ideas explicitly, there is no limit to interpretation.

Also in structuralism, in spite of its name, a tendency to leave mathematical grounds is present. It is *real* structure the structuralists are after, not ideal, mathematical structure. But as long as *nothing but* structure is seen, it is already surreptitiously being idealised. Therefore, in structuralism there is always an essentially non-structural principle - such as power, force or spontaneity - lurking in the background. Critics of the mathematical point of view usually underestimate its power. Either it eventually turns out that they have remained within it or they adopt its abstract opposite and in this way remain indebted to it.

#### THE AGE OF MATHEMATISM

Dijksterhuis concludes his *Mechanisation of the World Picture* with the remark:

The mechanisation, which the world-picture underwent in the transition from antique to modern natural science, consisted in the introduction of a description of nature by means of the mathematical concepts of classical mechanics; it indicates the beginning of the mathematisation of natural science, which obtains its completion in twentieth century physics.

But this is only seen from the direction of the ultimate *effect*. In my view, the technical as well as the philosophical sources of the rise of modern science already introduced very strong tendencies towards mathematisation. It is in accordance with the natural development of technology that technical concepts are made more and more explicit. Of course this does not explain that this development took place in this particular historical period. But one thing is clear: in order to make technical concepts explicit, one must measure and calculate. Moreover, on the philosophical side, medieval Aristotelianism hardly left room<sup>5</sup> for another basis to be criticised on than precisely the mathematical Platonism that arose in the Renaissance. The breakthrough of both tendencies - the technological development and mathematical Platonism - and their fruitful meeting in a particular place and time can probably be explained by fundamental changes in society.<sup>6</sup> What is important here, however, is the result of the breakthrough: the firm belief that measurement and mathematical calculation, and nothing else, will lead to insight into the phenomena of nature. For Galileo the book of nature was written in mathematical signs, and for Newton mathematical space and time were absolute, whereas experienced space and time were considered to be only

---

5) Especially for those who - like Cusanus and the humanists, and unlike most of the modern philosophers - knew perfectly well what it was all about, and where the strength and weakness of this world-picture was to be located.

6) Scheler Max, *Die Wissensformen und die Gesellschaft*, Bern 1969.

relative. Nature came to be seen as mathematical *in itself*, and the distinction between mathematics and physics became obsolete. In the eighteenth century 'mathematics' still encompassed a whole range of disciplines, from arithmetic to machine-construction. Only in the nineteenth century did a new form of 'pure' mathematics emancipate itself from natural science and technology. But by then the mathematical style of thinking had been thoroughly spread among scientists and technicians.

#### WHAT IS MATHEMATISM?

But the prevalence of a particular style does not itself constitute mathematism, which is rather a - usually implicit - metaphysical position connected to the feeling that the 'mathematical' style is so self-evident that it does not need any foundation. In this way this style is itself taken as the foundation of science and philosophy. As a consequence, the objectivity and generality of the style have to be regarded as objectivity and generality without qualification. The object of mathematical thought can be characterised as *structure*, which is more general than what is usually understood by quantity, but is by no means identical to metaphysical universality or being. If unqualified objectivity is identified with mathematical objectivity, the fundamental nature of reality becomes structure, which is differentiated only by higher or lower degrees of complexity. This is exactly in line with the philosophy, ascribed to Pythagoras, according to which the essence of the universe is *number*. Number for the ancients was the principle of what is mathematical, and it is still often regarded as a fundamental paradigm of structure.<sup>7</sup> The Pythagorean world view is a fundamental and ever recurring metaphysical perspective. In Plato's Academy, Speusippos and Xenocrates took up this line of thought and in the Renaissance it was popular with humanists such as Pico della Mirandola. Even today it is explicitly adhered to by some theoretical physicists, who doubt whether 'matter' is to be regarded as a useful concept in physics.

On the other hand we all know that structure is not something immediately given. We can see different structures in one and the same phenomenon and we can technically give different structures to our surroundings. And in pure mathematics, structure is the result of postulation or thought-construction. So structure is in a certain sense our product. It is the *structurability* of the world, which is fundamental.

So mathematism has two sides to it, expressible in two ideal-typical theses:

1. Structurability is the *essence* of everything.
2. To know something is the same as to give it structure.

This is a completely coherent metaphysical position, in which being is identified with *mathematical* intelligibility, instead of intelligibility without qualifications. But the question is, whether or not this world view is unduly *restrictive*. Does it rule

---

7) In this connection Kronecker's saying: 'The natural numbers are made by the Lord, the rest [of mathematics] is human work' is usually quoted.

out any other perspectives which we find particularly plausible? One could ask whether anything exists which is not - in a certain sense - structurable, and it would be difficult to find an example. On the other hand, one could ask whether in fact there exists anything the essence of which *is* its structurability. Perhaps one could think that the essence of *space* is its structurability. But once one imagines something *in* it, a non-structural quality is introduced, which distinguishes the space occupied by the 'something' from empty space. Trying to reduce this quality to structure again, could very well lead to an infinite regress. If indeed it is felt as absurd from the point of view of common sense to express mathematism as an explicit philosophy - in the same way as it is felt as absurd to express scepticism<sup>8</sup> as an explicit position - what then are the grounds for this feeling of absurdity?

Let me compare this situation with the current aporia in debates about the scope of artificial intelligence. If one mentions a human skill, not yet simulable by computer programs, the AI defender will say: if you describe it exactly and clearly (i.e. mathematically) to me, I shall find a way to simulate it, and if you cannot describe it in this way, then it is nothing at all. But then, if it is so described, it is probably not the same as it was before. What, however, is the *difference*? We have the feeling that, as soon as we *describe* this difference, a corresponding correction of the program will eliminate it.

We are so immersed in mathematism that we simply cannot imagine a kind of exactness *surpassing* mathematical exactness. For how could we prove that e.g. intelligence is *not* reconstructible in mathematical terms, if not by using a description of mathematical reconstructibility itself, showing its restrictions. But such a description should evidently be clearer and more self-evident than any mathematical construction. In a traditional philosophical framework metaphysics could perform this task, and that is why mathematics and metaphysics must be rivals in a mathematistic world.

On the other hand, it seems to be precisely the development of information-technology that tends to change this situation. In this field structures are of course important, but they can no longer be considered as purely mathematical. They are not invented for the sake of clarifying the domain of the ideally structural or the inner laws of nature, but for the sake of their use in a context of human practice. In the perspective of pure mathematics they are clumsy and opportunistic. They have nothing of the proverbial mathematical elegance, their adequacy cannot be rigorously proven and their functioning cannot be completely tested.

Mathematicians as well as metaphysicians stand here awkwardly looking at something of which they claim to know the principles, but to which they cannot apply them. The two may become brothers again. But before this new brotherhood is celebrated, it is

---

8) Scepticism disregards its own claim for truth. Therefore it is immediately refutable by showing its 'pragmatic' self-contradiction. But even then it is not refuted as a general *attitude* in life. Hegel saw this clearly in the chapter on scepticism in the *Phenomenology of Spirit*. (Hegel G.W.F., *Phaenomenologie des Geistes*, ed. Hoffmeister, Felix Meiner, Hamburg 1952 p.52; Miller A.V. (Transl.), *Hegel's Phenomenology of Spirit*, Oxford 1977.) Cf. also Michael N. Foster, *Hegel and Scepticism*, Harvard 1989



advisable to analyze the past period of rivalry, in which victory seemed to dwell on the mathematical side.

#### METAPHYSICS IN A MATHEMATICAL STYLE, AND ITS FATE

Mathematical abstraction results in a certain structure, which is essentially one of the specific realizations of the structurability of a field of experience, and therefore it is contingent. Mathematical structure is grasped by - ideal or real - actualization of the potency of all things of our experience to be divided in thought into interrelated parts. This actualization essentially includes arbitrariness, and can in that sense be called a *construction*, although it may very well be a *reconstruction* of a known phenomenon. Philosophical reflection on the other hand aims at necessity, for the coherence of its objects - which I shall call *principles* - cannot depend upon tradition, convention or postulation. Any blending of mathematical and philosophical reflection bears the suggestion that there exist *necessary constructions*, which is a *contradictio in adjecto*. So if metaphysics is implicitly contaminated with such a blending, it is an easy prey to criticism depicting it as either absurd or obscure. A construction has definite inner relationships, definite elements and definite properties. All these are definite, because they have been *defined* to be such as they are, and this means that there is arbitrariness in them. Principles nor their relationships, on the other hand, can be understood as the result of definition, they must on the contrary be *presupposed* in definitions. They constitute the perspectives in which we can try to conceptualize or reconstruct experience. Their relationships are beyond definition, because they are constitutive for the meaning of a definition.<sup>9</sup> Nevertheless, in their implicit form, these relationships are better known than explicitly defined structures. They are implicitly but effectively known to us, and attempts to express them explicitly are experienced as highly artificial. They are not axioms, nor 'necessary truths,' nor expressible in a judgement or theorem without already presupposing them. We can investigate them, but we can never use them, apply them or draw conclusions from them *outside* the perspective they constitute. Yet, if we want to investigate principles, we must somehow express the results of this investigation. This is where the difficulties begin, for how to express such results in a form which must necessarily be determined either by tradition or by construction? Philosophy seems to hesitate continuously in its form of expression between mathematics and literature.

Literature is suggestive to us on the basis of culture and tradition. It can express truth, it can make one think, and it can point towards insights into necessary connections. But it lacks liability to critical investigation of its evidence. It either convinces or does not, but in the latter case one can rarely lay ones finger on the spot.

---

9) If an axiomatic theory is e.g. understood as a definition of a particular kind of structure (not of a definite structure, for all interesting theories are non-categorical), this presupposes the consistency of the theory. But it follows from Gödel's well known results that this consistency requires a *stronger* theory to be proven. The real reason why we believe in the consistency of the theory is, that we believe we already understand nature of the kind of structure it is meant to deal with. And we believe this, because we are thoroughly convinced of the applicability of the principle of structurability to a certain field of experience. Therefore, the principle of structurability is a presupposition of any mathematical theory.

Reducing philosophical prose to 'literary text' means depriving it of its real ambition: expressing intelligible necessity as such.

Mathematics, on the other hand, owes its intellectual force and its certainty to the systematic representation of its objects. In its various forms of representation there exists a structural relationship between the intended mathematical objects and the way they are expressed. This specifically mathematical relationship of sign and meaning is not necessary in a strict sense, but it characterizes a mathematical discipline so strongly, that *within* the discipline it appears as necessary. Geometry without figures and algebra without formulas is not impossible, and in some periods of the development of these disciplines purely linguistic expression was even normal, but, as Leibnitz observes, it is very hard in this way to travel a long distance without getting exhausted.<sup>10</sup>

Philosophical systematization, however, cannot aim at representing certain structures in such a rigorous way. It has to transcend its own particular structure, not into a literary expressive imagination, but into the intellectual challenge of its proper aim: establishing real insight. Such systematization has the function to prevent thought from stopping at too low a level of understanding, it provides the formulations of the problems, but it is never itself a solution. Philosophy is the encouragement of the intellect to recognize that it knows more than it thinks it knows. Participating in philosophical thought always requires that we give up some prejudice concerning what we imagine to be the definition of 'knowing.' This distinguishes the intellectual challenge in philosophy from its mathematical counterpart. In mathematics the challenge is directed towards the faculty of imagining of and reasoning about new and unheard-of structures. The principle of 'knowing' in mathematical reflection, however, always remains the same: structurability. In philosophy there is no fixed principle of knowing, only the attempt at explicitly knowing the principles guiding all of our knowledge. The 'exactness' of philosophical expression, therefore, is of a negative nature. Its function is to prevent a premature feeling of understanding. All beginning students of philosophy complain about this. They justly feel that philosophical language aims at making things more difficult instead of easier. Why cannot this be said in a more simple way? In a certain sense this resembles the situation in mathematics. There too things are said in a complicated form. But one feels that the reason for it is, that they *are* really complicated. In philosophy, however, anyone who has feeling for what it is all about, becomes convinced that understanding the complicated cannot be the ultimate aim here. Principles must be simple, and it is because of their simplicity that it is difficult to grasp them. So why cannot simple things be expressed in a simple way? The answer of course is, that simple expressions suggest to the untrained the wrong kind of simplicity of the content. In 'occult disciplines' of certain religious societies this is no problem, because the expressions

---

10) "Sans cela nostre esprit ne sçauroit faire un long chemin sans s'égarer" [Without that the mind could not go a long way without getting exhausted] G.W. Leibnitz in a letter to Galloys from 1677. In: G.W. Leibnitz, *Die philosophischen Schriften*, hrsg. von C.I. Gerhardt, Hildesheim, 1965.



are only meant for the initiated, who are supposed to understand them properly. Philosophy, being a *rational* discipline, must necessarily provide its own initiation. It cannot separate a cult of initiation from the expression of its actual contents. In a rational discipline one becomes initiated because one takes up the intellectual challenge, one understands what is interesting about it, whereas in initiation rites one participates not in the first place because one understands what they are good for, but because someone with authority says they must be undergone in order to understand what they were good for afterwards.

Mathematical and philosophical *expression* have, as we now understand, diametrically opposed criteria of adequacy. Mathematical expression is better, in the measure in which it allows us to connect mathematically the structure of our language with the structures expressed in that language. The more rigorous this connection becomes, the more our way of expression gains the character of a formalism useful for accurate proof and computation. Philosophical expression, on the other hand, is better in the measure in which it prevents the intellect from clinging to certain definite structures of knowledge and self-expression. Mathematical language should enable us to concentrate on definite structures, philosophical language should prevent such concentration with the aim of opening up our minds for the origin of our perspectives without presupposing any initiation into extraordinary realms of experience.

For this reason any attempt to develop metaphysics following the mathematical paradigm must necessarily end in the fundamental rejection of all forms of metaphysics. If both disciplines start to claim a common domain, they become rivals, and if this common domain is structurability, mathematics is in for a glorious victory. The dilemma between the mathematical and the philosophical criterion of adequacy of expression is unsolvable. There is no dialectical solution either, because to choose for dialectics is already to choose for the philosophical criterion. As soon as it is tried to understand dialectics as a formalism in the sense of mathematical logic, complete rejection of it is not far behind.

On the one hand to choose the mathematical criterion for philosophy, leads to nihilism. If, on the other hand, mathematical reflection acquires metaphysical pretensions, it cannot very long remain content to be pure mathematics. It has to incorporate some philosophical reflection, and in the measure in which it succeeds in expressing this incorporation explicitly, it disqualifies itself as mathematics. In the measure, however, in which it succeeds in satisfying the requirements of mathematical expression, it becomes philosophically irrelevant. In its naive form it becomes dogmatic because it postulates some more or less arbitrary constraining framework, which nevertheless is infected by the metaphysical claim that it expresses necessity. In reaction to this dogmatism it then becomes nihilistic, for the arbitrary character of the construction is brought to the foreground. It will be insisted then that 'anything goes.' In this case, 'anything' is not really anything of course, but any *construction*,<sup>11</sup> and that is not what we are looking for in metaphysics. Therefore this trail leads us into nothingness.

---

11) Cf. B. Taureck, *Das Schicksal der philosophischen Konstruktion*, Wien, 1975.

In fact such a development has taken place in contemporary philosophy. The period in the history of philosophy which is currently labelled 'modern philosophy' ends with Hegel's famous system. The tension between systematic rigour on the one hand, and the aim of expressing the openness of the human mind and the dynamical character of thought on the other, is still reconciled here. The dogmatic and nihilistic side are still held together in the truly philosophical conception of the absolute idea. But the suggestion is very strong that the absolute idea, binding together all principles like a 'one ring'<sup>12</sup> is not only meant to be completely intelligible to us, but also to justify the specific structure of Hegel's system. Yet this suggestion is somewhat misleading, because Hegel himself never hesitated to make additions and corrections. The problem is, that in its systematic structure, there is no place for expressing this openness *with respect to the structure itself*. This is a curious paradox: the system aims at the expression of the transcendental openness of the human mind, that is its ability to grasp transcendental principles by intellectual perception; yet it is not able to express this openness with respect to the philosophical method by which it is composed! The system, therefore, still has some traits of the 'necessary constructions' of modern philosophy. This is precisely the impossible contamination of mathematics and metaphysics which tends to discredit all modern metaphysical positions,<sup>13</sup> and which seems to be the basis of the widespread present consensus on the impossibility of philosophical systems.

But the '*faute hypercorrecte*' is as usual in philosophy as it is in practice. Because it has not become clear that it is the mathematical paradigm which still constituted the trap of Hegel's system, philosophical positions opposing to German idealism or to modern philosophy in general, such as Marxism, vitalism, existentialism, positivism and structuralism, however critical they are with respect to the modern tradition, are by no means free from this same paradigm. Essentially they all switch to and fro between the dogmatic form, which suggests a necessary construction, and the voluntaristic form, which essentially expresses the abstract notion of a freedom which is only limited by the consequences of its own decisions, such as only the

---

12) See J.R.R. Tolkien, *The Fellowship of the Ring*.

13) Heidegger's notion of '*Seinsvergessenheit*' can be interpreted as a philosophical expression of this confusion. It is coined to criticise ontological fixation of the opposition of subject and object. Such a fixation is a characteristic of mathematical reflection. It seems to be this mathematical element in modern metaphysics, which falls under Heidegger's criticism, and that makes it also clear why he understands modern technology as the realization of such metaphysics. The '*Verdinglichung des Seins*,' the blurring of the ontological difference, reminds us of what is done in mathematical reflection: creating ideal entities as actualizations of a potency. This potency - structurability - is of another order than its ideal actualizations - the mathematical objects -, and it is indeed 'forgotten' and inexpressible in mathematical thought. In the mathematical degree of reflection mathematical being as such is indeed in a certain sense *absent*, but absence cannot be written on the account of ancient and scholastic metaphysics: as a metaphysical trend it is thoroughly modern. Heidegger understood rightly that the confusion of the mathematical and the philosophical degree of reflection, which he did not interpret as a confusion but as a fate - *Seinsgeschick* -, must necessarily lead to nihilism.

mathematician really *possesses* in relation to the sphere of ideal structures. Those philosophies all present themselves as absolutely valid insights on the one hand, but reject any claim to knowledge of the absolute on the other. To such philosophical currents, metaphysics counts as an ideological claim to authority which hampers human freedom. The 'necessary construction' is deconstructed and shown to be only one of infinitely many possible ones. Dogmatic systematics has passed into dogmatic nihilism and the project of metaphysics as such has become suspect.<sup>14</sup> Only a clear recognition of the role of the mathematical paradigm in the process leading to this conclusion can still save it.

---

14) Th. Adorno expressed the suspicion that this anti-metaphysical trend has been a process of flight from something which could not be left behind. "The process by which metaphysics continuously ended up where it was conceived to lead away from, has reached its vanishing point" [Th.W. Adorno, *Negative Dialektik*, Suhrkamp, Frankfurt a/M, 1966 p.356.]



# The Wouthuysen Equation

Michiel Hazewinkel

*CWI, Department Algebra & Geometry,  
Kruislaan 413, 1098 SJ Amsterdam,  
The Netherlands*

**Dedication.** I dedicate this paper to Prof. P.C. Baayen, at the occasion of his retirement on 20 December 1994. The beautiful equation which forms the subject matter of this paper was invented by Wouthuysen after he retired.

## **Abstract.**

The four complex variable Wouthuysen equation arises from an original space-time lattice approach to spinor waves and elementary particles. Here the complete space of solutions is described. It consists of one isolated point and one branched  $S_4$ -covering-space over the circle with 8 branching points of order 6, 24 branching points of order 4 and 12 "turning points". The 24 branching points of order 4 are also turning points for two of the four branches.

## 1. THE EQUATIONS

The equations are for four complex variables of unit norm

$$\begin{aligned} 2 - (z_1^2 + z_2^2 + z_3^2 + z_4^2) - (z_1 + z_2 + z_3 + z_4) \\ + (z_1 z_2 + z_1 z_3 + z_1 z_4 + z_2 z_3 + z_2 z_4 + z_3 z_4) = 0 \end{aligned} \quad (1.1)$$

$$\|z_1\| = \|z_2\| = \|z_3\| = \|z_4\| = 1 \quad (1.2)$$

with in addition a stationary phase condition

$$z_1 z_2 z_3 z_4 = 1 \quad (1.3)$$

In terms of real parameters. There are 8 parameters and (1.1), (1.2) together give 6 conditions (2 from (1.1) and 1 each from  $\|z_i\| = 1$ ,  $i = 1, \dots, 4$ ). Given (1.2), (1.3) only gives one extra condition. So by equation counting one could expect 1-dimensional families of solutions. This does indeed turn out to be the case.

Note that the equations (1.1) - (1.3) are symmetric in  $z_1, z_2, z_3, z_4$ . So there is a natural action of the symmetric group on 4 letters,  $S_4$ , and the solutions fall into  $S_4$ -orbits.

## 2. A NUMBER OF SPECIAL SOLUTIONS

*2.1 Solutions with at least one  $z_i$  equal to 1.* These are

$$(1, 1, 1, 1), \text{ a single solution invariant under } S_4 \quad (2.2)$$

$$(1, \zeta_3, \zeta_3, \zeta_3), (\zeta_3, 1, \zeta_3, \zeta_3), (\zeta_3, \zeta_3, 1, \zeta_3), (\zeta_3, \zeta_3, \zeta_3, 1) \quad (2.3)$$

$$(1, \zeta_3^2, \zeta_3^2, \zeta_3^2), (\zeta_3^2, 1, \zeta_3^2, \zeta_3^2), (\zeta_3^2, \zeta_3^2, 1, \zeta_3^2), (\zeta_3^2, \zeta_3^2, \zeta_3^2, 1) \quad (2.4)$$

Here  $\zeta_3 = -\frac{1}{2} + \frac{1}{2}i\sqrt{3}$  is a primitive 3-rd root of unity. These form two  $S_4$ -orbits of size 4 each.

$$(1, 1, \zeta_3, \zeta_3^2), (1, 1, \zeta_3^2, \zeta_3), (1, \zeta_3, 1, \zeta_3^2), (1, \zeta_3^2, 1, \zeta_3) \quad (2.5)$$

$$(1, \zeta_3, \zeta_3^2, 1), (1, \zeta_3^2, \zeta_3, 1), (\zeta_3, 1, 1, \zeta_3^2), (\zeta_3^2, 1, 1, \zeta_3)$$

$$(\zeta_3, 1, \zeta_3^2, 1), (\zeta_3^2, 1, \zeta_3, 1), (\zeta_3, \zeta_3^2, 1, 1), (\zeta_3^2, \zeta_3, 1, 1)$$

This set of solutions forms a single  $S_4$  orbit of size 12. As it turns out (2.2) - (2.5) are the only solutions with at least one  $z_i = 1$ ; see section 3 below for details.

*2.6 Solutions with additional symmetry (besides those in 2.1)*

$$z_i = \pm \frac{1}{3}\sqrt{3} \pm \frac{1}{3}j\sqrt{6}, \quad j = \sqrt{-1} \quad (2.7)$$

This solution satisfies (up to permutations),  $z_2 = -z_1$ ,  $z_4 = -z_3$  and is in fact the only solution with the property. It also satisfies (up to permutations),  $z_2 = \bar{z}_1, z_4 = \bar{z}_3$ .

$$\begin{aligned} z_{1,2} &= \left(-\frac{1}{8} + \frac{3}{8}\sqrt{5}\right) \pm j\left(\frac{1}{8}\sqrt{15} + \frac{1}{8}\sqrt{3}\right), \\ z_{3,4} &= \left(-\frac{1}{8} - \frac{3}{8}\sqrt{5}\right) \pm j\left(\frac{1}{8}\sqrt{15} - \frac{1}{8}\sqrt{3}\right) \end{aligned} \quad (2.8)$$

(up to permutation). This solution also has  $\bar{z}_2 = z_1$ ,  $\bar{z}_4 = z_3$  and there is in fact, besides (2.2), (up to permutation) one one-dimensional family of such solutions on which both (2.7) and (2.8) are located.

*2.9 Solutions with  $z_1 + z_2 + z_3 + z_4 = 0$*

Under this additional condition  $(z_1 + \dots + z_4)^2 = 0$ , so  $z_1^2 + \dots + z_4^2 = -2(z_1z_2 + \dots + z_3z_4)$ , so

$$z_1 z_2 + \dots + z_3 z_4 = -\frac{2}{3} \quad (2.10)$$

Also, using  $z_1 z_2 z_3 z_4 = 1$ ,  $z_1 z_2 z_3 + z_1 z_3 z_4 + z_1 z_2 z_4 + z_2 z_3 z_4 = z_4^{-1} + z_2^{-1} + z_3^{-1} + z_1^{-1} = \bar{z}_4 + \bar{z}_2 + \bar{z}_3 + \bar{z}_1 = 0$  because  $||z_i|| = z_i \bar{z}_i = 1$ . Hence  $z_1 z_2 z_3 + z_1 z_2 z_4 + z_1 z_3 z_4 + z_2 z_3 z_4 = 0$ . Thus the  $z_1, \dots, z_4$  are solutions of the equation

$$z^4 - \frac{2}{3}z^2 + 1 = 0 \quad (2.11)$$

The solutions of this are

$$\frac{1}{3} \pm \frac{2}{3}\sqrt{2} \quad (2.12)$$

and so the  $z_1, z_2, z_3, z_4$  are equal to

$$\pm \sqrt{\frac{1}{3} \pm \frac{2}{3}\sqrt{2}} = \pm \frac{1}{3}\sqrt{3} \pm \frac{1}{3}i\sqrt{6} \quad (2.13)$$

which is again the special solution (2.7).

#### 2.14 Solutions with at least one $z_i$ equal to $-1$ .

There are (up to permutations) three solutions with at least one  $z_i = -1$ . These are

$$z_1 = z_2 = -1, z_{3,4} = (7 + \sqrt{33})^{-1}(3 + \sqrt{33} \pm 2j\sqrt{10 + 2\sqrt{33}}) \quad (2.15)$$

making up one  $S_4$ -orbit of size 12, and

$$-1, \zeta_6^5 = \frac{1}{2} - \frac{1}{2}j\sqrt{3}, \frac{1}{2}\sqrt{3} + \frac{1}{2}j, -\frac{1}{2}\sqrt{3} - \frac{1}{2}j \quad (2.16)$$

$$-1, \zeta_6 = \frac{1}{2} + \frac{1}{2}j\sqrt{3}, \frac{1}{2}\sqrt{3} - \frac{1}{2}j, -\frac{1}{2}\sqrt{3} + \frac{1}{2}j \quad (2.17)$$

and all permutations (making up two complex conjugate  $S_4$  orbits of size 24 each).

### 3. SOLUTIONS WITH AT LEAST ONE $z_i$ EQUAL TO 1.

Permuting the  $z_i$  if necessary, assume  $z_1 = 1$ . Then (1.1) reduces to

$$z_2^2 + z_3^2 + z_4^2 = z_2 z_3 + z_2 z_4 + z_3 z_4 \quad (3.1)$$

This scales. So take  $z = z_2$  and consider

$$1 + z_3^2 + z_4^2 = z_3 + z_4 + z_3 z_4 \quad (3.2)$$

Let  $w_3 = z_3 - 1$ ,  $w_4 = z_4 - 1$ . Then (3.2) turns into

$$w_3^2 + w_4^2 = w_3 w_4 \quad (3.3)$$

This scales again. So consider

$$1 + w_4^2 = w_4 \quad (3.4)$$

which has the solutions

$$w_4 = \frac{1}{2} \pm \frac{1}{2}j\sqrt{3} = \zeta_6, \zeta_6^5 \quad (3.5)$$

where  $\zeta_6 = \frac{1}{2} + \frac{1}{2}j\sqrt{3}$  is a 6-th root of unity. Thus the solutions of (3.2) are of the form

$$z_3 = 1 + w, \quad z_4 = 1 + w\zeta_6, \quad w \in \mathbb{C} \quad (3.6)$$

(including  $w = 0$ ). And those of (3.3) are

$$w_3 = w, \quad w_4 = w\zeta_6 \quad (3.7)$$

From (3.7) it follows that  $w_4$  and  $w_3$  make an angle of  $60^\circ$  with one another, and that they are of equal length. For  $z_3 = 1 + w_3, z_4 = 1 + w_4$  to be on the unit circle,  $w_3$  and  $w_4$  must be on the circle of radius 1 with centre at  $-1$ . Hence they must be conjugate and it readily (see Figure 1) follows that the

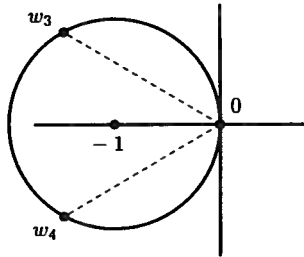


FIGURE 1.

only possibilities are

$$w_3, w_4 = -\frac{3}{2} \pm j\sqrt{3} \quad \text{or} \quad w_3, w_4 = 0$$

(e.g. because the triangle formed by  $0, w_3, w_4$  must have all sides equal) and hence there are only the three possibilities

$$\zeta_3 = \zeta_3, \quad z_4 = \zeta_3^2; \quad z_3 = \zeta_3^2, \quad z_4 = \zeta_3; \quad z_3 = z_4 = 1$$

Thus the possible solutions of (3.1) are



$$(z_2, z_3, z_4) = (z, z, z), (z, \zeta_3 z, \zeta_3^2 z), (z, \zeta_3^2 z, \zeta_3 z)$$

and the solutions of (1.1) - (1.2) with at least one  $z_i$  equal to 1 are (up to permutations):

$$(1, z, z, z), z \in \mathbb{C}; (1, z, \zeta_3 z, \zeta_3^2 z), z \in \mathbb{C}; (1, z, \zeta_3^2 z, \zeta_3 z), z \in \mathbb{C}$$

The requirement (1.3),  $z_1 z_2 z_3 z_4 = 1$ , translates in all these cases to  $z = 1, \zeta_3, \zeta_3^2$  and putting this in gives the 4 solution orbits (2.2) - (2.5) listed above.

#### 4. SOLUTIONS WITH NO $z_i$ EQUAL TO 1.

To study the solutions of (1.1) - (1.3) for which no  $z_i$  is equal to 1, first use the transformation

$$w_i = z_i - 1, \quad i = 1, 2, 3, 4 \quad (4.1)$$

(which has already proved to be useful above). This turns equation (1.1) into

$$w_1^2 + w_2^2 + w_3^2 + w_4^2 = w_1 w_2 + w_1 w_3 + w_1 w_4 + w_2 w_3 + w_2 w_4 + w_3 w_4 \quad (4.2)$$

The second tool is the Cayley transform  $\phi : \mathbb{R} \rightarrow S^1 = \{z \in \mathbb{C} : \|z\| = 1\}$  given by (see Figure 2)

$$\phi(r) = \frac{r - j}{r + j}, \quad j = \sqrt{-1} \quad (4.3)$$

This mapping is 1-1 and onto  $S^1 \setminus \{1\}$ .

Let

$$z_i = \frac{r_i - j}{r_i + j}, \quad i = 1, \dots, 4 \quad (4.4)$$

Then

$$w_i = \frac{-2j}{r_i + j}, \quad i = 1, \dots, 4 \quad (4.5)$$

Set

$$w_i = r_i + j, \quad i = 1, \dots, 4 \quad (4.6)$$

Then the equation (4.2) becomes

$$v_1^{-2} + v_2^{-2} + v_3^{-2} + v_4^{-2} = v_1^{-1} v_2^{-1} + v_1^{-1} v_3^{-1} + \dots + v_3^{-1} v_4^{-1} \quad (4.7)$$

Multiply this with  $v_1^2 v_2^2 v_3^2 v_4^2$ , to obtain

$$v_2^2 v_3^2 v_4^2 + v_1^2 v_3^2 v_4^2 + v_1^2 v_2^2 v_4^2 + v_1^2 v_2^2 v_3^2 = v_1 v_2 v_3 v_4 (v_3 v_4 + \dots + v_1 v_2) \quad (4.8)$$

Let  $e_1, e_2, e_3, e_4$  be the elementary symmetric functions in the  $v_1, \dots, v_4$ ; i.e.

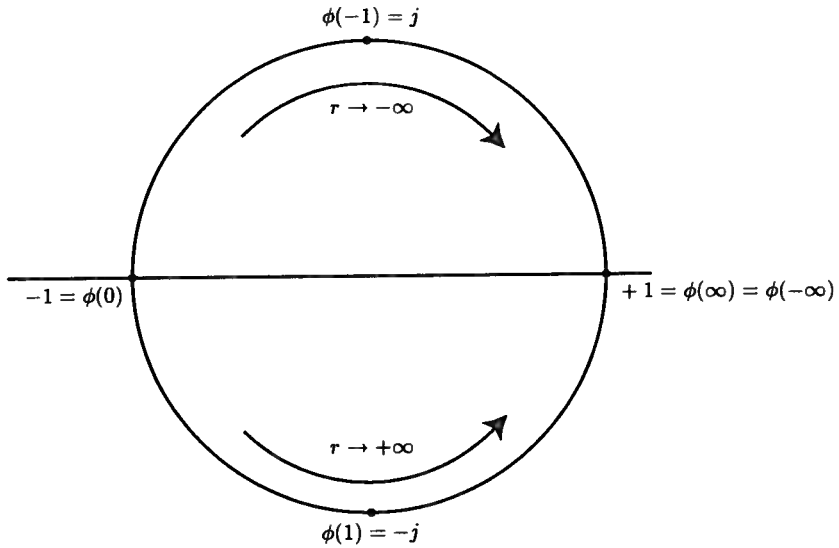


FIGURE 2.

$$\begin{aligned}
 e_1 &= v_1 + v_2 + v_3 + v_4, & e_2 &= v_1v_2 + \dots + v_3v_4, \\
 e_3 &= v_1v_2v_3 + \dots + v_2v_3v_4, & e_4 &= v_1v_2v_3v_4
 \end{aligned}
 \tag{4.9}$$

Then (4.8) becomes

$$e_3^2 = 3e_2e_4 \tag{4.10}$$

Now let  $f_1, f_2, f_3, f_4$  be the elementary symmetric functions in the  $r_1, r_2, r_3, r_4$ , i.e.  $f_1 = r_1 + r_2 + r_3 + r_4$ , etc. Then

$$\begin{aligned}
 e_1 &= f_1 + 4j, & e_2 &= f_2 + 3jf_1 - 6 \\
 e_3 &= f_3 + 2jf_2 - 3f_1 - 4j, & e_4 &= f_4 + jf_3 - f_2 - jf_1 + 1.
 \end{aligned}
 \tag{4.11}$$

Putting this into (4.10) gives the following equations for the  $f_1, f_2, f_3, f_4$

$$f_3^2 + 3f_1f_3 - f_2^2 - 5f_2 + 2 - 3f_2f_4 + 18f_4 = 0 \quad (4.12)$$

$$f_2f_3 + 10f_3 - 3f_1 - 9f_1f_4 = 0$$

Now

$$z_1z_2z_3z_4 = \frac{(r_1 - j)(r_2 - j)(r_3 - j)(r_4 - j)}{(r_1 + j)(r_2 + j)(r_3 + j)(r_4 + j)} \quad (4.13)$$

Let

$$w = (r_1 - j)(r_2 - j)(r_3 - j)(r_4 - j) = f_4 - jf_3 - f_2 + jf_1 + 1 \quad (4.14)$$

By (4.13), equation (1.3) means  $\bar{w} = w$  and by (4.14) this means

$$f_1 = f_3 \quad (4.15)$$

Putting this in (4.12) we see that there are two possibilities

$$f_1 = f_3 = 0 \quad (4.16A)$$

$$f_2 = 9f_4 - 7 \quad (4.16B)$$

In case A, the first equation of (4.12) becomes

$$f_2^2 + (5 + 3f_4)f_2 - (18f_4 + 2) = 0 \quad (4.17A)$$

and in case B, the first equation of (4.12) becomes

$$f_1^2 = 27f_4^2 - 30f_4 + 3 = 3(9f_4 - 1)(f_4 - 1) \quad (4.17B)$$

So, to find all solutions of (1.1) - (1.3) for which no  $z_i$  is equal to 1 it is necessary and sufficient to consider the equation

$$r^4 + f_1r^3 + f_2r^2 + f_3r + f_4 = 0 \quad (4.18)$$

under the conditions

$$\text{Family A : } f_1 = f_3 = 0 \text{ and } f_2^2 + (5 + 3f_4)f_2 - (18f_4 + 2) = 0$$

$$\text{Family B : } f_1 = f_3, f_2 = 9f_4 - 7, f_1^2 = 27f_4^2 - 30f_4 + 3 = 3(9f_4 - 1)(f_4 - 1)$$

and to find out for which cases all four roots of (4.18) are real.

To conclude this section let's find out whether the families A and B can intersect. For an intersection we have  $f_1 = 0 = f_3$  and, hence from (4.17B),  $f_4 = 1/9, f_2 = -6; f_4 = 1, f_2 = 2$  Then and only then are all four of (4.16) - (4.17) satisfied.

If  $f_4 = 1, f_2 = 2, f_1 = f_3 = 0$ , The solutions of (4.18) are

$$j, j, -j, -j \quad (4.19)$$

i.e. two pairs of coinciding non real solutions. This gives no solution to the Wouthuysen equation, but will still be usefull later.

If  $f_4 = 1/9$ ,  $f_2 = -6$ ,  $f_1 = f_3 = 0$ , the solutions of (4.18) are

$$\pm\left(\frac{1}{3}\sqrt{15} + \frac{2}{3}\sqrt{3}\right), \pm\left(\frac{1}{3}\sqrt{15} - \frac{2}{3}\sqrt{3}\right) \quad (4.20)$$

which are all four real and which give the special solution (2.8):

$$\begin{aligned} z_{1,2} &= \left(-\frac{1}{8} + \frac{3}{8}\sqrt{15}\right) \pm j\left(\frac{1}{8}\sqrt{15} + \frac{1}{8}\sqrt{3}\right), \\ z_{3,4} &= \left(-\frac{1}{8} - \frac{3}{8}\sqrt{15}\right) \pm j\left(\frac{1}{8} - \frac{3}{8}\sqrt{15}\right) \end{aligned} \quad (4.21)$$

## 5. THE FAMILY A

In this case the equation becomes

$$r^4 + f_2 r^2 + f_4 = 0.$$

with

$$f_2^2 + (5 + 3f_4)f_2 - (18f_4 + 2) = 0 \quad (5.2)$$

We shall use  $f_4$  as the main parameter. This will turn out to be the right choice, even though (5.2) suggests that  $f_2$  might be easier to work with.

For (5.1) to have four real roots, it is necessary and sufficient that  $f_4 \geq 0$ ,  $f_2 \leq 0$ ,  $f_2^2 \geq 4f_4$  (besides  $f_2$  real). The conditions  $f_4 \geq 0$  and  $f_2 \leq 0$  imply that only the solution

$$f_2 = -\frac{5}{2} - \frac{3}{2}f_4 - \frac{1}{2}\sqrt{9f_4^2 + 132f_4 + 33} \quad (5.3)$$

of (5.2) qualifies. If  $f_2$  is given by (5.3) then

$$f_2^2 \geq \frac{1}{4}(9f_4^2 + 132f_4 + 33) > 4f_4$$

So, the family A consists of precisely one family of solutions parametrized by  $f_4 \geq 0$ . Because  $f_2^2 > f_4$ , the two solutions of

$$y^2 + f_2 y + f_4 = 0 \quad (5.4)$$

are unequal. So the only case in which the four solutions of (5.1) can have two or more equal is when  $f_4 = 0$ . Then

$$r_1 = r_2 = 0, \quad r_{3,4} = \pm \frac{1}{2}\sqrt{10 + 2\sqrt{33}} \quad (5.5)$$

corresponding to the special solution (2.10) of the Wouthuysen equations.

## 6. THE FAMILY B

In this case the equation becomes

$$r^4 + f_1 r^3 + f_2 r^2 + f_1 r + f_4 = 0 \quad (6.1)$$

subject to following conditions on the coefficients

$$f_2 = 9f_4 - 7, \quad f_1^2 = 3(f_4 - 1)(9f_4 - 1) \quad (6.2)$$

and the question is when (6.1) will have all solutions real. This certainly requires  $f_1$  to be real, which by (6.2) implies that  $f_4 \leq 1/9$ , or  $f_4 \geq 1$ . Thus there are four subfamilies to be considered

$$f_4 \geq 1, \quad f_1 = \sqrt{27f_4^2 - 30f_4 + 3} \quad (B1)$$

$$f_4 \geq 1, \quad f_1 = -\sqrt{27f_4^2 - 30f_4 + 3} \quad (B2)$$

$$f_4 \leq \frac{1}{9}, \quad f_1 = \sqrt{27f_4^2 - 30f_4 + 3} \quad (B3)$$

$$f_4 \leq \frac{1}{9}, \quad f_1 = -\sqrt{27f_4^2 - 30f_4 + 3} \quad (B4)$$

Under  $(r_1, r_2, r_3, r_4) \mapsto (-r_1, -r_2, -r_3, -r_4)$ ,  $f_2$  and  $f_4$  remain the same and  $f_1$  and  $f_3$  change sign. Hence (B1) (for a given value of  $f_4$ ) gives four real solutions iff (B2) does so (for the same value of  $f_4$ ). Similarly for (B3) and (B4). Thus it suffices to examine (B3) and (B1).

The discriminant of (6.1) is equal to

$$D = \prod_{i < k} (r_i - r_k)^2 \quad (6.3)$$

where  $r_1, r_2, r_3, r_4$  are the four roots of (6.1). It turns out that under (6.2)

$$D = -2^{10}(2f_4^3 - 7f_4^2 + 8f_4 - 3) = -2^{10}(f_4 - 1)^2(2f_4 - 3). \quad (6.4)$$

This is a substantial calculation but it is less surprising than it maybe looks. First,  $D$  is of course a polynomial in the  $f_1, f_2, f_3, f_4$  and it is homogeneous of degree 12 where  $f_i$  has weight  $i$ ,  $i = 1, \dots, 4$ . Under  $r_i \mapsto -r_i$ ,  $i = 1, \dots, 4$ ,  $D$  remains invariant. As  $f_1, f_3$  change sign under  $r_i \mapsto -r_i$  and  $f_2, f_4$  remain invariant,  $f_1$  and  $f_3$  can only occur in the monomials in  $D$  in the forms  $f_1^2, f_1 f_3, f_3^2$ . However, the substitutions (6.2) are not homogeneous so that the degree could become as high as 12. The monomials in the discriminant of a fourth degree polynomial are of maximal degree 6 in  $f_1, f_3$  combined. Thus

a polynomial of degree 6 in  $f_4$  could occur. A final drop in degree of 3 occurs because there are three coinciding roots at  $f_4 = \infty$ . Finally because there are coinciding roots of (6.1) at  $f_4 = 1$  one of the roots of  $D$  must be 1.

For the subfamily (B3) (and (B4)) we have that at  $f_4 = 1/9$  there are four different real solutions, see (4.20). Because  $D \neq 0$  for  $-\infty < f_4 < 1/9$ , this must remain so for the whole family. Thus (B3) and (B4) represent two one dimensional families of solutions to the Wouthuysen equations parametrized by  $f_4 \leq \frac{1}{9}$ .

For  $f_4 \geq 1$ , i.e. the families (B1) and (B2),  $D = 0$  at  $f_4 = 3/2$ . For this value of  $f_4$  (6.1) becomes (for (B1))

$$r^4 + \frac{5}{2}\sqrt{3}r^3 + 6^{1/2}r^2 + \frac{5}{2}\sqrt{3}r + \frac{3}{2} = 0 \quad (6.5)$$

with the solutions

$$-\sqrt{3}, -\sqrt{3}, -\frac{1}{4}\sqrt{3} + \frac{1}{4}j\sqrt{5}, -\frac{1}{4}\sqrt{3} - \frac{1}{4}j\sqrt{5} \quad (6.6)$$

At  $f_4 = 1$ , equation (6.1) has four non real solutions, viz.  $j, j, -j, -j$ . So for  $1 < f_4 < 3/2$ , it remains the case that (6.1) has four non real solutions (because for this to change  $D$  must assume the value zero). As  $D \neq 0$  for  $3/2 < f_4 < \infty$ , the family (B1) and (B2) have for these values of  $f_4$  either four non real solutions or two real and two non real (complex conjugate) solutions. As it turns out the latter is the case. A numerical check shows e.g. that at  $f_4 = 10$  the four solutions are approximately

$$-47.287, -0.606, -0.564 \pm 0.177j$$

In both cases (B1) and (B2) do not contribute to solutions of the Wouthuysen equation.

For later use we also need the solutions of the (B3) and (B4) families at  $f_4 = 0$ . The equation for the (B3) case then becomes

$$r^4 + \sqrt{3}r^3 - 7r^2 + \sqrt{3}r = 0 \quad (6.7)$$

with solutions

$$0, \sqrt{3}, 2 - \sqrt{3}, -2 - \sqrt{3} \quad (6.8)$$

## 7. MATCHING THE SOLUTIONS WITH A $z_i$ EQUAL TO 1 TO THE A, B3, B4 FAMILIES

Under  $\phi : \mathbb{R} \rightarrow S^1$ ,  $+\infty$  goes to 1, and so does  $-\infty$ . (So the true parameter space is the circle  $\phi(\mathbb{R}) = \phi(\{f_4\})$ ). To see how the solutions with a  $z_i$  equal to 1 fit with the A, B3 and B4 families, it therefore suffices to study what happens to the corresponding solutions as  $f_4 \rightarrow \infty$  (for the A-family) and as  $f_4 \rightarrow -\infty$  (for the B3 and B4 families).

7.1 The A-family for  $f_4 \rightarrow \infty$ .

First consider an A-family of solutions

$$r^4 + f_2 r^2 + f_4 = 0 \quad f_2 = -\frac{5}{2} - \frac{3}{2}f_4 - \frac{1}{2}\sqrt{9f_4^2 + 132f_4 + 33} \quad (7.2)$$

As  $f_4 \rightarrow \infty$ ,  $f_4^{-1}f_2$  goes to  $-3$ . Let  $s = r^{-1}$ . Then the equation for  $s$  is

$$s^4 + f_4^{-1}f_2s^2 + f_4^{-1} = 0 \quad (7.3)$$

which in the limit  $f_4 \rightarrow \infty$ , goes to

$$s^4 - 3s^2 = 0 \quad (7.4)$$

It follows that as  $f_4 \rightarrow \infty$ , two solutions of (7.2) go each to  $-\infty$  or  $+\infty$  and the other two go to  $-\frac{1}{3}\sqrt{3}$ ,  $\frac{1}{3}\sqrt{3}$ . However, the four solutions of (7.2) cannot cross as  $f_4 \rightarrow \infty$  ( $f_4 > 0$ ), therefore the only possibility is that one goes to  $-\infty$  and the other to  $+\infty$ .

So up to permutations the limit solutions are

$$(-\infty, -\frac{1}{3}, \sqrt{3}, \frac{1}{3}\sqrt{3}, +\infty) \quad (7.5)$$

which under  $\phi : \mathbb{R} \rightarrow S^1$  corresponds to the solutions (2.5)

$$(1, \zeta, \zeta^2, 1) \quad (7.6)$$

where  $\zeta = \zeta_3$ .

And indeed a small numerical check shows that for  $f_4 = 10^3$ ,  $10^5$ , respectively, the solutions of (7.2) are, respectively, approximately equal to

$$-54.890, \quad -0.576, \quad 0.576, \quad 54.890$$

$$-547.735, \quad -0.577, \quad 0.577, \quad 547.735$$

while  $\frac{1}{3}\sqrt{3}$  is about 0.577.

7.7 The B3-family for  $f_4 \rightarrow \infty$ .

Now let's consider a B3-family of solutions

$$r^4 + f_1 r^3 + f_2 r^2 + f_1 r + f_4 = 0, \quad (7.8)$$

$$f_2 = 9f_4 - 7, \quad f_1 = \sqrt{27f_4^2 - 30f_4 + 3}$$

as  $f_4 \rightarrow \infty$  ( $f_4 \leq 1/9$ ). As  $f_4 \rightarrow \infty$ , because  $f_1 \propto 3\sqrt{3}|f_4|$ ,

$$f_4^{-2}f_2 \rightarrow 9, f_4^{-1}f_1 \rightarrow -3\sqrt{3} \quad (7.9)$$

Let  $s = r^{-1}$ . Then the equation for  $s$  is

$$s^4 + f_4^{-1}f_1s^3 + f_4^{-1}f_2s^2 + f_4^{-1}f_1s + f_4^{-1} = 0 \quad (7.10)$$

which in the limit,  $f_4 \rightarrow \infty$ , goes to

$$s^4 - 3\sqrt{3}s^3 + 9s^2 - 3\sqrt{3}s = s(s - \sqrt{3})^3 = 0 \quad (7.11)$$

with solutions

$$0, \sqrt{3}, \sqrt{3}, \sqrt{3} \quad (7.12)$$

It follows that as  $f_4 \rightarrow \infty$  one of the solutions of (7.8) goes to  $\infty$  or  $-\infty$  and the others to  $\frac{1}{3}\sqrt{3}$ .

Now at  $f_4 = 0$  the solutions of (7.8) are

$$-2 - \sqrt{3}, 0, 2 - \sqrt{3}, \sqrt{3} \quad (7.13)$$

The roots cannot cross as  $f_4 \rightarrow -\infty$ , and the smallest one,  $-(2 + \sqrt{3})$ , cannot cross 0 again (because there are no zero solutions of (7.8) for  $f_4 < 0$ ). It follows that (7.13) must go to

$$(-\infty, \frac{1}{3}\sqrt{3}, \frac{1}{3}\sqrt{3}, \frac{1}{3}\sqrt{3}) \quad (7.14)$$

which corresponds to the solution

$$(1, \zeta^2, \zeta^2, \zeta^2) \quad (7.15)$$

of the Wouthuysen equation.

A numerical check gives that for  $f_4 = -10^4$ , the four solutions are approximately equal to

$$-51966.143, \quad 0.572, \quad 0.577, \quad 0.583$$

while  $\frac{1}{3}\sqrt{3}$  is about 0.577.

## 8. THE TOPOLOGICAL STRUCTURE OF THE SPACE OF SOLUTIONS

Apart from the identifications at  $r = \infty, -\infty$ , i.e. at  $z = 1$ , the picture of the solution space is made up of 12 pieces as depicted in Figure 3.

Here all special (intersection) points have been made fat dots and given their  $r$ -coordinates. For the points with an  $r = \pm\infty$  the corresponding  $z$ -coordinates have also been given. A crossing point of families that has not been made fat is not an existing crossing point but an artifact of the drawing. In particular there is no intersection of a B4-family with a B3-family for  $-\infty \leq f_4 < 1/9$ .

There are twelve such pieces. The other eleven are obtained by applying appropriate elements of  $S_4$  to the picture shown above. To get the complete



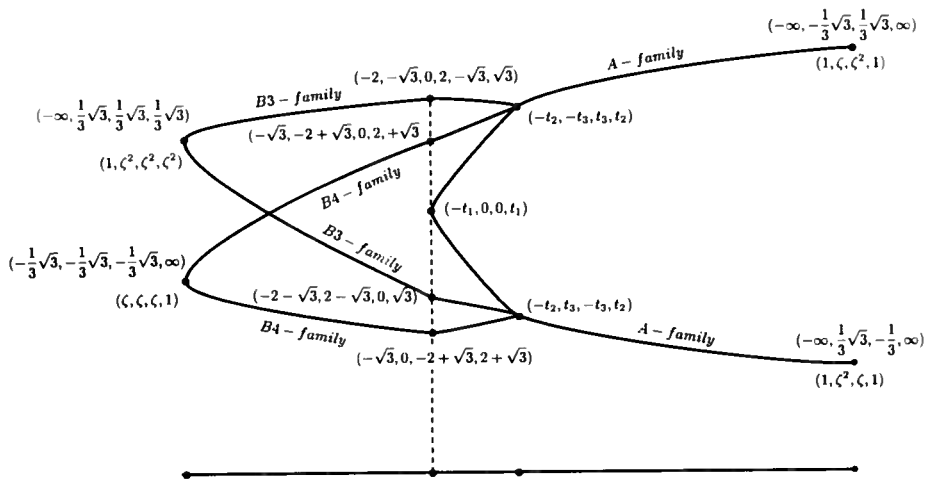


FIGURE 3.

global picture it suffices to identify the points above  $f_4 = \pm\infty$  according to the coordinates attached to them. The complete topological picture is given in Figure 4.

As above let  $\zeta = \zeta_3$

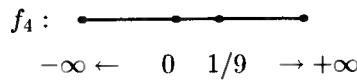
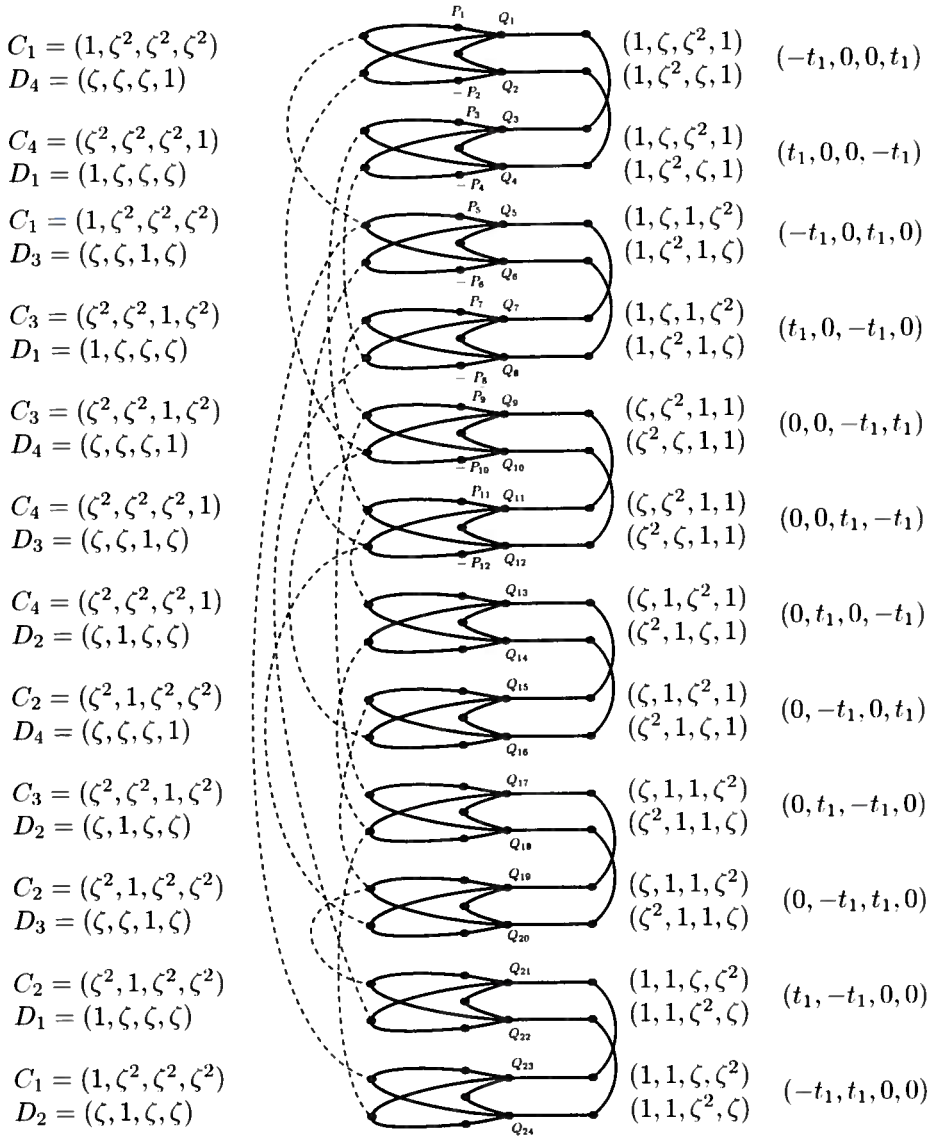


FIGURE 4.

The solution (1, 1, 1) is completely isolated, and all others are connected by the scheme drawn above, where the dotted lines indicate identifications. In the above (see also Figure 5)

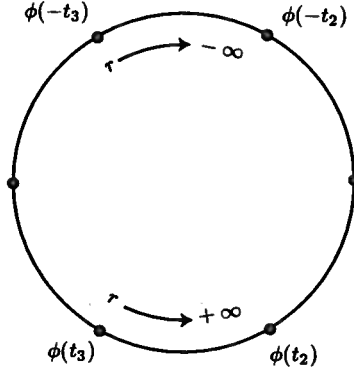


FIGURE 5.

$$Q_1 = \phi(-t_2, -t_3, t_3, t_2) = \left( -\frac{1}{8} + \frac{3}{8}\sqrt{5} + j\left(\frac{1}{8}\sqrt{3}\right), -\frac{1}{8} - \frac{3}{8}\sqrt{5} + j\left(\frac{1}{8}\sqrt{15} - \frac{1}{8}\sqrt{3}\right), \right. \\ \left. -\frac{1}{8} - \frac{3}{8}\sqrt{5} - j\left(\frac{1}{8}\sqrt{15} - \frac{1}{8}\sqrt{3}\right), -\frac{1}{8} + \frac{3}{8}\sqrt{5} - j\left(\frac{1}{8}\sqrt{3}\right) \right)$$

where  $t_2 = \frac{1}{3}\sqrt{15} + \frac{2}{3}\sqrt{3}$ ,  $t_3 = \frac{1}{3}\sqrt{15} - \frac{2}{3}\sqrt{3}$

And further, using the notation  $\sigma(s_1, s_2, s_3, s_4) = (s_{\sigma(1)}, s_{\sigma(2)}, s_{\sigma(3)}, s_{\sigma(4)})$ ,

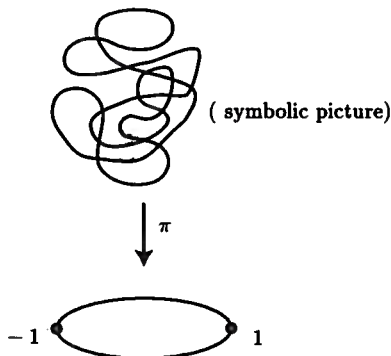
$$\begin{aligned} Q_2 &= (23)Q_1 = \phi(-t_2, t_3, -t_3, t_2) \\ Q_3 &= (14)Q_1 = \phi(t_2, -t_3, +t_3, -t_2) \\ Q_4 &= (23)(14)Q_1 = \phi(t_2, +t_3, -t_3, -t_2) \\ Q_5 &= (34)Q_1 = \phi(-t_2, -t_3, t_2, t_3) \\ Q_6 &= (234)Q_1 = \phi(-t_2, t_3, t_2, -t_3) \\ Q_7 &= (143)Q_1 = \phi(t_2, -t_3, -t_2, t_3) \\ Q_8 &= (1423)Q_1 = \phi(t_2, +t_3, -t_2, -t_3) \\ Q_9 &= (123)Q_1 = \phi(-t_3, t_3, -t_2, t_2) \\ Q_{10} &= (13)Q_1 = \phi(t_3, -t_3, -t_2, t_2) \\ Q_{11} &= (1234)Q_1 = \phi(-t_3, t_3, t_2, -t_2) \\ Q_{12} &= (134)Q_1 = \phi(t_3, -t_3, t_2, -t_2) \\ Q_{13} &= (124)Q_1 = \phi(-t_3, t_2, t_3, -t_2) \\ Q_{14} &= (1324)Q_1 = \phi(t_3, t_2, -t_3, -t_2) \end{aligned}$$

$$\begin{aligned}
Q_{15} &= (12)Q_1 = \phi(-t_3, -t_2, t_3, t_2) \\
Q_{16} &= (132)Q_1 = \phi(t_3, -t_2, -t_3, t_2) \\
Q_{17} &= (1243)Q_1 = \phi(-t_3, t_2, -t_2, t_3) \\
Q_{18} &= (13)(24)Q_1 = \phi(t_3, t_2, -t_2, -t_3) \\
Q_{19} &= (12)(34)Q_1 = \phi(-t_3, -t_2, t_2, t_3) \\
Q_{20} &= (1342)Q_1 = \phi(t_3, -t_2, t_2, -t_3) \\
Q_{21} &= (1432)Q_1 = \phi(t_2, -t_2, -t_3, t_3) \\
Q_{22} &= (142)Q_1 = \phi(t_2, -t_2, t_3, -t_3) \\
Q_{23} &= (243)Q_1 = \phi(-t_2, t_2, -t_3, t_3) \\
Q_{24} &= (24)Q_2 = \phi(-t_2, t_2, t_3, -t_3)
\end{aligned}$$

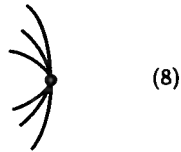
In words, the space of solutions of the Wouthuysen equation consists of one isolated point,  $(1, 1, 1, 1)$ , and a branched curve. This branched curve, and the isolated point, come with a natural projection to the circle. The group  $S_4$  acts on the space of solutions, leaving the isolated point invariant. The projection to the circle is invariant under this action. Let  $S$  denote the solution space and  $\pi : S \rightarrow S^1 = \mathbb{R} \cup \{\infty\}$  this invariant projection.

In terms of the  $r$ -coordinates,  $\mathbb{R} \cup \{\infty\}$ ,  $-\infty = +\infty$ , the picture is as follows

- (i) Above all  $1/9 < y < \infty$ , there are 24 points which form one  $S_4$ -orbit. The inverse under  $\pi$  of a small enough neighborhood of such a point consists of 24 disjoint intervals.



- (ii) Above  $y = \infty$  (corresponding to 1 under the Cayley transform) there are 21 points: the isolated solution point  $(1, 1, 1, 1)$ , which is an invariant point of the  $S^4$ -action, an  $S^4$ -orbit of size 12, and two complex conjugate  $S_4$ -orbits of size 4. These are branching points of order 6. Locally around one of the points of the orbit of size 12, the branched solution curve looks like an interval turning back. Locally around a point of the two  $S_4$ -orbits of size 4 the picture is a six branched star as depicted below. Thus the inverse image of a small interval around the point  $\infty$  of the circle  $\mathbb{R} \cup \{\infty\}$  looks like the disjoint union of 12 intervals, 8 six branched stars and one isolated point.



(8)



(12)



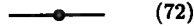
(1)

- (iii) Above  $y = \frac{1}{9}$  there are 24 points which form a single  $S_4$ -orbit. They are all branching points of order 4. The inverse of a small interval around  $y = 1/9$  looks like the disjoint union of 24 4-branched stars like depicted below.



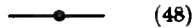
(24)

- (iv) Above all  $0 < y < 1/9$  there are 72 points, which from three  $S_4$ -orbits; two of these are complex conjugate, the third is invariant under complex conjugation. The inverse image of a small enough interval around these  $y$  looks like 72 disjoint copies of that interval



(72)

- (v) Above  $y = 0$  there are 60 points. They form one orbit of size 12 and two complex conjugate orbits of size 24. The points from the orbit of size 12 are "turning points" in a sense which should be clear from the picture below. The inverse image of a small interval around such a point consists of 60 disjoint intervals of which 12 are "turn back" intervals.



(48)



(12)

- (vi) Above  $-\infty < y < 0$  there are 48 points which form two complex conjugate orbits of size 24 each. The inverse image of a small interval around these  $y$  looks like the disjoint union of 48 copies of that interval

$$\text{---} \bullet \text{---} \quad (48)$$

So, in fact, the group  $\mathbb{Z}/(2) \times S_4$  acts on the space of solutions and  $\pi : S \rightarrow S^1$  is invariant under this action (of a group of order 48).

#### REFERENCES

1. S.A. WOUTHUYSEN, *A remarkable space time lattice and its spinor waves*, preprint, Univ. of Amsterdam, Astronomical Inst., AIAP - 1992 - 012.
2. S.A. WOUTHUYSEN, *A remarkable space time lattice and its spinor waves II*, preprint, Univ. of Amsterdam, 1994.

# Prehistory of the ASF+SDF System (1980–1984)

*Dedicated to Cor Baayen*

Jan Heering and Paul Klint

## 1 MONOLINGUAL BEGINNING

Our work on programming environments started in 1980 with the design of a dedicated environment for the Summer programming language [1], an object-oriented language with class definitions. Rather than a dedicated Summer environment, the general concept of a *monolingual environment* emerged [2]. In such an environment, a single language is used in different modes. More specifically, we investigated the requirements an integrated command/programming/-debugging language would have to satisfy. Since Summer had not been designed with this particular purpose in mind, it is not surprising that a monolingual environment for Summer would have involved a revision of the language. This may have been one of the reasons we never “instantiated” the monolingual concept for Summer, but there were other, more important, ones:

- At that time Leo Geurts, Lambert Meertens, and other members of the Afdeling Informatica were developing the B language system (later renamed to ABC), which had a monolingual character in the sense that the command and programming modes of the system were integrated. The development of a monolingual environment for a suitably revised version of Summer would have been a major effort without obvious additional benefits.
- We started to realize that a monolingual environment would be a closed world whose facilities could not be easily borrowed or reused by other languages. Since every application has its own language (however small), we decided it would be much more efficient to develop a generic multilingual environment. Its design was started in 1982.

## 2 A PROGRAMMING ENVIRONMENT BASED ON LANGUAGE DEFINITIONS

The idea was to base the generic environment on *language definitions*. These would consist of a combined syntax/prettyprinting section and two additional sections for static and dynamic semantics. The generic environment would support the interactive development of language definitions and their compilation to language specific subenvironments. It would view language definitions as libraries of language constructs from which individual constructs could be borrowed or reused to facilitate the construction of new definitions. A language

needing an **if**-statement, for instance, would probably be able to borrow a suitable one from another language for which a definition already existed in the system.

We had some experience with language definitions. Part of the semantics of Summer had been described in a formalism consisting of BNF-like rules with embedded variables to which semantic actions written in Summer itself were attached [3]. Furthermore, Gert Florijn and Geert Rolf had written PGEN, an LL(1) parser generator [4]. One of the things PGEN taught us was that molding grammars to fit the LL(1) restriction was no fun. This influenced our early decision to allow general context-free syntax in language definitions. Aloysius Tan designed a VLSI-algorithm to reduce the parsing time in the general context-free case to an acceptable value [5]. This was long before the syntax definition formalism SDF and lazy/incremental parser generation.

In the meantime, Henk Kroeze had experimented with a combined syntax/prettyping language for use in the first section of language definitions [6]. It turned out, however, that BNF rules with integrated prettypint instructions were unreadable, and this remained a problem.

Although the generic environment we had in mind obviously needed a built-in semantics definition formalism (we did not yet know which one), it would be possible to use any language for which a definition had been constructed as a semantics definition formalism in the system. The corresponding towers of language interpreters would be very inefficient, so they would have to be flattened by the removal of intermediate layers. This we planned to do by *partial evaluation*.

This system concept was discussed with Wim Böhm, Marleen Sint, and Arthur Veen at several Data Flow Club meetings in 1982. It was subsequently presented at the Colloquium Programmeeromgevingen in the fall of that year [7, 8] and at the NGI-SION Symposium in Amsterdam in March 1983.

### 3 ALGEBRAIC SPECIFICATION

The main decision facing us was what semantics definition method to use. The importance of partial evaluation in the system suggested a functional method without side-effects. Although denotational semantics would have been a natural choice, the closest we came to it was when we considered a statically scoped version of Lisp as a semantics definition formalism.

Among the papers on partial evaluation we studied were several by Valentin Turchin, which used the (string) rewrite rule language Refal, and we started discussing rewrite rules with Jan Bergstra. He taught us the relation between (term) rewrite rules and algebraic specifications. The fact that modularization was an important topic in the algebraic specification community was attractive to us in view of the modular construction of language definitions the generic environment had to support.

Although the algebraic semantics of programming languages was not a well developed subject, Jan Bergstra and Jan Willem Klop were working on *process*



*algebra* (the algebraic semantics of processes) and we somehow suspected that algebraic specifications would be suitable for describing the static and dynamic semantics of languages in the generic environment. We never considered using different formalisms for static and dynamic semantics since we did not see a clear distinction between them. In this we were perhaps influenced by the monolingual concept discussed in Section 1. At a later stage, we started by not making a distinction between lexical and context-free syntax description in the syntax definition formalism SDF, but this proved untenable.

After a joint excursion into object-oriented algebraic specification [9], we set out to give an algebraic definition of the toy language PICO. Since we did not yet have a well-developed algebraic specification formalism, it was designed simultaneously. This became ASF. The syntax definition formalism SDF did not yet exist either, so the PICO definition included an algebraically specified syntax of PICO and a parser.

The proper modularization of the PICO definition turned out to be a major problem whose solution involved the repeated redesign of the module construction operators of ASF. The modularization finally adopted was very reasonable, but it did not permit the reuse of individual PICO constructs in other language definitions. In this respect we did not achieve one of our original goals and this is still an open problem.

In the meantime, partial evaluation had not been forgotten. Although its algebraic semantics had not been studied in detail, it had been clear from the outset that algebraic specification and term rewriting were excellent frameworks for partial evaluation. As it turned out, partial evaluation involves the notion of  $\omega$ -completeness of algebraic specifications. Somewhat ironically, the idea to allow any language for which a definition had been constructed as a semantics definition formalism in the system, which had been the main reason for studying partial evaluation, was gradually abandoned with the advent of algebraic specifications. Anyway, we finished both the PICO definition [10] and the partial evaluation paper [11] virtually at the time the GIPE project started in January 1985. At that time the implementation of ASF consisted of a parser, a type checker, and a Structure Diagram generator, all of them written in Summer using the PGEN parser generator mentioned before. Term rewriting had not yet been implemented.

#### 4 TOWARDS THE ESPRIT/GIPE PROJECT

In July 1983 Paul Klint had visited INRIA Rocquencourt where he had familiarized himself with several generic environments [12]. One of them was the Mentor system which had been developed in the seventies by Véronique Donzeau-Gouge, Gérard Huet, Gilles Kahn, Bernard Lang, and others at INRIA [13]. In fact, Mentor was rather similar to what we had in mind for the syntactic part of the generic environment. Furthermore, its extension towards semantics had just begun with the development of the Typol language [14, 15], bringing INRIA's work even closer to ours.

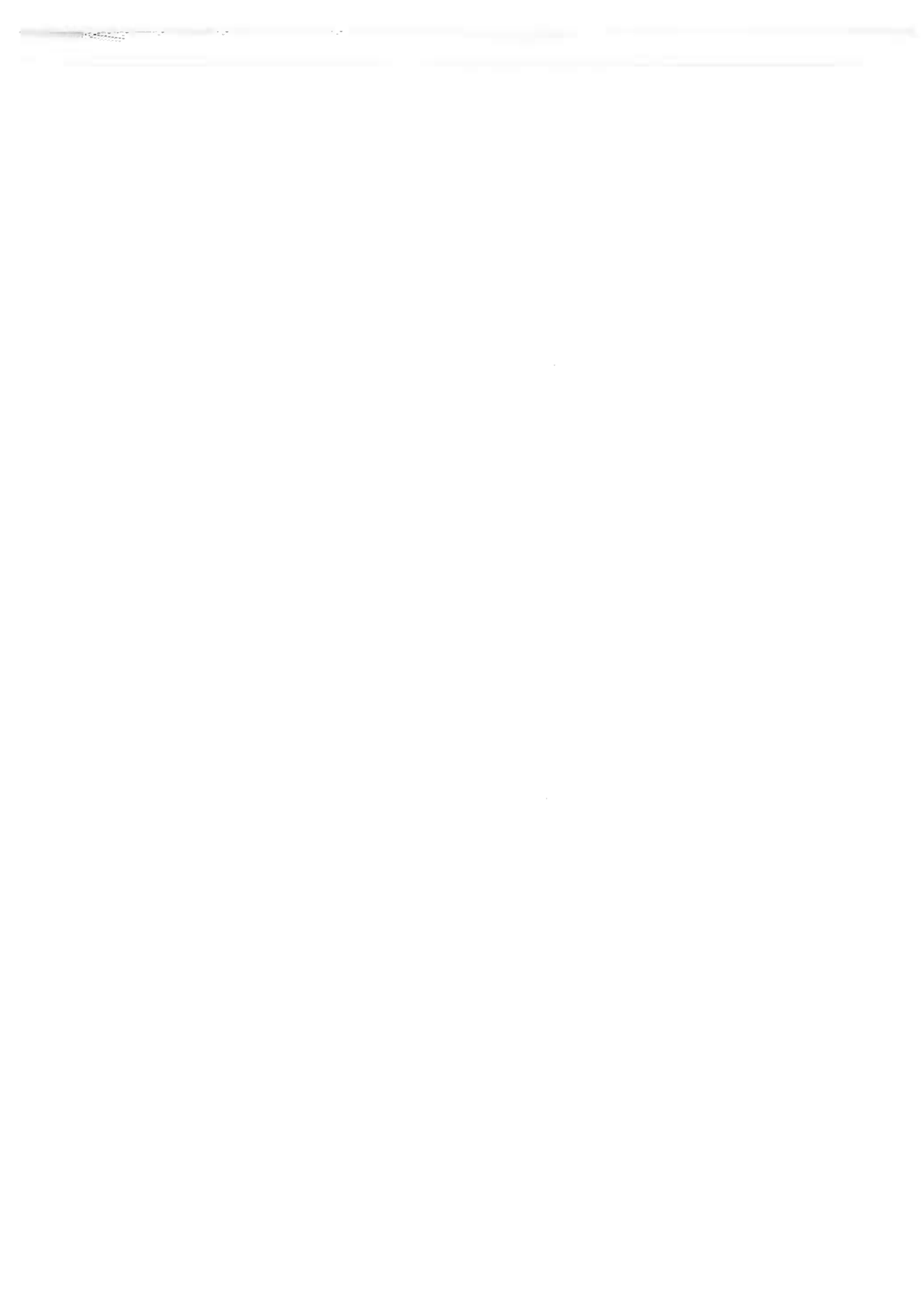
Typol was based on Plotkin's Structural Operational Semantics, but it may be interesting to note that earlier experiments had been done with Formol, an Ada-like specification language specially designed for writing denotational semantics definitions of programming languages. Formol specifications were considered too low-level, however, and denotational semantics was abandoned.

Paul's visit did not immediately lead to further co-operation with INRIA, but in the spring of 1984 Gilles Kahn proposed to submit a joint ESPRIT proposal on the Generation of Interactive Programming Environments. For INRIA, it would be basically an extension of Mentor with semantics facilities. For us, it would be a continuation of our work on a generic environment based on algebraic language definitions. The ensuing proposal (part of which was later published [16]) was accepted by the European Communities and the GIPE project started in January 1985 with the software companies BSO (The Netherlands) and SEMA-METRA (France) as industrial partners. When it ended 5 years later, GIPE II took over for another 4 years [17].

#### REFERENCES

1. P. Klint, *From Spring to Summer*, Ph.D. Thesis, TH Eindhoven, 1982. Published as LNCS, Vol. 205, 1985.
2. J. Heering and P. Klint, Towards monolingual programming environments, Report IW 185/81, Mathematisch Centrum, Amsterdam, December 1981. Published in *ACM Transactions on Programming Languages and Systems*, 7 (1985), pp. 183–213.
3. P. Klint, Formal language definitions can be made practical, Report IW 159/81, Mathematisch Centrum, Amsterdam, 1981. Published in J.W. de Bakker and J.C. van Vliet (Eds.), *Algorithmic Languages*, North-Holland, 1981, pp. 115–132, and in [1, Chapter 4].
4. G. Florijn and G. Rolf, PGEN—A general purpose parser generator, Report IW 157/81, Mathematisch Centrum, Amsterdam, 1981.
5. H.D.A. Tan, VLSI-algoritmen voor herkenning van context-vrije talen in lineaire tijd, Report IN 24/83, Mathematisch Centrum, Amsterdam, June 1983 (VLSI algorithms for the recognition of context-free languages in linear time—in Dutch). See also: A. Nijholt, Overview of parallel parsing strategies, in M. Tomita (Ed.), *Current Issues in Parsing Technology*, Kluwer Academic, 1991, Section 14.4.2.
6. H. Kroeze, Een taalonafhankelijke benadering van prettyprinten, Report IN 21/82, Mathematisch Centrum, Amsterdam, December 1982 (A language independent approach to prettyprinting—in Dutch).
7. J. Heering, Taaldefinities als kern voor een programmeeromgeving, in *Colloquium Programmeeromgevingen*, MC Syllabus 30, Mathematisch Centrum, Amsterdam, 1983, pp. 69–81 (A programming environment based on language definitions—in Dutch).
8. P. Klint, Partiële evaluatie als implementatiemethode voor een programmeeromgeving, in *Colloquium Programmeeromgevingen*, MC Syllabus 30,

- Mathematisch Centrum, Amsterdam, 1983, pp. 83–100 (Partial evaluation as an implementation method for a programming environment—in Dutch).
9. J.A. Bergstra, J. Heering, and J.W. Klop, Object-oriented algebraic specification: proposal for a notation and 12 examples, Report CS-R8411, CWI, Amsterdam, June 1984.
  10. J.A. Bergstra, J. Heering, and P. Klint, Algebraic definition of a simple programming language, Report CS-R8504, CWI, Amsterdam, February 1985. Published in J.A. Bergstra, J. Heering, and P. Klint (Eds.), *Algebraic Specification*, ACM Press Frontier Series, 1989, Chapter 2.
  11. J. Heering, Partial evaluation and  $\omega$ -completeness of algebraic specifications, Report CS-R8501, CWI, Amsterdam, January 1985. Published in *Theoretical Computer Science*, **43** (1986), 149–167.
  12. P. Klint, A survey of three language-independent programming environments, Report IW 240/83, Mathematisch Centrum, Amsterdam, 1983.
  13. V. Donzeau-Gouge, G. Huet, G. Kahn, and B. Lang, Programming environments based on structured editors: the Mentor experience, INRIA Research Report No. 26, 1980. Published in D.R. Barstow, H.E. Shrobe, and E. Sandewall (Eds.), *Interactive Programming Environments*, McGraw-Hill, 1984, pp. 128–140.
  14. Th. Despeyroux and V. Donzeau-Gouge, Typol: Introduction de spécifications sémantiques dans Mentor, INRIA Research Report, 1983 (Typol: Introduction of semantics specifications in Mentor—in French).
  15. Th. Despeyroux, Executable specification of static semantics, INRIA Research Report No. 295, 1984. Published in G. Kahn, D.B. MacQueen, and G. Plotkin (Eds.), *Semantics of Data Types*, LNCS Vol. 173, Springer, 1984, pp. 215–233.
  16. J. Heering, G. Kahn, P. Klint, and B. Lang, Generation of interactive programming environments, in The Commission of the European Communities (Eds.), *ESPRIT '85: Status Report of Continuing Work*, Part I, Elsevier Science Publishers, 1986, pp. 467–477.
  17. J. Heering and P. Klint, Work done at CWI/UvA—Final report, in: *Sixth Review Report ESPRIT Project 2177 (GIPE II)*, January 1994.



# Premo: An ISO Standard for a Presentation Environment for Multimedia Objects

*To Cor Baayen, at the occasion of his retirement*

I. Herman, P.J.W. ten Hagen, G. Reynolds

*CWI*

*Department of Interactive Systems*

*Kruislaan 413, 1098 SJ Amsterdam, The Netherlands*

*{ivan,paulh,reynolds}@cwi.nl*

PREMO is a major new ISO/IEC standard for graphics and multimedia, which addresses many of the concerns that have been expressed about existing graphics standards. In particular, it addresses the issues of configuration, extension, and interoperation of and between PREMO implementations. This paper gives an overview of PREMO and highlights its most significant features.

## 1 INTRODUCTION

The Graphical Kernel System GKS[1] was the first standard for computer graphics published by the International Organisation for Standardisation (ISO). It was followed by a series of complimentary standards, addressing different areas of computer graphics. Perhaps the best known of these are PHIGS[2], PHIGS PLUS[3], and CGM[4]. More recently, GKS[5] has been revised. These standardised functional specifications have had reasonable success either via direct implementations or through the influence they have had on the specification and development of other graphics packages (the most notable of this second category being the 3D extension of the X Window System, PEX[6, 7], which is largely based on PHIGS PLUS).

In spite of important differences in their functionality, these standards share a common architectural approach, which, although not a requirement defined within the documents, has resulted in implementations that are large monolithic libraries of a set of functions with precisely defined semantics. They reflect an approach towards graphical software libraries predominant in the seventies and the eighties. However, these standards have little chance of providing appropriate responses to the rapid changes in today's technology, and in

particular, they fail to fit into the software and hardware system architectures prevailing on today's systems.

The subcommittee responsible for the development and maintenance of graphics standards (ISO/IEC JTC1/SC24) recognised the need to develop a new line of graphics standards, along radically different lines from previous methods. To this end, a new project was started at an SC24 meeting at Chiemsee, Germany, in October 1992. Subsequent meetings (New Orleans, USA, January 1993; Steamboat Springs, USA, June 1993; Manchester, UK, November 1993; Amsterdam, The Netherlands, March 1994; Bordeaux, France, June 1994) resulted in a Draft for a new standard called PREMO (Presentation Environment for Multimedia Objects)[8]. This new work was approved by ISO/IEC JTC1 in February 1994, and is now a major ongoing activity in ISO/IEC JTC1/SC24/WG6.

The term "Presentation Environment" is of utmost importance in the specification of the scope of PREMO. PREMO, as well as the SC24 standards cited above, aims at providing a standard *programming* environment in a very general sense. The aim is to offer a standardised, hence conceptually portable, development environment that helps to promote portable graphics and multimedia applications. PREMO concentrates on *presentation techniques*; this is what primarily differentiates it from other multimedia standardisation projects.

One of the main differences between PREMO and previous standards within SC24 is the inclusion of multimedia aspects; hence this activity is of importance for both the multimedia and graphics communities. The purpose of this paper is to present the motivation behind the development of PREMO, its major goals, and its relationship to other multimedia standards. An overview of the architecture of PREMO is given, although much of the detail is still subject to changes that result from the technical review process within ISO.

## 2 MOTIVATION

Three requirements have shaped the architecture of PREMO:

- the appearance of new media;
- the need for configurable and extensible graphics packages;
- the requirements of distributed environments.

### 2.1 Incorporation of Various Media

Traditional computer graphics systems and graphics applications have primarily been concerned with what might be called the presentation of *synthetic graphics*, *i.e.*, displaying pictorial information, typically on a screen or paper. The aims of any two presentations may be very different. Two characteristic examples are:

- produce photorealistic images (*e.g.*, in commercial film production, or high quality animation) using very complex models describing the surrounding reality;
- produce ergonomically sound and easy-to-grasp images of complex computed or measured data (*e.g.*, in scientific visualisation, or medical imaging).

These aims determine different fields of interest within computer science, which are all referred to under the heading of “computer graphics” and which are all to be addressed by PREMO.

Developments over recent years have, however, resulted in new applications where synthetic graphics *in isolation* cannot cope with the requirements. Technology has made it possible to create systems which use, within the same application, different presentation techniques that are not necessarily related to synthetic graphics, *e.g.*, video, still images, and sound. Examples of applications where video output, sound, etc., and synthetic graphics (*e.g.*, animation) coexist are numerous and well-known. It is therefore a natural consequence to have development environments that are enriched with techniques supporting the display of different media in a consistent way, and which allow for the various media-specific presentation techniques to coexist within the same system.

“Coexistence” is not enough, though; *integration* is also necessary. For example, an audio display is not necessarily independent from the (synthetically generated) image being displayed: the viewer’s position in the model, or indeed the model itself when displayed, may influence the attributes of audio presentation. This influence may be very simple (*e.g.*, the volume may depend on the distance from the viewer), but it may also require very complicated sound processing techniques (*e.g.*, to take the acoustic properties of the room model into account for sound reflection and absorption). In other words, it should be possible to describe media objects *integrated* with geometry and with one another, and also to describe and control their mutual influence. The complete integration of various media and their presentation techniques within the same consistent framework is one of the major goals (and challenges) of PREMO, and one of the features which will make it very different from earlier SC24 standards, and indeed, other multimedia standards that are either already available or under development (such as HyTime[9], HyperODA[10], and MHEG[11]).

The introduction of new media brings new problems for PREMO that, hitherto, have been unknown in earlier SC24 standards. One of the most intricate issues of some importance is that of *synchronisation*, *e.g.*, synchronisation of video and sound presentation. This problem is well-known in the multimedia community; its integration with the more general demands of a presentation system will obviously be a challenge.

## 2.2 Configurable and Extensible Graphics Packages

As mentioned in §1, most traditional ISO graphics packages, as well as the majority of graphics systems available on the market-place, are defined as

monolithic libraries containing large sets of functions with precisely defined semantics. These libraries are frequently referred to as *kernels*. The choice of functionality for a specific kernel reflects the particular application areas which the kernel tries to address.

Modifying and extending the existing functionality of a kernel requires the definition of additional sets of functions. These functions may either add to or modify existing behaviour. However, modification of the standard interface is not allowed, which often means that these new definitions form completely separate packages on top of the standard with their own sets of well-defined functions.

This rigidity of current ISO graphics standards is in a sharp contrast with the extraordinary diversity of the algorithms used in computer graphics, in visualisation, and in other related application areas. Radically new visualisation techniques are developed, old and apparently well-established algorithms are constantly re-visited. This diversity and fervent activity is very well reflected in the proceedings of the major computer graphics and visualisation conferences worldwide (such as, for example, the ACM SIGGRAPH, Eurographics, and Nicograph annual conferences and workshops, IEEE's Visualisation conferences, etc.).

As a consequence, major rendering techniques, which are almost commonplace in advanced graphics applications, cannot be integrated into SC24 standards; the most startling examples being ray-tracing and radiosity. Although these graphics standards include a rudimentary mechanism to add new graphics primitives, for example in the form of the GDP, (Generalised Drawing Primitives), this mechanism does not give the full power needed by a number of applications to add new display algorithms and/or to modify some aspects of the ones included in the package in use<sup>1</sup>.

Note that the inclusion of different media into a new standard makes this type of problem more acute. The techniques to achieve integration of media are extremely disparate, and they use the results of various fields of computing technology, like, for example, high quality synthetic graphics, image processing, speech synthesis, etc. Some of the techniques are also application dependent. It is almost impossible to define a closed programming environment which would satisfactorily encompass all these needs; even if a specification could be finished, complete implementations would be so complex that the entire product would lag behind current technology.

The usual approach to solve such problems is to use object-oriented techniques. This is also the approach that has been adopted by PREMIO. Object-oriented techniques have already been used for graphics and for multimedia, and they have proven their values in using inheritance as a tool for extensibility and user configurability (see, *e.g.*, [12, 13, 14, 15, 16]). Using inheritance, additional information may be integrated into an existing object of a graphics system, allowing extensive reuse of inherited methods. Referring to the

---

<sup>1</sup>Escapes also offer some possibilities for modifying algorithms in a restricted way, but such extensions lead away from portability.



example above, in a carefully designed object-oriented system it would be possible to redefine the reflection equations of a “shader object” only, and thereby make full use of the power of the surrounding system with the shading method adapted for a particular use.

### 2.3 Distribution

It is no longer necessary to argue in favour of distributed environments; their widespread availability has made their use very natural in both academia and industry. Some graphics and multimedia applications and tools are notoriously computationally intensive, and as such are prime candidates to exploit the advantages offered by a distributed environment.

There have been numerous projects in the past which have tried to use, *e.g.*, GKS or PHIGS in a distributed setting; it was never easy. Indeed, the SC24 graphics standards were not particularly well prepared for distribution (see, for example, [17, 18, 19]). In contrast, and using the terminology which has become widespread in the past years, particular PREMO implementations may offer multimedia or graphics “services” on a network; hence, the PREMO specification should allow for the straightforward implementation of such services.

Object-oriented technology also provides a framework to describe distribution in a consistent manner. Objects can be considered as closed entities which provide “services” via their methods; from the point of view of the object specification it is immaterial how an object method is realised: within the same program, or via calls across a network.

Defining complex object-oriented systems to be used in a distributed environment leads to software engineering issues, whose complete solution would go far beyond the charter (and the experiences) of the PREMO working group. Instead, the PREMO specification will make use of techniques developed elsewhere, both within and outside ISO. Currently, another ISO working group (ISO/IEC JTC1/SC21 WG7) is working on what is called the “Open Distributed Processing Initiative” (ODP); PREMO intends to rely on the experiences of this working group, and include their results into the PREMO document proper. The goal is to develop a specification which would be compliant with ODP. A liaison agreement has also been set up with the Object Management Group<sup>2</sup> (OMG), whose CORBA specification[20] has already influenced the current design of PREMO.

## 3 GENERAL ARCHITECTURE

Underlying all of PREMO is a concise conceptual framework, comprising a description technique (not detailed here), an abstract object model used for the definition of data types and the operations upon them, and the notion of components which contain and organise the PREMO functionality needed to address specific problem areas.

---

<sup>2</sup>The Object Management Group is primarily an industrial consortium established to define a unifying model amongst/from a number of emerging object technologies.

### 3.1 The Conceptual Framework

The conceptual framework addresses three fundamental areas: an object model, the activity of objects, and events and event handling.

#### 3.1.1 Object Model

At the earliest stages of the PREMO project specification it became clear that a concise framework, *i.e.*, a precise *object model*, would be needed to ensure the smooth cooperation among objects within PREMO and also to provide a consistent approach to some of the technical issues raised by multimedia programming in general. Such an object model was adopted at an early stage of the PREMO project. This object model is traditional, being based on subtyping and inheritance. The PREMO object model supports both multiple supertypes and multiple inheritance.

As said earlier (c.f. §2.2), subtyping and inheritance provide the basic mechanism in PREMO for extensibility and configurability.

In PREMO, a strong emphasis is placed on the ability of objects to be active. This feature of PREMO stems from the need for synchronisation in multimedia environments (§2.1). Conceptually, different media (*e.g.*, a video sequence and a corresponding sound track) may be considered as parallel activities that have to reach specific milestones at distinct and possibly user definable synchronisation points. In many cases, specific media types may be directly supported in hardware. In some cases, using strictly specified synchronisation schemes, the underlying hardware can take care of synchronisation. However, a general object model should offer the capability of describing synchronisation in general terms as well (see also [14, 15, 16] for similar approaches taken in multimedia programming systems).

Allowing objects to be active does not contradict the OMG object model. However, some details of object requests have to be specified in more precise terms for PREMO, in contrast with the OMG object model. In PREMO, objects may define their operations as being *synchronous*, *asynchronous*, or *sampled*. The intuitive meaning of these notions is:

- If the operation is defined to be *synchronous*, the caller is suspended until the callee has serviced the request.
- If the operation is defined to be *asynchronous*, the caller is not suspended, and the service requests are queued on the callee's side. No return value is allowed in this case.
- If the operation is defined to be *sampled*, the caller is not suspended, but the service requests are not queued on the callee's side. Instead, the respective requests will overwrite one another as long as the callee has not serviced the request.

The unusual feature of this model, compared to traditional message passing protocols, is the introduction of sampled messages. Yet, this feature is not

unusual in computer graphics. Consider the well-known idea of sampling a logical input device, *e.g.*, locator position values. A separate object modelling (or directly interfacing) a locator can send thousands of motion notification messages to a receiver object, and this latter can just “sample” these messages using the sampled message facility.

Using active objects, synchronisation appears to be no more and no less than synchronisation of concurrent processes, *i.e.*, concurrent active objects in PREMO. This does *not* mean that synchronisation becomes easy. What it *does* mean is that the terminology, the results, the machinery, etc, of the theory and the practice of concurrent programming can be reused in PREMO. There are other issues of synchronisation that can be considered *quality of service* issues, which go beyond this basic synchronisation model. Nevertheless, the model provides a clean and straightforward framework on which other such facilities can be built.

### 3.1.2 Events, Event Model

The PREMO framework includes the notion of non-objects, primarily for efficiency reasons. Non-objects have no requests defined on them, they cannot take part in subtyping and inheritance hierarchies.

*Events* form a special category of PREMO non-object types, and are the basic building block for the PREMO event model. Events and their propagation (described by the event model) play a fundamental role in the synchronisation mechanism.

The event model is based on three concepts: events, event registration, and event handling. An *event* can model any action that occurs at a definite time. Events are created by *event sources*, and are consumed by *event clients*, both of which are objects. A basic characteristic of an event is its distinct type, which is one of the characteristics that a client uses to identify the events in which it is interested.

Whereas in object communication, the caller specifies the recipient of each operation request, in event communication, events are not addressed to specific recipients. Instead, it is the recipient that determines which events it wishes to receive. An object can register interest in receiving specific events produced by the various objects. As part of the registration process, a client can specify one of its (asynchronous or sampled) methods to receive events forwarded by an **Event Handler** object (defined by the so-called Fundamental Component, see §4.1). Prospective event recipients specify which events they are interested in by registering constraint lists with an **Event Handler**. Each constraint list defines the event names and parameter values which the event recipient wishes to receive. In the most common case, the constraint list specifies the name of an event in which the object is interested. Issuing an event by the event source means sending a message to an **Event Handler** object which dispatches the event to the interested event clients.

### 3.2 Components

The object model, the event model, the concept of non-objects, etc., described in §3.1, give a conceptual framework for all the basic notions in PREMIO. *Components* allow for a structuring of the PREMIO standard in terms of the services provided.

A component in PREMIO is a collection of object types and non-object data types, from which objects and non-objects can be instantiated. Objects within one component are designed for a close cooperation and offer a well-defined set of functional capabilities for use by other objects external to the component. A component can offer *services* as in OMG (see §2.3), *i.e.*, services usable in a distributed environment, or it may be used as a set of objects directly linked to an application.

Components may be organised in component inheritance hierarchies. For example, in Figure 1, both components B and C inherit from component A. This means that object types in B and C are subtypes of types defined in A (see §3.1.1). All PREMIO objects are subtypes of a common PREMIO supertype, so this rule enables new types of objects to be defined. As far as subtyping and/or inheritance are concerned, objects within components B and C are all distinct types: no type in B may be a subtype of a type in C and vice versa.

The rule on component inheritance does not imply that objects in different components have to have a subtyping relationship in order to be able to communicate with one another. Again referring to Figure 1, B can of course make use of the *services* offered by component C. Components may also specify how they exploit functionality from other components, with the option of hiding this from the client. Hence components may become clients of other components' services.

Underlying all PREMIO components is a *Foundation Component* providing functionality which is necessary for all PREMIO components. It is mandatory that all other PREMIO components inherit from this Foundation Component (described in more details in §4.1).

The rules for components are part of the standard. These rules form the basis, in conjunction with the object model, for the properties of configuration, customisation, extension, and interoperation.

## 4 COMPONENT STRUCTURE

With the above description of the conceptual framework and the component model, we now describe the structure of the PREMIO standard in more detail.

The initial PREMIO standard will:

- define the exact conceptual framework for multimedia presentation, along the lines described in §3.1, *i.e.*, the object model, the event model, etc.;
- define rules for components, their interrelationships, inheritance, conformance rules, etc.;

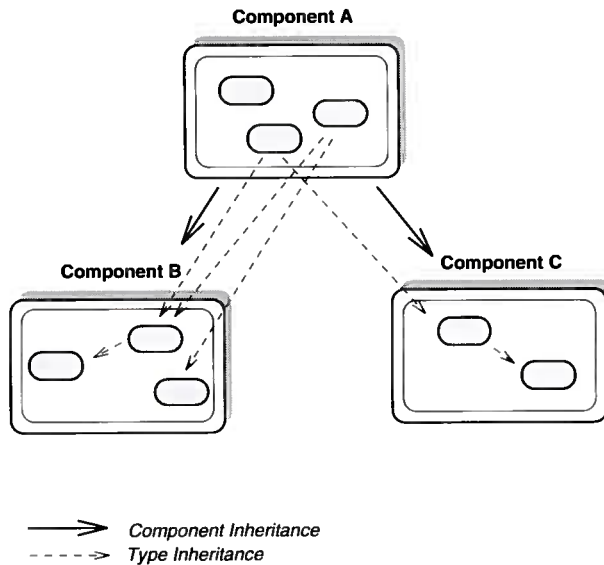


FIGURE 1. Component inheritance.

- include the specification of the Foundation Component;
- include the specification of some other components, namely:
  - a component for Multimedia System Services (see §4.2);
  - a Modelling, Presentation, and Interaction Component, which will provide for the basis of components inherently related to modelling, geometry, traditional computer graphics, etc.

PREMO should, however, be thought of as an evolving standard; new components will be added in the future. On the basis of the Modelling, Presentation, and Interaction Component, components may also be added to ensure applications using current SC24 standards will continue to work, and be upwards compatible. Two types of components are planned: expression of existing SC24 standards as PREMO components, *e.g.*, PHIGS or GKS, or new components, *e.g.*, a pure audio component, or a component for virtual reality. Although the exact component hierarchy is not yet finalised (June 1994), Figure 2 gives a view of the expected hierarchy of standardised components.

In the following sections, highlights of some of the components referred to above are given. The reader should remember, however, that the specification of these components is still an ongoing activity.

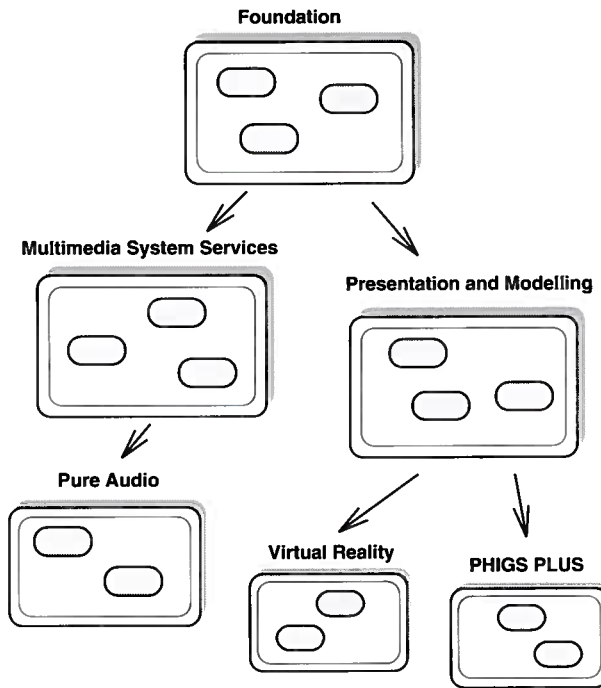


FIGURE 2. Component hierarchy.

#### 4.1 Foundation Component

The foundation component is a collection of *foundation objects*. Foundation objects are those which support a fundamental set of services suitable for use by a wide variety of other components.

It is beyond the scope of this paper to give an exhaustive specification of all foundation objects defined in the foundation component; only some highlights are given here. The list of foundation objects includes the following object types:

- The **PREMO Life-cycle Manager** object provides object life cycle services for PREMO objects. This includes the creation of new objects, destruction of object and object references, keeping track of object references. The separate management of object life-cycles and associated object references is essential if a component intends to offer services in a distributed environment.

In fact, PREMO defines two such life cycle manager objects, whose functionalities are identical, but they manage remote, service objects and local object respectively. This distinction is necessary to control objects which

offer services over, e.g., a distributed environment and, alternatively, to have objects which are to be used in a local setting only.

- **Data** objects. The semantics associated with a data object define the construction and modification interface of a particular data object. Examples are geometric 2D or 3D points, colour, matrices, with related operations and other attributes, video frames, frequency spectra, etc.
- **Producer** objects provide an encapsulation for defining the processing of **Data** objects and the production of refined or transmuted **Data** objects. **Producer** objects may receive **Data** objects from any number of sources and deliver **Data** objects to any number of destinations. Specific subtypes of type **Producer** may place restrictions on the number of sources and destinations of **Data** objects if necessary. Specific types of **Producer** object are characterised by the behaviour made visible through their associated sets of operations.
- A **Porter** object is the PREMIO foundation object which interconnects to systems and environments defined outside of PREMIO, e.g., files, physical devices.
- The role of a **Controller** is to coordinate cooperation among objects. A **Controller** object is an autonomous and programmable finite state machine (FSM). Transitions are triggered by messages sent by other objects. Actions of the FSM correspond to messages sent to other objects. The actions of a **Controller** object may cause messages to be sent to other **Controller** objects, thus a hierarchy of **Controllers** can be defined.
- **Event Handler** objects provide methods to register interest in certain events, for dispatching events to the interested objects, manage constraint lists for events, etc. These objects also play a fundamental role in synchronisation mechanisms.

As an example of how these notions can be used, let us see how basic, event-based, synchronisation can be expressed with these objects. Synchronisation is handled by using synchronisation events that are sent by synchronisation sources to event handlers. An **Event Handler** then forwards the event to objects that have registered their interest in these events. The interested objects could be either objects that are the immediate target in the synchronisation, or controller objects for more elaborate synchronisation. Figure 3 illustrates a more complex case: two **Event Handlers** take care of two independent clock events, but, for one of them, the same event may also be “simulated” by another object. A separate controller receives these events and, based on its own internal state, may then dispatch a synchronisation call to two other PREMIO objects.

The combination of **Event Handlers** and **Controllers** can also be used for schemes where actions are scheduled to take place at a certain time. In this



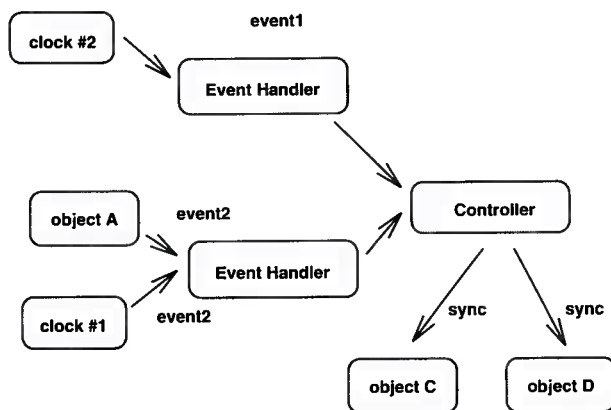


FIGURE 3. Synchronisation.

case, a clock object (to be provided by a higher-level component) can be used to trigger the action at the right time. This allows for the more general notion of temporal synchronisation.

#### 4.2 Multimedia System Services

The primary goal of the Multimedia System Services (MSS), defined as a recommended practice by the IMA (Interactive Multimedia Association), is to provide an infrastructure for building multimedia computing platforms that support interactive multimedia applications dealing with *synchronised, time-based*, media in a heterogeneous distributed environment. The emphasis is very much on distributed services for “low level” media processing; MSS does *not* include any concepts for geometry, modelling, etc. Instead, it is concerned with problems like the definition of abstract media devices, resource control, connections among virtual devices (in the form of so-called streams), etc.

Active cooperation between the ISO PREMIO group and IMA and resulted in the decision encapsulate MSS within PREMIO. Figure 2 shows how MSS will be integrated into PREMIO: it will form a separate component, relying on the objects defined in the Foundation Component. The design of these objects already reflects the requirements of MSS. A first implementation of MSS will be available (independently of PREMIO) in the course of 1995, and the first draft for an integration with PREMIO will be available in 1996.

#### 4.3 Presentation, Modelling, and Interaction Component

The Presentation, Modelling, and Interaction Component (PMI) of PREMIO combines media control with modelling and geometry. This is an abstract component from which concrete modelling and presentation components are



expected to be derived. Thus, for example, a virtual reality component that is derived, at least in part, from the Presentation, Modelling, and Interaction Component, might refine the renderer objects of the PMI component to objects most appropriate in the virtual reality domain. This component introduces abstractions for such things as modellers, modelling objects and their properties, scenes, renderers, etc. Objects with geometry may be placed into scenes, and may subsequently be transformed and visualised. This notion is a general one and applies equally well to objects that do not have a clear graphical representation. For example, an audio object with spatial properties can be located within a scene and appropriate rendering algorithms can take this into account to achieve a stereo audio effect. The abstractions defined in this component will also allow for the inclusion of objects with time properties.

The Presentation, Modelling, and Interaction Component of PREMO heavily relies on an existing reference model, called the Computer Graphics Reference Model (CGRM)[21], developed within the same ISO group (ISO/IEC JTC1/SC24) some years ago. In fact, the PMI could be viewed as the adaptation of CGRM (which is an abstract framework) to the object oriented environment defined by PREMO

Based on the Presentation, Modelling, and Interaction Component, more "concrete" components will be developed. Activities have already started on the development of a Virtual Reality component, and other possibilities (*e.g.*, pure audio component, solid modelling component) are currently explored.

## 5 A FORMAL APPROACH TO DEVELOPING THE PREMO STANDARD

The graphics standards community have in the past employed formal methods in only a very limited sense. The semantics of first generation graphics standards, such as GKS and PHIGS, were described using natural language, and in some cases this has meant that ambiguities have crept into the specifications. The PREMO RG plans to address this problem by employing formal methods at an early stage and to continue this activity throughout PREMO's development. This task started after the July 1993 PREMO meeting and some early results are documented in [22, 23]. The intention is to provide a formal specification of the PREMO object model and some of its components, where the main emphasis is placed on feeding results back into the standard's development. This is essentially a complimentary activity and it is not currently planned that this should replace the usual natural language description. The formalism used is based on Z[24] and Object-Z[25].

## 6 TIMETABLE

The current timetable for the work progress in PREMO is as follows:

Draft International Standard:	June 1996
International Standard final text:	June 1997

## 7 EXPERIMENTAL IMPLEMENTATIONS

In the near future, work will also begin on an experimental implementation of the PREMO standard. The major emphasis of this work will be to provide a proof of concepts for the main paradigms and the models advocated by the PREMO document. The implementation of the object model will require a major effort; indeed, the requirements of this model go far beyond what is offered “by default” by languages like C++[26]. Fortunately, tools already exist which will make this activity easier. The environment developed within the ESPRIT MADE project[27], primarily its object model implementation[16], will be used as the basic tools to develop a first, experimental implementation of PREMO. <sup>3</sup>

## ACKNOWLEDGEMENTS

Obviously, PREMO is a teamwork project, involving a large number of experts from a number of industrial and academic institutions involved in ISO/IEC JTC1/SC24/WG6. Instead of trying to list everybody and thereby incurring the danger of forgetting and perhaps offending somebody, we prefer to omit such a long list. We would just like to express our gratitude to all the members of the ISO/IEC JTC1/SC24/WG6 rapporteur group.

## REFERENCES

1. International Organisation for Standardisation, Geneva. *Information processing systems — Computer graphics — Graphical Kernel System (GKS) functional description (ISO IS 7942)*, 1985.
2. International Organisation for Standardisation, Geneva. *Information processing systems — Computer graphics — Programmer’s Hierarchical Interactive Graphics System (PHIGS) (ISO IS 9592)*, 1988.
3. International Organisation for Standardisation. *Information processing systems — Computer graphics — Programmer’s Hierarchical Interactive Graphics System (PHIGS) — Part 4, Plus Lumière und Surfaces (PHIGS PLUS) (ISO DIS 9592-4)*, 1991.
4. International Organisation for Standardisation, Geneva. *Information processing systems — Computer graphics — Metafile for the storage and transfer of picture description information (ISO IS 8632)*, 1987.
5. International Organization for Standardisation, Geneva. *Information processing systems — Computer graphics — Graphical Kernel System (GKS) functional description (ISO/IEC 7942-1:1994)*, 1994.
6. W. Clifford, J. McConnell, and J. Saltz, “The development of PEX,” in *Eurographics’88 Conference Proceedings* (D. Duce and P. Jancène, eds.), (Amsterdam), North-Holland, 1988.

---

<sup>3</sup>Note that elements of the MADE object model, *e.g.*, the notion of sampled messages, have already significantly influenced the development of the PREMO object model.

7. R. Rost, J. Friedberg, and P. Nishimoto, "PEX: A network-transparent 3D graphics system," *IEEE Computer Graphics & Applications*, vol. 9, pp. 14–25, 1989.
8. International Organisation for Standardisation, *Presentation Environment for Multimedia Objects (PREMO)*; ISO/IEC 14478, June 1994.
9. International Organisation for Standardisation, *Information Technology — Hypermedia/Time-based Structuring Language (HyTime)*, ISO/IEC 10744:1992(E), 1992.
10. International Organisation for Standardisation, Geneva, *Information technology — Open Document Architecture (ODA) and Interchange Format — Temporal relationships and non-linear structures (ISO/IEC DIS 8613-14:1993)*, 1993.
11. International Organisation for Standardisation, *Information Technology — Coded Representation of Multimedia and Hypermedia Information Objects (MHEG)*, ISO/IEC CD 13522 ed., June 1993.
12. P. Wißkirchen and K. Kansy, "The new graphics standard — object oriented!," in *Advances in Object-Oriented Graphics I* (E. Blake and P. Wißkirchen, eds.), EurographicSeminar Series, Berlin – Heidelberg – New York – Tokyo: Springer-Verlag, 1991.
13. M. Kaplan, "The design of the Doré system," in *Advances in Object-Oriented Graphics I* (E. Blake and P. Wißkirchen, eds.), EurographicSeminar Series, Berlin – Heidelberg – New York – Tokyo: Springer-Verlag, 1991.
14. V. de May, C. Breiteneder, L. Dami, S. Gibbs, and D. Tschritzis, "Visual composition and Multimedia," *Computer Graphics Forum (Eurographics'92)*, vol. 11, no. 3, pp. C9–C21, 1992.
15. V. de May and S. Gibbs, "A multimedia component kit," in *Proceedings of ACM Multimedia'93* (P. Rangan, ed.), (Anaheim, CA), pp. 291–300, ACM Press, August 1993.
16. F. Arbab, I. Herman, and G. Reynolds, "An object model for multimedia programming," *Computer Graphics Forum (Eurographics'93 Conference Issue)*, vol. 12, pp. C101–C114, September 1993.
17. I. Herman, T. Tolnay-Knefely, and A. Vincze, "XGKS — a multitask implementation of GKS," *Computers and Graphics*, vol. 8, 1984.
18. G. Reynolds, "A token based graphics system," *Computer Graphics Forum*, vol. 5, pp. 139–145, June 1986.
19. D. Arnold and M. Hinds, "On implementing parallel GKS," *Computer Graphics Forum*, vol. 8, 1989.
20. Object Management Group, *The Common Object Request Broker: Architecture and Specification; OMG Document Number 91.12.1, Revision 1.1*, 1992.
21. International Organisation for Standardisation, *Introduction to the Computer Graphics Reference Model*, ISO/IEC JTC 1/SC24 N849, 1992.
22. International Organization for Standardisation, Geneva, *Report of the ISO/IEC JTC1/SC24 Special Rapporteur Group on Formal Description Techniques*, 1994.

23. D. Duce, D. Duke, P. ten Hagen, and G. Reynolds, "PREMO – an initial approach to a formal definition," *Computer Graphics Forum (Eurographics'94 Conference Issue)*, vol. 13, pp. C393–C406, September 1994.
24. B. Potter, J. Sinclair, and D. Till, *An Introduction to Formal Specification and Z*. International Series in Computer Science, New York London Toronto Sydney Tokyo Singapore: Prentice Hall, 1991.
25. R. Duke, P. King, G. Rose, and G. Smith, "The Object-Z specification language: Version 1," Tech. Rep. 91-1, The University of Queensland, Queensland, Australia, April 1991.
26. B. Stroustrup, *The C++ Programming Language*. Reading, Massachusetts: Addison-Wesley, second ed., 1991.
27. I. Herman, G. Reynolds, and J. Davy, "MADE: A multimedia application development environment," in *Proc. of the IEEE International Conference on Multimedia Computing and Systems, Boston (ICMCS'94)* (L. Belady, S. Stevens, and R. Steinmetz, eds.), (Los Alamitos), IEEE CS Press, 1994.

# On the History of Runge-Kutta Methods

*To Cor Baayen at the occasion of his retirement*

P.J. van der Houwen

Runge-Kutta methods are widely-used methods for the integration of initial-value problems for ordinary differential equations. They can also be used for the time integration of initial-value problems for time-dependent partial differential equations by applying the so-called Method of Lines. The method of lines transforms the partial differential equation into a system of ordinary differential equations by discretization of the space variables, so that formally any ordinary differential equation solver can be employed for the time integration of the resulting initial-value problem. However, since ordinary differential equations originating from space-discretized partial differential equations have a special structure, not every ordinary differential equation solver is appropriate. For example, the well-known fourth-order Runge-Kutta method is highly inefficient if the partial differential equation is parabolic, but it performs often quite satisfactory if the partial differential equation is hyperbolic. In this contribution, we concentrate on the role played by Runge-Kutta methods in the numerical integration of time-dependent partial differential equations. In particular, we shall describe the research carried out at CWI.

## 1. THE METHOD OF LINES

The method of lines transforms initial-boundary value problems for time-dependent partial differential equations (PDEs) into initial-value problems (IVPs) for systems of ordinary differential equations (ODEs). This is achieved by discretization of the space variables using finite difference, finite element or finite volume approximations. The connection of PDEs with systems of ODEs was already known to Lagrange (see the historical notes in the book of Hairer-Nørsett-Wanner [10, p. 25]). In 1759 Lagrange already observed that his mathematical model for the propagation of sound in terms of a system of second-order ODEs is related to d'Alembert's equation  $u_{tt} = u_{xx}$  for the vibrating string. However, the actual use of

the space-discretized approximation in numerically solving initial-boundary value problems for PDEs seems to start with Rothe in 1930 [32], and is therefore also called Rothe's method (see [10, p. 3]).

In this paper, we shall restrict our considerations to the case where the spatial discretization of the PDE leads to an IVP of the form

$$(1.1) \quad \frac{dy(t)}{dt} = f(t, y(t)), \quad y(t_0) = y_0,$$

where  $t$  is the time variable and  $y_0$  contains the given initial values. Notice that the boundary conditions are lumped into the righthand side function  $f$ .

The IVP (1.1) has a number of specific characteristics that play a crucial role in selecting a suitable integrator. Firstly, the system (1.1) can be extremely large, particularly, if it originates from a problem with 2 or 3 spatial dimensions. Secondly, the system is usually extremely stiff (here, (1.1) is considered to be stiff if the solution components corresponding to eigenvalues of the Jacobian  $\partial f / \partial y$  that are close to the origin are dominating). Thirdly, the required order of accuracy in time is rather modest (usually not exceeding the order of the spatial discretization, that is, at most order three). Hence, we are led to look for low-order, stiff ODEIVP solvers that are storage economic.

One approach is to look for conventional, general purpose ODEIVP methods that meet these requirements. There are two often used integrators, the second-order trapezoidal rule and the first-order backward Euler method, that belong both to the class of Runge-Kutta methods. They were respectively used by Crank and Nicolson [4] and by Laasonen [24] in their papers of 1947 and 1949 for solving heat flow problems. In the PDE literature, these methods also known as the *Crank-Nicolson* and *Laasonen methods*. An integration method that combines the second-order accuracy of the Crank-Nicolson method and the high stability of the Laasonen method is offered by the two-step method based on backward differentiation (known as the *BDF2 method*). BDF methods were proposed in 1952 by Curtiss and Hirschfelder [3] for solving stiff ODEs and became popular by the papers of Gear in 1967-1968, and in particular by his book [7] of 1971. The Crank-Nicolson, Laasonen and BDF2 methods are applicable to a wide class of space-discretized PDEs (not only heat flow problems) and have comparable computational complexity. In order to solve the implicit relations, one usually applies Newton iteration which leads to a large linear system in each iteration. For one-dimensional problems, these linear systems can be solved by *direct* methods that are in general highly efficient because the band structure of the system can be fully exploited. However, in more than one spatial dimension, direct solution methods usually are out of the question and we have to resort to an *iterative* method. If  $L_N$  denotes the number of Newton iterations,  $L_S$  the number of linear system iterations,  $d$  the spatial dimension, and  $\Delta$  the spatial grid size, then the computational complexity of these methods is  $O(L_N L_S \Delta^{-d})$ . Often used linear-system-iteration methods are conjugate gradient type methods that require at least  $O(\Delta^{-1/2})$  iterations. Hence, the total computational work involved for integrating the unit time interval with stepsize  $h$  is at least  $W = O(L_S h^{-1} \Delta^{-d-1/2})$ .

In order to reduce the huge amount of work when integrating higher-dimensional problems, new methods have been developed. The remainder of this paper will be

devoted to such methods. Since it is not feasible to present a complete survey, we shall confine ourselves to Runge-Kutta type methods that are tuned to PDEs in two or more spatial dimensions. We shall discuss explicit RK methods for parabolic and hyperbolic problems (spectrum of the Jacobian  $\partial f / \partial y$  along the negative axis and imaginary axis, respectively), and splitting methods represented as RK methods with fractional stages.

## 2. EXPLICIT RUNGE-KUTTA METHODS

Consider the  $s$ -stage RK method

$$(2.1) \quad \mathbf{Y} = \mathbf{e} \otimes \mathbf{y}_n + h (\mathbf{A} \otimes \mathbf{I}) \mathbf{F}(t_n \mathbf{e} + ch, \mathbf{Y}),$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + h (\mathbf{b}^T \otimes \mathbf{I}) \mathbf{F}(t_n \mathbf{e} + ch, \mathbf{Y}),$$

where  $h$  is the integration step,  $\mathbf{y}_n$  and  $\mathbf{y}_{n+1}$  represent approximations to the exact solution vector  $\mathbf{y}(t)$  at  $t = t_n$  and  $t = t_{n+1}$ ,  $\otimes$  denotes the Kronecker product, the  $s$ -dimensional vector  $\mathbf{e}$  is the vector with unit entries,  $\mathbf{I}$  is the identity matrix whose dimension equals that of the IVP, and the  $s$ -by- $s$  matrix  $\mathbf{A}$  and the  $s$ -dimensional vectors  $\mathbf{b}$  and  $\mathbf{c} := \mathbf{A}\mathbf{e}$  contain the RK parameters. The  $s$  components  $Y_i$  of  $\mathbf{Y}$  represent intermediate approximations to the exact solution values  $y(t_n + c_i h)$  and  $\mathbf{F}(t_n \mathbf{e} + ch, \mathbf{Y})$  contains the derivative values  $(\mathbf{f}(t_n + c_i h, Y_i))$ . In the following, the dimensions of  $\mathbf{e}$  and  $\mathbf{I}$  may vary, but will always be clear from the context in which they appear.

If  $\mathbf{A}$  is strictly lower triangular, then (2.1) defines an *explicit* RK method (the first method of this type was proposed by Runge [33] about 100 years ago). Explicit RK methods are relatively cheap, provided that the integration step  $h$  can be chosen sufficiently large. For stiff ODEs, the step is restricted by a stability condition of the form

$$(2.2) \quad h < \frac{\beta}{\rho(J_n)}, \quad J_n := \frac{\partial \mathbf{f}(t_n, \mathbf{y}_n)}{\partial \mathbf{y}},$$

where  $\rho(J_n)$  is the spectral radius of  $J_n$  and  $\beta$  is the so-called stability boundary. In the case of *parabolic* and *hyperbolic* problems, where the Jacobian of the righthand side function respectively has (more or less) negative and imaginary eigenvalues,  $\beta$  denotes the *real* stability boundary  $\beta_{\text{real}}$  or the *imaginary* stability boundary  $\beta_{\text{imag}}$  of the RK method. The real stability boundary is defined by the maximum length of the negative interval  $(-\beta, 0)$  that is contained in the region where the *stability polynomial*  $R_s(z) := \mathbf{1} + \mathbf{b}^T (\mathbf{I} - z\mathbf{A})^{-1} \mathbf{e}$  assumes values within the unit circle. Similarly, the imaginary stability boundary is defined by the maximum length of the interval  $(0, i\beta)$  where  $R_s$  is bounded by 1.

### 2.1. Conventional RK methods

For conventional RK methods,  $R_s(z)$  is given by the Taylor polynomial of degree  $s$  in  $z$ , that is, the polynomial that coincides with the truncated Taylor expansion of

$\exp(z)$  at  $z = 0$ . Let us first consider the parabolic case. The *real* stability boundary of Taylor polynomials is (approximately) given by (cf. [15, p.236] and [20,21])

$$(2.3) \quad \beta_{\text{real}} \approx 0.368 (s+1) \sqrt{19(s+1)} .$$

This approximation is already quite close for  $s \geq 4$ . We conclude from (2.2) and (2.3) that we can take any step we want by choosing  $s$  sufficiently large, but these formulas also show that for large  $s$  the total number of function calls needed for integrating the unit interval with maximum step  $h = \beta_{\text{real}} \rho^{-1}(J_N) \approx 0.368 s \rho^{-1}(J_N)$  is given by  $N_f \approx 2.7\rho(J_N)$ , that is, *independent of  $s$* . Hence, conventional RK methods are as costly as the explicit Euler method (but of course highly accurate as  $s$  increases). Since  $f$  has  $O(\Delta^{-d})$  components and since for parabolic problems  $\rho(J_N) = O(\Delta^{-2})$ , the computational work can be estimated by  $W = O(\Delta^{-d-2})$ . This differs by a factor of order  $O(h\Delta^{-3/2})$  from the estimate derived for the Crank-Nicolson, Laasonen and BDF2 methods (when applied to higher-dimensional problems). Usually, this factor is quite large (e.g. if  $h = O(\Delta)$ ), so that conventional RK methods are not the way to solve space-discretized PDEs of *parabolic* type. They are "too costly and too accurate".

Next we consider the hyperbolic case. It happens that for the imaginary stability boundary  $\beta_{\text{imag}}$  we do not always obtain nonzero values. If  $z^{-(p+1)}[R_s(z) - \exp(z)] \rightarrow C_{p+1}$  as  $z \rightarrow 0$ , where  $p$  denotes the order of accuracy of the RK method, then it can straightforwardly be shown that  $\beta_{\text{imag}}$  is only nonzero if either  $C_{p+1} i^p < 0$  for  $p$  even or  $C_{p+1} i^{p+1} < 0$  for  $p$  odd. For the Taylor polynomials this implies that the imaginary stability interval is empty for  $p = 1, 2, 5, 6, 9, 10, \dots$ . For the other orders, quite reasonable values are obtained. For example, for  $p = 3, 4, 7, 8$ , we have  $\beta_{\text{imag}} \approx 1.7, 2.8, 1.7, 3.4$ . Taking one of these latter methods and assuming that  $\rho(J_N) = O(\Delta^{-1})$ , the total computational work associated with the unit interval can be estimated by  $W = O(\Delta^{-d-1})$ . This is a factor of order  $O(h\Delta^{-1/2})$  better than the estimate derived for the Crank-Nicolson, Laasonen and BDF2 methods. Hence, unlike the situation for parabolic problems, conventional RK methods seem to be preferable for *hyperbolic* problems.

## 2.2. Parabolic RK methods

Our conclusion that for parabolic problems explicit RK methods are "too costly and too accurate" suggests sacrificing accuracy in order to reduce computational costs. By observing that an  $s$ -stage RK method of order  $p$  possesses a stability polynomial  $R_s$  of the form

$$(2.4) \quad R_s^{(p)}(z) := \beta_0 + \beta_1 z + \beta_2 z^2 + \dots + \beta_s z^s, \quad \beta_j = \frac{1}{j!}; \quad j = 0, \dots, p; \quad s > p,$$

where the coefficients  $\beta_j, j = p+1, \dots, s$ , are free parameters, it is natural to use these free parameters for obtaining larger stability boundaries. For parabolic problems, where the eigenvalues of the Jacobian often are along the negative axis, we are led to construct polynomials  $R_s^{(p)}(z)$  with increased *real* stability boundary. Having found an appropriate stability polynomial  $R_s^{(p)}$ , it is always possible to



construct an RK method with  $R_s^{(p)}$  as its stability polynomial (see e.g. [15]). Such methods will be called *parabolic RK methods*.

Until now, closed form solutions for the polynomials with maximal real stability boundaries (to be called *optimal polynomials*) are only known for  $p = 1$ . They are given by the shifted Chebyshev polynomials

$$(2.5) \quad C_s^{(1)}(z) := T_s\left(1 + \frac{z}{s^2}\right), \quad \beta_{\text{real}} = 2s^2,$$

where  $T_s(z) := \cos(s \arccos(z))$  denotes the first kind Chebyshev polynomial of degree  $s$ . They have been rediscovered in the literature again and again (even in recent years, see e.g. [2]). As far as I know, they were first mentioned for integrating parabolic equations: in 1958 by Yuan' Chzao-Din in his thesis [41], in 1959 by Franklin in his paper [6] that appeared in the Journal for Mathematical Physics, and in 1960 by Guillou and Lago in the Proceedings [9] of the first conference of AFCAL (the French Association for Computing). These authors were not aware of each others work.

For  $p \geq 2$ , only approximate solutions have been constructed. In the thesis of Metzger [28] in 1967, we find numerical approximations for  $p \leq 4$ ,  $s \leq 5$ , and in a NASA report of Lomax [27] of 1968, a general approach for computing the coefficients was indicated. Lomax conjectured that the optimal polynomials satisfy the so-called *equal ripple* property, that is, the optimal polynomial has  $s-p$  local extrema  $+1$  or  $-1$  (this property was actually proved by Riha [31] in 1972 who also showed the unique existence of the optimal polynomials for all  $p$  and all  $s > p$ ). Using the equal ripple property, an iterative method can be constructed for the numerical computation of the coefficients. However, this equal-ripple-iteration method needs rather accurate initial iterates in order to converge. Presumably for this reason, Lomax did not use the equal-ripple-property approach, and instead, computed least squares approximations for  $p = 2$  and  $s \leq 10$ . Again, Metzger, Lomax and Riha found their results independently.

At CWI we used the least squares approach of Lomax for generating initial iterates to start the equal-ripple-iteration method. In this way, we computed the optimal stability polynomials, together with their real stability boundaries, for  $p \leq 4$  and  $s \leq 10+p$  (tables for the coefficients can be found in [13,14]). These computations indicated that  $\beta_{\text{real}}$  increases quadratically with  $s$  as  $s$  increases. In fact, we found

$$(2.6) \quad \beta_{\text{real}} = \gamma_p s^2 \quad \text{as } s \rightarrow \infty, \quad \gamma_2 = 0.814, \quad \gamma_3 = 0.489, \quad \gamma_4 = 0.341.$$

The quadratic behaviour is important. It implies that the total number of function calls needed for integrating the unit interval with maximum step  $h = \gamma_p s^2 \rho^{-1}(J_n)$  is now given by  $N_f = (\gamma_p s)^{-1} \rho(J_n)$ , which is a factor  $2.7 \gamma_p s$  less than the number of function calls needed for conventional RK methods. Hence, for large values of  $s$ , RK methods generated by (2.5) are much cheaper than conventional RK methods, provided that they are available for large values of  $s$ . Unfortunately, the numerical computation of the optimal polynomials becomes increasingly more difficult as  $s$  increases. This motivated us to look for analytical expressions for nearly optimal polynomials that are valid for arbitrary high values of  $s$ . In 1971, Bakker [1] derived in his Master thesis for  $p = 2$  and  $p = 3$  analytically given polynomials which are

quite close approximations to the optimal stability polynomials, in the sense that the stability boundaries are close to the maximal attainable values. These polynomials, to be called the *Bakker polynomials*, are given by

$$(2.7) \quad B_s^{(2)}(z) = \frac{2s^2 + 1}{3s^2} + \frac{s^2 - 1}{3s^2} T_s\left(1 + \frac{3z}{s^2 - 1}\right), \quad \beta_{\text{real}} \approx \frac{2}{3} (s^2 - 1), \quad s > 2,$$

$$(2.8) \quad B_s^{(3)}(z) = 1 + \frac{3\beta^2 - 2(40k^2 - 1)\beta}{576k^4} - \frac{3\beta^2 - 2(36k^2 - 1)\beta}{512k^4} T_{2k}\left(1 + \frac{2z}{\beta}\right) \\ + \frac{3\beta^2 - 2(4k^2 - 1)\beta}{4608k^4} T_s\left(1 + \frac{2z}{\beta}\right), \quad k := \frac{s}{6}, \quad s = 6, 12, 18, \dots,$$

$$\beta_{\text{real}} = \beta := \frac{2}{9} s^2 - 1 + \frac{1}{9} \sqrt{\frac{8s^4 - 60s^2 + 297}{5}} \\ \approx \frac{2}{9} s^2 \left(1 + \sqrt{\frac{2}{5}}\right) \approx 0.363 s^2 \text{ as } s \rightarrow \infty,$$

where again  $T_s$  denotes the first kind Chebyshev polynomial of degree  $s$  (in addition, Bakker actually proved the quadratic behaviour of the real stability boundaries of the optimal polynomials and obtained lower and upper bounds for  $\gamma_p$  up to  $p = 15$ ). A comparison of (2.6) with (2.7) and (2.8) reveals that the Bakker polynomials respectively possess 80% and 75% of the maximal attainable, asymptotic stability boundary. Later on in 1982, we found for  $p = 2$  an even better approximation given by (cf. [17])

$$(2.9) \quad A_s^{(2)}(z) = \frac{2}{2 - z} - \frac{z}{2 - z} T_s\left(\cos(\pi/s) + \frac{1 - \cos(\pi/s)}{z}\right), \\ \beta_{\text{real}} = \frac{2}{[\tan(\pi/2s)]^2} \approx 8 \frac{s^2}{\pi^2} \approx 0.810 s^2 \text{ as } s \rightarrow \infty.$$

These polynomials are not the optimal ones, but yield 99.5 % of the maximal attainable, asymptotic stability boundary!

The parabolic RK methods generated by the analytically given polynomials (2.5), (2.7) and (2.9) enable us to select an integration step  $h$  on the basis of accuracy considerations and to adapt the number of stages according to the stability condition  $s \approx (\gamma_p^{-1} h p(J_n))^{1/2}$ . Hence, effectively, we have an *unconditionally stable* method. As we remarked earlier, given the stability polynomial, many RK methods possessing this stability polynomial are possible. One of the most simple implementations of first-order or second-order RK methods with stability polynomial (2.4) reads

$$\begin{aligned}
 \mathbf{Y}_i &= \mathbf{y}_n + a_i h \mathbf{f}(t_n + c_{i-1}h, \mathbf{Y}_{i-1}), \quad a_i := \frac{\beta_{s-i+2}}{\beta_{s-i+1}}, \quad i = 1, \dots, s, \\
 (2.10) \quad \mathbf{y}_{n+1} &= \mathbf{y}_n + h \mathbf{f}(t_n + h, \mathbf{Y}_s),
 \end{aligned}$$

where  $a_i$  is assumed to vanish. This implementation is of the form (2.1) with  $\mathbf{b} = \mathbf{e}_s$  and a matrix  $A$  with zero entries except for the lower off-diagonal entries. We shall call (2.10) the *diagonal* implementation. Unfortunately, when we actually applied the diagonal implementation with  $s \approx (\gamma_p^{-1} h \rho(J_n))^{1/2}$ , it turned out that the numerical solution lost accuracy for larger values of  $s$ . On a computer with 14 digits arithmetic,  $s$  should not be greater than 12. This is caused by the development of *internal* instabilities within a single step. Just as the step values  $\mathbf{y}_n$  are required to be stable by imposing the (external) stability condition  $h < \beta_{\text{real}} / \rho(J_n)$ , we also have to require that the internal values  $\mathbf{Y}_i$  are stable. In the implementation (2.10), the internal perturbations satisfy the recursion  $\Delta \mathbf{Y}_i = a_i h J_n \Delta \mathbf{Y}_{i-1} = R_{i-1}(h J_n) \Delta \mathbf{Y}_1$ , where the so-called *internal stability polynomials*  $R_i(z)$  are of degree  $i$  in  $z$ . This leads to the *internal* stability conditions  $h < \alpha_i / \rho(J_n)$ ,  $i = 1, \dots, s$ , where  $\alpha_i$  denotes the stability boundary associated with  $R_i$ . For large values of  $s$ , these conditions are much more restrictive than the external stability condition  $h < \beta_{\text{real}} / \rho(J_n)$ . As a consequence, the main advantage of the polynomials (2.5), (2.7) and (2.9), viz. that they are available for arbitrarily large values of  $s$ , cannot be exploited.

Fortunately, it is possible to avoid, or at least to suppress the internal instabilities, just by choosing another implementation than (2.10). The first attempt to internal stabilization of RK methods with many stages is due to Gentsch and Schlüter [8] in 1978, who 'rediscovered' the shifted Chebyshev polynomials (2.5) and exploited the fact that these polynomials possess  $s$  real zeroes  $z_i$  on the negative axis. Although their approach was restricted to linear IVPs, it can directly be extended to nonlinear problems to obtain an RK method of the form

$$\begin{aligned}
 \mathbf{Y}_1 &= \mathbf{y}_n, \quad \mathbf{Y}_{i+1} = \mathbf{Y}_i - \frac{1}{z_i} h \mathbf{f}(t_n + c_i h, \mathbf{Y}_i), \quad i = 1, \dots, s-1, \\
 (2.11) \quad \mathbf{y}_{n+1} &= \mathbf{Y}_s - \frac{1}{z_s} h \mathbf{f}(t_n + c_s h, \mathbf{Y}_s).
 \end{aligned}$$

This implementation may be interpreted as an RK method that is factorized in a sequence of Euler steps and will be called the *factorized* implementation. If the zeroes  $z_i$  are ordered such that  $z_i < z_{i+1}$  or  $z_i > z_{i+1}$ , then the performance the factorized implementation is hardly better than that of the diagonal implementation as  $s$  increases. However, Gentsch and Schlüter reported satisfactory results for extremely large values of  $s$  (up to 997) if special orderings of the  $z_i$  are used. A disadvantage in actual applications is that a suitable ordering depends on  $s$ .

When reading the paper of Gentsch and Schlüter, we suddenly realized, that the problem of internal stabilization was already solved a long time ago by numerical analysts working in *elliptic* PDEs! The spatial discretization of elliptic PDEs leads

to the problem of solving linear systems  $Ay = b$ , where  $A$  is known to have a negative spectrum in the negative interval  $(-\rho(A), 0)$  with  $\rho(A)$  large positive. A well-known iterative method for solving such problems is due to Richardson, who proposed in his paper [30] of 1910 the recursion  $y_i = y_{i-1} + \alpha_i(Ay_{i-1} - b)$ , where the parameters  $\alpha_i$  are chosen such that after  $s$  iterations, the polynomial  $P_s$  occurring in the error formula  $y_s - y = P_s(A)(y_0 - y)$  has a small norm in the eigenvalue interval  $(a, b)$  of  $A$ . Various approaches to achieve this have been proposed. Richardson suggested choosing  $P_s$  such that it has uniformly distributed zeros in  $(a, b)$ , Stiefel proposed to minimize an integral measure of  $P_s$  (cf. [36]), but most numerical analysts prefer to minimize the maximum norm of  $P_s$ . The latter approach leads to shifted Chebyshev polynomials that are very similar to (2.5). This process is now known as Richardson's method of *first* degree. However, application of this method for large values of  $s$  suffers the same internal instability as the method (2.11). Just as Gentsch and Schlüter, one has tried to improve the stability by special choices of the ordering of the parameters  $\alpha_i$  (see e.g. the experiments of Young [40] in 1954), but a real break-through was due to Stiefel [36] in 1958. He observed that Chebyshev polynomials satisfy a *stable* three-terms recursion, so that using a three-terms recursion for the iterates  $y_i$ , rather than the two-terms recursion of Richardson, would avoid the instability problem. This two-step iteration method is known as Richardson's method of *second* degree or, in the more recent literature, the *Chebyshev semi-iterative* method. Realizing that the stability polynomials (2.5), (2.7) and (2.9) are also expressions in terms of shifted Chebyshev polynomials, brought us to construct internally stable implementations of the corresponding parabolic RK methods (cf. [15,16]). For the second-order consistent polynomials  $A_s^{(2)}$  and  $B_s^{(2)}$ , it was pointed out by Sommeijer (see [15]), that it is even possible to make the  $Y_i$  not only stable, but also second-order accurate approximations to the exact solution at the intermediate points  $t_n + c_i h$ ,  $i = 1, \dots, s$ .

The internally stable Runge-Kutta method generated by the Bakker polynomials  $B_s^{(2)}$  performs slightly better than the method generated by  $A_s^{(2)}$  (its smaller stability boundary is compensated by its smaller error constants). It is a highly efficient integrator for general heat flow problems, particularly for 2D and 3D problems. We called it the *Runge-Kutta-Chebyshev method*, but it could equally well have been called the *Runge-Kutta-Bakker method*. A detailed study of its convergence is presented in [37] and an extensive performance evaluation can be found in [12]. The Runge-Kutta-Chebyshev method has been implemented by Sommeijer as the code RKC and is available through netlib [34].

Another code that is based on stabilized RK methods is the code DUMKA developed by Lebedev and his coworkers of the Institute for Numerical Mathematics of the Russian Academy of Science. They approximate the optimal stability polynomials by so-called Zolotarev polynomials. Like Gentsch and Schlüter, internal stability is achieved by a special ordering of the stages rather than using recurrence relations. More details can be found in the references [25, 26].

Finally, we compare the total computational work of conventional and parabolic RK methods needed for integrating the unit interval with a given step  $h$ . Assuming that  $s$  is defined by  $s \approx (\gamma_p^{-1} h \rho(J_n))^{1/2}$ , we find for the stabilized RK methods  $W = h^{-1} s O(\Delta^{-d}) = h^{-1} (\gamma_p^{-1} h \rho(J_n))^{1/2} O(\Delta^{-d}) = O(h^{-1/2} \Delta^{-d-1})$ . Comparing this estimate with that derived for conventional RK methods, we see that the computational complexity of the stabilized RK methods differ by a factor of order

$O(h^{1/2}\Delta^{-1})$ . With respect to the Crank-Nicolson, Laasonen and BDF2 methods using conjugate gradient type iteration methods, the stabilized RK methods are at least competitive.

### 2.3. Hyperbolic RK methods

Instead of maximizing the *real* stability boundary of stability polynomials of the form (2.4), we may also maximize the *imaginary* stability boundary, to obtain a *hyperbolic RK method* that should be suitable for integrating hyperbolic problems that have Jacobians with imaginary eigenvalues.

For  $p = 1$ , the optimal polynomials are given by

$$(2.12) \quad I_s^{(1)}(z) = (-i)^s \left[ i T_{s-1}\left(\frac{iz}{s-1}\right) - \left(1 + \frac{z^2}{(s-1)^2}\right) U_{s-2}\left(\frac{iz}{s-1}\right) \right], \quad \beta_{\text{imag}} = s - 1,$$

where  $s \geq 2$  and  $U_s(z) := \sin((s+1) \arccos(z)) / \sin(\arccos(z))$  denotes the second kind Chebyshev polynomial of degree  $s$ . For *odd* values of  $s$ , these polynomials were given in 1972 in [13] (a proof can be found in [14]). At the time, it was not realized that (2.12) is also valid for *even* values of  $s$ , because in [13] the polynomials  $I_s^{(1)}$  were represented in the form

$$(2.13) \quad I_s^{(1)}(z) = T_k\left(1 + \frac{z^2}{2k^2}\right) + \frac{z}{k} \left(1 + \frac{z^2}{4k^2}\right) U_{k-1}\left(1 + \frac{z^2}{2k^2}\right), \quad \beta_{\text{imag}} = s - 1,$$

with  $s = 2k+1$ ,  $k \geq 1$ , which cannot directly be extended to even values of  $s$ . It turns out that the odd-degree polynomials are identical to the optimal polynomials corresponding to  $p = 2$ , i.e.  $I_s^{(2)}(z) = I_s^{(1)}(z)$  for  $s$  odd.

In 1984 Kinnmark and Gray [22] derived the representation (2.12) which is valid for all values of  $s$ . This result was also obtained, independently, by Sonneveld and van Leer [35] in 1985.

Kinnmark and Gray [23] have also derived approximations to the optimal polynomials  $I_s^{(3)}$  for  $s$  odd and to  $I_s^{(4)}$  for  $s$  even. These *Kinnmark-Gray polynomials* are given by

$$(2.14) \quad K_s^{(3)}(z) = \frac{1}{\beta^2 + 1} \left[ 1 + z + i^{s-1} \beta^2 T_{s-1}\left(\frac{iz}{\beta}\right) + \frac{1}{2} i^{s+2} \beta \left\{ (s-2) T_s\left(\frac{iz}{\beta}\right) - s T_{s-2}\left(\frac{iz}{\beta}\right) \right\} \right],$$

$$\beta_{\text{imag}} = \beta := \sqrt{(s-1)^2 - 1}, \quad \text{odd } s \geq 3$$

and

$$(2.15) \quad K_s^{(4)}(z) = \sqrt{\frac{1}{\beta^2 + 1}} \left[ i^{s+1} \beta T_{s-1}\left(\frac{iz}{\beta}\right) + \frac{1}{2} i^s \left\{ (s-2) T_s\left(\frac{iz}{\beta}\right) - s T_{s-2}\left(\frac{iz}{\beta}\right) \right\} \right],$$

$$\beta_{\text{imag}} = \beta := \sqrt{(s-1)^2 - 1}, \quad \text{even } s \geq 4.$$

Earlier, in 1983, Vichnevetsky [38] had already proved that  $\beta_{\text{imag}} \leq s - 1$  for all  $p$  and  $s$ . Hence, this result of Vichnevetsky indicates that the Kinnmark-Gray polynomials are extremely close approximations to the optimal ones. However, it also indicates that, unlike the situation for parabolic problems, *hyperbolic* RK methods are hardly more effective than conventional RK methods with nonempty imaginary stability intervals.

### 3. SPLITTING METHODS

Just as RK methods, splitting methods compute in each step two or more intermediate stages. However, unlike RK methods, these stages are not expressed in the full righthand side of the PDE, but in *fractions* of the righthand side. Almost all splitting methods proposed in the literature can be represented in RK format. This approach was followed in [19] to develop a unified treatment of splitting methods and allows a straightforward derivation of the order conditions and stability functions.

Let the righthand side function in (1.1) is reduced to autonomous form and split according to

$$(3.1) \quad \mathbf{f}(\mathbf{y}(t)) = \sum_{i=1}^{\sigma} \mathbf{f}_i(\mathbf{y}(t)),$$

and consider the RK type method

$$(3.2) \quad \mathbf{Y} = \mathbf{e} \otimes \mathbf{y}_n + h \sum_{k=1}^{\sigma} (\mathbf{A}^{(k)} \otimes \mathbf{I}) \mathbf{F}_k(\mathbf{Y}),$$

$$\mathbf{y}_{n+1} = (\mathbf{e}_s^T \otimes \mathbf{I}) \mathbf{Y},$$

where  $\mathbf{F}_k(\mathbf{Y})$  contains the derivative values ( $\mathbf{f}_k(\mathbf{Y}_j)$ ). If  $\sigma = 1$ , then (3.2) reduces to the RK method (2.1) with  $\mathbf{b}^T = \mathbf{e}_s^T \mathbf{A}$ . The method  $\{(3.1), (3.2)\}$  will be called a  *$\sigma$ -terms RKS method with  $s$  fractional stages*. RKS methods consist of two components, the righthand side splitting (3.1) and the splitting scheme (3.2).

Restricting our discussion to first-order and second-order methods and using the compact notation in terms of the matrices  $\mathbf{A}^{(k)}$ , we have first-order accuracy if

$$(3.3) \quad \mathbf{e}_s^T \mathbf{A}^{(j)} \mathbf{e} = 1, \quad j = 1, \dots, \sigma,$$

and second-order accuracy if, in addition,

$$(3.4) \quad \mathbf{e}_s^T \mathbf{A}^{(j)} \mathbf{A}^{(k)} \mathbf{e} = \frac{1}{2}, \quad j, k = 1, \dots, \sigma.$$

In actual computations, the time-dependent parts originating from time-dependent boundary conditions, cannot be dealt with by simply writing (1.1) in autonomous form and need a more careful treatment. In this overview, we shall not elaborate on this aspect of splitting methods (see e.g. [5]).

The linear stability of RKS methods can be analysed by means of the test equation

$$(3.5) \quad \mathbf{y}'(t) = \sum_{k=1}^{\sigma} \mathbf{J}_k \mathbf{y}(t),$$

where  $\mathbf{J}_k$  is the Jacobian matrix  $\partial \mathbf{f}_k(\mathbf{y}_n) / \partial \mathbf{y}$ . It will be assumed that  $\mathbf{J}_k$  has its eigenvalues in the left halfplane. Defining  $\mathbf{Z}_k = h\mathbf{J}_k$ ,  $k = 1, \dots, \sigma$ , we deduce

$$\mathbf{Y} = \mathbf{e} \otimes \mathbf{y}_n + \sum_{k=1}^{\sigma} (\mathbf{A}^{(k)} \otimes \mathbf{Z}_k) \mathbf{Y} = (\mathbf{I} - \mathbf{S})^{-1} (\mathbf{e} \otimes \mathbf{y}_n),$$

$$\mathbf{S} := \sum_{k=1}^{\sigma} (\mathbf{A}^{(k)} \otimes \mathbf{Z}_k).$$

Hence,

$$\mathbf{y}_{n+1} = (\mathbf{e}_s^T \otimes \mathbf{I}) \mathbf{Y} = (\mathbf{e}_s^T \otimes \mathbf{I}) (\mathbf{I} - \mathbf{S})^{-1} (\mathbf{e} \otimes \mathbf{y}_n)$$

$$= (\mathbf{e}_s^T \otimes \mathbf{I}) (\mathbf{I} - \mathbf{S})^{-1} (\mathbf{e} \otimes \mathbf{I}) \mathbf{y}_n.$$

Thus, the *stability function* is given by

$$(3.6) \quad \mathbf{R} = (\mathbf{e}_s^T \otimes \mathbf{I}) \left( (\mathbf{I} \otimes \mathbf{I}) - \sum_{k=1}^{\sigma} (\mathbf{A}^{(k)} \otimes \mathbf{Z}_k) \right)^{-1} (\mathbf{e} \otimes \mathbf{I}).$$

### 3.2. Splitting methods as RKS methods

This survey paper is concluded with an example of a family of splitting methods that can be represented as an RKS method. For a more detailed analysis of RKS methods we refer to [18].

Consider the two-terms, three-stage splitting scheme defined by

$$(3.7) \quad \mathbf{A}^{(1)} = \begin{pmatrix} 0 & 0 & 0 \\ 1/2 & 0 & 0 \\ 1/2 & 0 & 1/2 \end{pmatrix}, \quad \mathbf{A}^{(2)} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

This scheme is second-order accurate whatever we choose for  $\mathbf{f}_1$  and  $\mathbf{f}_2$ . Presumably, the first splitting method proposed in the literature generated by the splitting scheme (3.7) is the Peaceman-Rachford method [29] of 1955. If (3.7) is applied to a space-discretized, two-dimensional PDE in which the righthand side  $\mathbf{f}$  can be split into an

$x$ -dependent part  $f_1$  and a  $y$ -dependent part  $f_2$ , then the so-called ADI (Alternating Direction Implicit) method of Peaceman and Rachford is obtained. Other well-known splitting methods generated by (3.7) are the Hopscotch methods proposed by Gourlay in 1970. These methods are obtained by dividing the grid points on which the PDE is discretized in two groups  $G_1$  and  $G_2$ , and by defining  $f_1$  and  $f_2$  such that they vanish on  $G_1$  and  $G_2$ , respectively. On rectangular grids, often used examples are the Line Hopscotch and the Odd-Even Hopscotch methods which arise if  $G_1$  and  $G_2$  contain grid points lying on alternating lines and diagonals, respectively.

#### ACKNOWLEDGEMENT

The author is grateful to Dr. B.P. Sommeijer for his interest in this survey paper and for his many comments to improve the presentation of the available material.

#### REFERENCES

- [1] Bakker, M. (1971): Analytic aspects of a minimax problem (Dutch), Report TN 62, Mathematisch Centrum, Amsterdam.
- [2] Burrage, K. (1985): Order and stability properties of explicit multivalued methods, *Appl. Numer. Math.* 1, 363-379.
- [3] Curtiss, C.F. & Hirschfelder, J.O. (1952): Integration of stiff equations, *Proc. Nat. Acad. Sci. U.S.* 38, 235-43.
- [4] Crank, J. & Nicolson, P. (1947): A practical method for numerical integration of solutions of partial differential equations of heat-conduction type, *Proc. Cambridge Philos. Soc.* 43, 50-67.
- [5] Fairweather, G. & Mitchell, A.R. (1967): A new computational procedure for A.D.I. methods, *SIAM J. Numer. Anal.* 4, 163-170.
- [6] Franklin, J.N. (1959): Numerical stability in digital and analogue computation for diffusion problems, *J. Math. Phys.* 37, 305-315.
- [7] Gear, C.W. (1971): Numerical initial value problems in ordinary differential equations, Prentice Hall, Englewood Cliffs N.J..
- [8] Gentzsch, W. & Schlüter, A. (1978): On one-step methods with cyclic stepsize changes for solving parabolic differential equations (German), *Z. Angew. Math. Mech.* 58, T415-T416.
- [9] Guillo, A. & Lago, B. (1960): Stability regions of one-step and multistep formulas for differential equations; Investigation of formulas with large stability boundaries (French), 1<sup>e</sup> Congrès de l'Association Française de Calcul, AFCAL, Grenoble, Sept. 1960, 43-56.
- [10] Hairer, E., Nørsett, S.P. & Wanner, G. (1987/91): Solving ordinary differential equations I. Nonstiff problems, Springer-Verlag, Berlin.
- [11] Henrici, P. (1962): Discrete variable methods in ordinary differential equations, Wiley, New York.
- [12] Hofmann, S. (1992): First and second-order Runge-Kutta-Chebyshev methods for the numerical integration of parabolic differential equations and stiff ordinary differential equations (in German), Master's thesis, University of Wuppertal, Germany.
- [13] Houwen, P.J. van der (1972): Explicit Runge-Kutta methods with increased stability boundaries, *Numer. Math.* 20, 149-164.



- [14] Houwen, P.J. van der (1977): Construction of integration formulas for initial-value problems, North-Holland, Amsterdam.
- [15] Houwen, P.J. van der (1981): On the time integration of parabolic differential equations, in: G.A. Watson (ed.): Numerical analysis, Lecture Notes in Mathematics 912, Springer, New-York, 157-168.
- [16] Houwen, P.J. van der & Sommeijer, B.P. (1980): On the internal stability of explicit m-stage Runge-Kutta methods for large values of m, *Z. Angew. Math. Mech.* 60, 479-485.
- [17] Houwen, P.J. van der & Sommeijer, B.P. (1982): A special class of multistep Runge-Kutta methods with extended real stability interval, *IMA J. Numer. Anal.* 2, 183-209.
- [18] Houwen, P.J. van der & Sommeijer, B.P. (1994): Runge-Kutta splitting methods, in preparation.
- [19] Houwen, P.J. van der & Verwer, J.G. (1979): One-step splitting methods for semi-discrete parabolic equations, *Computing* 22, 291-309.
- [20] Jeltsch, R. & Nevanlinna, O. (1981): Stability and accuracy of time discretizations for initial value problems, Report HTKK-MAT-A187, Helsinki University of Technology.
- [21] Jeltsch, R. & Nevanlinna, O. (1981): Stability of explicit time discretizations for solving initial value problems, *Numer. Math.* 37, 61-91.
- [22] Kinnmark, I.P.E. & Gray, W.G. (1984): One-step integration methods with maximum stability regions, *Math. Comput. Simulation* 16, 87-92.
- [23] Kinnmark, I.P.E. & Gray, W.G. (1984): One-step integration methods of third-fourth order accuracy with large hyperbolic stability limits, *Math. Comput. Simulation* 16, 181-184.
- [24] Laasonen, P. (1949): On a method for solving the heat flow equation (German), *Acta Math.* 81, 309-323.
- [25] Lebedev, V.L. (1987): Explicit difference schemes with time-variable steps for solving stiff systems of equations (in Russian), Preprint No. 177, Dept. of Numerical Mathematics, USSR Acad. Sc., Moscow.
- [26] Lebedev, V.L. (1994): How to solve stiff systems of equations by explicit difference schemes, In: Numerical Methods and Applications (ed. G.I. Marchuk), CRC Press, Boca Raton, Ann Arbor, London-Tokyo, 45-80.
- [27] Lomax, H. (1968): On the construction of highly stable, explicit numerical methods for integrating coupled ODEs with parasitic eigenvalues, NASA Technical Note NASAIN D/4547.
- [28] Metzger, C.L. (1967): Runge-Kutta methods whose number of stages exceeds their order (French), These (Troisieme cycle), Université de Grenoble.
- [29] Peaceman, D.W. & Rachford, H.H. Jnr. (1955): The numerical solution of parabolic and elliptic differential equations, *J. Soc. Indust. App. Math.* 3, 28-41.
- [30] Richardson, L.F. (1910): The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam, *Philos. Trans. Roy. Soc. London, Ser. A* 210, 307-357 and *Proc. Roy. Soc. London, Ser. A* 83, 335-336.
- [31] Riha, W. (1972): Optimal stability polynomials, *Computing* 9, 37-43.

- [32] Rothe, E. (1930): Two-dimensional parabolic boundary value problems as limiting case of one-dimensional boundary value problems (German), *Math. Annalen*, 102, 650-670.
- [33] Runge, C. (1895): On the numerical solution of differential equations (German), *Math. Ann.* 46, 167-178.
- [34] Sommeijer, B.P. (1991): RKC, a nearly-stiff ODE solver, available through netlib (mail: netlib@ornl.gov, send rkc.f from ode).
- [35] Sonneveld, P & Leer, B. van (1985): A minimax problem along the imaginary axis, *Nieuw Archief voor Wiskunde* (4) 3, 19-22.
- [36] Stiefel, E.L. (1958): Kernel polynomials in linear algebra and their numerical applications, *Nat. Bur. Stand. Appl. Math. Series* 49, 1-22.
- [37] Verwer, J.G., Hundsdorfer, W.H. & Sommeijer, B.P. (1990): Convergence properties of the Runge-Kutta-Chebyshev method, *Numer. Math.* 57, 157-178.
- [38] Vichnevetsky, R. (1983): New stability theorems concerning one-step numerical methods for ordinary differential equations, *Math. Comput. Simulation* 25, 199-205.
- [39] Yanenko, N.N. (1971): *The method of fractional steps*, Springer-Verlag, Berlin.
- [40] Young, D.M. (1954): On Richardson's method for solving linear systems with positive definite matrices, *J. Math. Phys.* 32, 243-255.
- [41] Yuan' Chzao-Din (1958): Some difference schemes for the solution of the first boundary value problem for linear differential equations with partial derivatives (Russian), Thesis, Moskow State University.

# The Essence of the Law of Large Numbers

Michael Keane

CWI

P.O. Box 94079

1090 GB Amsterdam

Electronic mail: keane@cwi.nl

The law of large numbers, not really a law but a mathematical theorem, is at the same time a justification for application of statistics and an essential tool for the mathematical theory of probability. As such, it must be taught to many students. The traditional method for this, using independent and identically distributed random variables, was developed by Kolmogorov in the 1930's, and explains well what happens, and much more, at this level of generality. However, it has recently come to light that the reason for the validity of this theorem in its general setting, that of stationarity, is much simpler than was first thought. In this short article, I shall try to explain to the general audience towards whom this collection is directed, the essence of the law of large numbers. A complete treatment should certainly include many references and interesting historical comments, and I apologize for their absence here.

Let me start with the *basic law of large numbers* by considering, very simply, an infinite sequence

$$x_0, x_1, x_2, \dots$$

each of whose elements is either 0 or 1. Perhaps it will help (or hinder!) to think of  $x_n$  as the result of the  $n^{\text{th}}$  trial of an uncertain experiment, with  $x_n = 1$  designating success and  $x_n = 0$  failure. Let

$$a_n = \frac{x_0 + x_1 + \dots + x_{n-1}}{n} \quad (n \geq 1)$$

denote then the average numbers of successes up to time  $n$ . It is very easy to see mathematically that for some sequences  $x$ ,

$$\lim_{n \rightarrow \infty} a_n$$

exists, while for other sequences  $x$ , this is not the case. One can only affirm with certainty that

$$\lim_{n \rightarrow \infty} (a_{n+1} - a_n) = 0,$$

but nothing impedes the averages  $a_n$  from oscillating more and more slowly as  $n$  grows. Thus it seems that further discussion is useless, and that uncertainty here must be accepted.

Phenomenologically, however, we are faced with the fact that in certain situations, such limits seem to exist, and the society makes seemingly understandable statements concerning the percentage of smokers dying of cancer, the probability of rain tomorrow, or an industrial average yield. We are confronted with the question as to whether nature produces sequences whose averages do converge, and why. Of course, this is not a mathematical question, and in order to say something mathematically sensible, one must adopt a model.

The currently accepted model, and it is difficult to see how it could be replaced by something else, is that for a given situation in which such sequences  $x$  appear, in principle all sequences are possible, but there is also a mass distribution with total mass 1 over the set of sequences, which assigns to each "event" which might occur a *probability*, this being the total mass of those sequences for which the event occurs. If an event, for instance the existence of  $\lim_{n \rightarrow \infty} a_n$ , has probability 1, then one says that the event will occur *almost surely*.

The determination of such a mass distribution in different practical situations is one of the most important tasks for probabilists, and requires a good mixture of mathematics, other sciences, and good old common sense. First principles are of utmost importance, as determining such an object by experimentation resembles very much a cat chasing its own tail! One of the basic properties of such a mass distribution, already alluded to briefly above, is that of *stationarity*. We say that the probability measure (= mass distribution) is *stationary* if the events have time-homogeneous probabilities. That is, shifting any event forwards or backwards in time does not change its probability.

Perhaps a brief remark on mass distributions is in order. There is a branch of mathematics, measure theory, which deals extensively with the specification and manipulation of such objects. However, one can understand well most arguments and principles by using the intuitive notion, which is my intention here.

Now we can state the

#### BASIC LAW OF LARGE NUMBERS:

If  $x = (x_0, x_1, \dots)$  is a stationary sequence of zeroes and ones, then  $\lim_{n \rightarrow \infty} a_n$  exists almost surely.

Just to be sure that you are (mathematically) still with me: A unit mass distribution on sequences of zeroes and ones is given; it is stationary. Then the set of all sequences  $x$  for which  $\lim_{n \rightarrow \infty} a_n$  exists has total mass 1. The set of sequences for which this limit does not exist has mass 0. Remember, this is a theorem, and I want to explain the proof.

To understand the proof will require the level of first-year university analysis,

given the intuitive acceptance of the mass distribution notion. We begin by defining

$$\bar{a} := \limsup_{n \rightarrow \infty} a_n;$$

this always exists, and  $0 \leq \bar{a} \leq 1$ . It is also clear that if we had started observing  $x$  at a later time point, the value  $\bar{a}$  would be the same:

$$\bar{a} = \limsup_{n \rightarrow \infty} \frac{x_k + x_{k+1} + \dots + x_{k+n-1}}{n}$$

for any  $k \geq 0$  and any sequence  $x$ .

Next, we need a way to measure how close we are to the lim sup,  $\bar{a}$ . Thus, let  $\epsilon > 0$  be a fixed positive number, and for each  $k \geq 0$ , define

$$N_k := \min \left\{ n \geq 1 : \frac{x_k + x_{k+1} + \dots + x_{k+n-1}}{n} \geq \bar{a} - \epsilon \right\}.$$

By the definition of lim sup, the set on the right is non-empty and hence  $N_k$  is finite for each  $k$ . The crucial point we need to address concerns the size of the numbers  $N_k$ ; to make our idea clear, let us examine the simplest case first.

CASE 1. Suppose that for each  $\epsilon > 0$  there exists a (large) positive integer  $M$  such that for each  $k$ ,  $N_k \leq M$  almost surely. (That is, the set of sequences  $x$  for which  $N_k \leq M$  has total mass 1.)

REMARK: Note that by our assumption of stationarity the events  $N_k \leq M$  for different  $k$  all have the same probability.

If now  $x$  is such a sequence that for each  $k$ ,  $N_k \leq M$ , we claim that  $\lim_{n \rightarrow \infty} a_n$  exists. The idea is that, as  $n$  gets larger,  $a_n$  can only change more and more slowly, and that then wandering is impossible because the lim sup is reached again and again within  $M$  steps. Formally, one proceeds as follows. Fix  $\epsilon > 0$  and choose any  $n > M/\epsilon$ . Then starting at the beginning of  $x$ , break  $x$  up into pieces of lengths at most  $M$  such that the average of  $x$  over each piece is at least  $\bar{a} - \epsilon$ . Stop at the piece containing the coordinate  $n$ . Then it is clear that

$$x_0 + x_1 + \dots + x_{n-1} \geq (n - M) (\bar{a} - \epsilon),$$

so that

$$a_n = \frac{x_0 + x_1 + \dots + x_{n-1}}{n} \geq (1 - \epsilon) (\bar{a} - \epsilon) \geq \bar{a} - 2\epsilon$$

for each  $n > M/\epsilon$ ; it follows that  $\lim_{n \rightarrow \infty} a_n = \bar{a}$  exists.

REMARK: Note that only the last piece is of importance; it must not become too long.

Actually, the same type of argument works in the general case, when combined with an idea coming originally from non-standard analysis.

CASE 2: General case. By the remark after Case 1, it remains true that the events  $N_k \leq M$  all have the same probability, for any  $k$  and fixed  $M$ . Since  $N_k$  is finite for each  $x$ , we may not be able to find an  $M$ , for  $\epsilon > 0$  given, such that these events have probability 1, but we certainly can choose  $M$  so large that for any  $k$ , the probability of  $N_k \leq M$  is less than  $\epsilon$ .

Fix now such an integer  $M$ , given  $\epsilon > 0$ . Next, we want to make the same inequality work for us, but we are impeded whenever  $N_k > M$ . So let us change  $x$  at those places to insure quick arrival at the lim sup. Namely, define

$$x_k^* := \begin{cases} x_k & \text{if } N_k \leq M \\ 1 & \text{if } N_k > M. \end{cases} \quad (k \geq 0)$$

Then clearly  $x_k^* \geq x_k$  for each  $k$ , so that if we set

$$N_k^* := \min \left\{ n \geq 1 : \frac{x_k^* + \dots + x_{k+n-1}^*}{n} \geq \bar{a} - \epsilon \right\}$$

(same  $\bar{a}$ ), then  $N_k^* \leq N_k$ , and moreover if  $k$  is such that

$$N_k > M,$$

then we have

$$N_k^* = 1,$$

since setting  $x_k^* = 1$  insures immediate arrival above  $\bar{a} - \epsilon < 1$ .

Now we are almost ready. As above, breaking  $x^*$  up into pieces yields for  $n > M/\epsilon$ .

$$x_0^* + x_1^* + \dots + x_{n-1}^* \geq (n - M) (\bar{a} - \epsilon),$$

but now we cannot conclude anything about the sequence  $x$  because we have replaced it by  $x^*$ .

Instead, we now need to use our mass distribution to calculate the average value of each side of the inequality over all sequences  $x$ , called by probability theory the *expectation* and denoted by  $\mathbb{E}(\cdot)$ . Let

$$\mathbb{E}(x_0) =: p$$

and

$$\mathbb{E}(x_0^*) =: p^*;$$

by stationarity,  $\mathbb{E}(x_p^*) = p^*$  for all  $k$ , and by the choice of  $M$ , we have

$$p^* \leq p + \epsilon.$$

Of course,  $p$  is just the probability that  $x_k = 1$ , and  $p^*$  the probability that  $x_k^* = 1$ , for any  $k$ . Now, taking expectations of each side of the inequality results in

$$n(p + \epsilon) \geq np^* \geq (n - M)(\mathbb{E}(\bar{a}) - \epsilon) \quad (n \geq M/\epsilon).$$

Now divide by  $n$ , send  $n$  to infinity and then  $\epsilon$  to zero, giving

$$\mathbb{E}(\bar{a}) = \mathbb{E}(\limsup_{n \rightarrow \infty} \frac{x_0 + \dots + x_{n-1}}{n}) \leq p.$$

Finally, apply the entire argument above to the “mirrored” 0 – 1–sequence  $y_k = 1 - x_k$ ; an easy calculation (exercise!) shows that

$$\mathbb{E}(\liminf_{n \rightarrow \infty} \frac{x_0 + \dots + x_{n-1}}{n}) \geq p.$$

But for any sequence  $x$ , certainly

$$\liminf_{n \rightarrow \infty} \frac{x_0 + \dots + x_{n-1}}{n} \leq \limsup_{n \rightarrow \infty} \frac{x_0 + \dots + x_{n-1}}{n};$$

it is an elementary fact of expectations or averaging that the three inequalities then must be equalities, the last one almost surely. Hence  $\limsup = \liminf$  for a set of sequences of total mass one, i.e. the limit exists almost everywhere. This concludes the proof of the basic law of large numbers.

In concluding, we state without proof that this method can be widely extended with minor, straight-forward modifications to the most general laws of large numbers based on stationarity. The above proof should, however, in my opinion be included in basic probability courses, since it so clearly shows the nature of the interplay of stationarity assumptions and the existence of statistical limits.





# Het Europese ESPRIT Programma: Een Persoonlijk Perspectief

P. Klint

*Opgedragen aan Cor Baayen voor zijn bijdrage aan de Europese samenwerking op het gebied van Wiskunde en Informatica.*

## 1 INLEIDING

*Historie.* Het *European Strategic Programme for Research and Development in Information Technologies*, kortweg ESPRIT, is in 1983 gestart met als oorspronkelijk doel om de Europese IT industrie van de vereiste basistechnologie te voorzien die nodig zou zijn om de jaren negentig te overleven, om Europese samenwerking te bevorderen, en om bij te dragen aan de ontwikkeling van internationaal geaccepteerde standaards.

De doelstellingen van het ESPRIT programma zijn geleidelijk verschoven. In de eerste fase (1983-1986) stond het creëren van samenwerking op Europese schaal tussen onderzoekscentra en bedrijven centraal. In de tweede fase (1986-1990), lag het accent meer op precompetitief onderzoek, d.w.z. gemeenschappelijke onderzoeksinspanningen die pas ná het ESPRIT project door de deelnemers in producten omgezet moesten worden. In de huidige derde en laatste fase ligt de nadruk vooral op het ontwikkelen van producten die aan het einde van een project marktrijp zijn. Voor onderzoek is in deze fase nauwelijks plaats meer.

Naast dit zogenaamde *hoofdprogramma* van ESPRIT, bestaat er sinds enkele jaren een *Basic Research Action* (BRA) waarin op kleine schaal fundamenteel onderzoek gesubsidieerd wordt. Ik beperk me in dit artikel verder tot het hoofdprogramma van ESPRIT.

*Eigen ESPRIT achtergrond.* Ikzelf ben direct betrokken geweest bij drie ESPRIT projecten. Indirect heb ik de voortgang van diverse andere ESPRIT projecten kunnen meebeleven doordat mijn collega's daaraan deelnamen. Daarnaast ben ik incidenteel als beoordelaar van projecten opgetreden. Al met al voldoende voor een persoonlijk perspectief, maar duidelijk onvoldoende voor een objectieve beoordeling van het ESPRIT programma als geheel.

## 2 VAN VOORSTEL TOT PROJECT

*Call for Proposals.* In het ESPRIT programma is een aantal keren een "call for proposals" uitgeschreven waarin Europese bedrijven, universiteiten en onderzoeksinstituten uitgenodigd werden om projectvoorstellen in te dienen op van te voren, ruim omschreven, gebieden.

Hierop werd door de doelgroep gereageerd door oude contacten te hernieuwen of door nieuwe contacten te leggen om zo te komen tot een consortium van aanvragers die tezamen een voorstel bij de EG indienen. De samenstelling van een consortium wordt deels door inhoudelijke en deels door opportunistische overwegingen bepaald. Het ligt natuurlijk voor de hand om inhoudelijk geïnteresseerde partijen bij elkaar te brengen. Daarnaast spelen factoren als deelname van een van de grote twaalf IT industrieën, het land van herkomst (bij voorkeur Zuid-Europa of Ierland), of aanzien van een partij binnen ESPRIT een grote rol. Bovendien stelt de EG zelf voorwaarden aan de samenstelling, o.a. qua verdeling over Europese landen, verhouding bedrijven/universiteiten en dergelijke.

*De motivatie.* Waarom doen partners mee? Deze vraag is in zijn algemeenheid natuurlijk moeilijk te beantwoorden, maar er zijn toch wel een aantal veelvoorkomende motieven te noemen. Allereerst, om een *gemeenschappelijk doel* te realiseren. Deze situatie doet zich, bij voorbeeld, voor als partijen met vergelijkbare onderzoeksactiviteiten bezig zijn en besluiten deze binnen een ESPRIT project gezamenlijk, en waarschijnlijk intensiever, te bundelen. Ten tweede, is het *importeren van technologie* een overweging om samenwerking te zoeken met technologisch voorliggende partners. Ten derde, en het spiegelbeeld van het vorige punt, kan het *exporteren van technologie* een drijfveer zijn voor een technologisch geavanceerde partij om te proberen de zelf ontwikkelde technologie te propageren via partners die deze kunnen toepassen in hun producten. Tenslotte kunnen de *verhuur van menskracht* ("body shopping") en *publiciteit* drijfveren zijn om aan een ESPRIT voorstel deel te nemen. In de eerste jaren van ESPRIT werd er overigens in academische kring nogal neergekeken op deelname aan ESPRIT projecten, maar met het schaarser worden van de (universitaire) onderzoeksmiddelen hoort men tegenwoordig nauwelijks meer iets van deze skepsis.

*Het voorstel.* Een typisch voorstel is ruwweg honderdvijftig pagina's dik en bestaat uit drie delen. Deel I ( $\pm 30$  pagina's) bevat administratieve en financiële informatie. Deel II (meestal de "Technical Annex" genoemd,  $\pm 90$  pagina's) beschrijft de technisch/inhoudelijke doelstellingen van het project en geeft een zeer gedetailleerd plan (inclusief staaf- en PERT-diagrammen) hoe deze doelstellingen bereikt zullen worden. Essentieel onderdeel is een lijst van tussenresultaten ("deliverables") en de tijdstippen waarop deze geproduceerd zullen worden. Deel III ( $\pm 30$  pagina's) bevat informatie over de deelnemende partijen, toe te passen management- en marketingtechnieken, en de CV's van de staf die het project zal gaan uitvoeren. De omvang van projecten varieert van enkele tientallen mensjaren tot vele honderden.

*De selectie.* De drie onderdelen van een voorstel worden apart beoordeeld. Delen I en III door de EG zelf, deel II door externe deskundigen die per aan-

vraagronde, ad hoc, door de EG ingehuurd zijn. Zij moeten in korte tijd (ongeveer een week) vele tientallen voorstellen beoordelen. Nadat aanvragen in deze eerste ronde op administratieve en technisch/inhoudelijke kwaliteiten beoordeeld zijn, volgt een meer politieke beoordelingsronde waarin factoren zoals het aantal projecten per thema, en de verdeling van subsidies over de verschillende landen, een rol spelen. In deze ronde kunnen ook al onderhandelingen met de aanvragers beginnen over het toevoegen van partners, reduceren van budgetten, of het combineren van verschillende aanvragen over vergelijkbare onderwerpen.

*Het goedgekeurde project.* Als een project eenmaal goedgekeurd is wordt het voor 50% door de EG gefinancierd en begint het, volgens plan, zijn werk. Het is standaard dat elke zes tot twaalf maanden de voortgang van het project bekeken wordt op een "review" bijeenkomst. In een sessie van één tot twee dagen worden de geproduceerde deliverables mondeling toegelicht en besproken met een aantal externe deskundigen ("reviewers") en de EG official die voor het project verantwoordelijk is.

### 3 POSITIEVE ASPECTEN VAN ESPRIT

Hoewel het ESPRIT programma nog niet geleid heeft tot een aantoonbare versterking van de marktpositie van de Europese IT industrie, zijn er toch een aantal, minder tastbare, positieve effecten te noemen.

*De samenwerking.* Er is duidelijk een Europese infrastructuur en samenwerkingscultuur voor IT onderzoek ontstaan. Waar eerst cultuurverschillen en vooroordelen bepalend waren (de "grondige" Duitsers versus de "nonchalante" Fransen) is langzaam maar zeker een wederzijdse appreciatie op basis van prestaties gegroeid. Het kan niet genoeg benadrukt worden hoe belangrijk het is dat vooral jonge medewerkers in dit soort projecten leren inzien dat niet de cultuurgebonden *werkwijze en werkhouding* van buitenlandse partners van belang is, maar dat men vooral op elkaars *feitelijke prestaties* moet letten. Ook de samenwerking tussen bedrijven en universiteiten is door ESPRIT duidelijk bevorderd.

Andere positieve effecten op de deelnemers zijn een toegenomen inzicht en bewustzijn van technologische ontwikkelingen die op Europese schaal plaatsvinden en een verbreding van de horizon tot buiten grenzen van de eigen discipline.

Tenslotte mag niet onvermeld blijven dat voor de meeste onderzoekers ESPRIT projecten eenvoudigweg de grootste samenwerkingsverbanden zijn waaraan zij ooit deelgenomen hebben en de ervaring die daarbij opgedaan is (zowel inhoudelijk als qua management, planning, communicatie, en interne politiek van een project) lijkt me van grote waarde voor de toekomst.

*Het plannen van onderzoek.* Het was voor veel onderzoekers een nieuwe ervaring om gedetailleerde meerjarenplannen te maken voor onderzoeksprojecten

en om regelmatig, uitvoerig, over de voortgang daarvan te moeten rapporteren. Hoewel er vele negatieve kanten aan deze gedetailleerde plannen zitten, zoals ik hieronder zal toelichten, denk ik toch dat onderzoekers geleerd hebben op welke punten planning voor hun onderzoek nuttig is, dat het regelmatig produceren van resultaten essentieel is, en dat zij zich meer bewust geworden zijn van de noodzaak (en de moeilijkheid) om de relevantie en de voortgang van het eigen onderzoek te verdedigen.

#### 4 NEGATIEVE ASPECTEN VAN ESPRIT

*De selectieprocedure.* De call for proposals en de daarop volgende selectieprocedure hebben een aantal negatieve kanten. Allereerst is het opstellen van een voorstel zeer arbeidsintensief. Een investering in de orde van grootte van één meusjaar lijkt me normaal. Ik denk dat vooral de zeer gedetailleerde planning (en al zijn financiële en organisatorische consequenties) daaraan debet is. Ten tweede, bevordert de hierboven beschreven selectieprocedure projectaanvragen met steeds ambitieuzere doelstellingen. Als een aanvraag geen nieuw-ogend idee (“vonk”) bevat, is de kans op afwijzing groot. Hierdoor maken solide, meer op graduele verbetering dan op revolutionaire vindingen gerichte aanvragen nauwelijks kans. De vlucht naar voren in steeds spectaculairder aanvragen is zo onvermijdelijk maar mijns inziens voor de feitelijke technische vooruitgang ongewenst.

*Het plan.* Het is een bekend verschijnsel dat plannenmakers hun plannen met de realiteit verwarren. Een goed plan kan de realiteit natuurlijk in sterke mate beïnvloeden, maar vaak blijkt de realiteit te weerbarstig en zullen plannen bijgesteld moeten worden. Iedereen weet ook dat het gedetailleerd voorspellen van complexe, technische, ontwikkelingen op een termijn van drie à vier jaar onmogelijk is. Toch werken ESPRIT projecten met zeer gedetailleerde plannen met een dergelijke tijdschaal. Vaak gebruikt men plannen die er eerder op gericht lijken om de ambitieuze doelstellingen van een project te adstrueren, dan om de doelstellingen van het project te realiseren. Anders gezegd, de plannen bevatten vaak wenselijkheden in plaats van realistische, effectieve, stappen om een gegeven doel te bereiken.

De situatie is natuurlijk paradoxaal. Gezien vanuit de EG dient het verlenen van subsidies zo zorgvuldig mogelijk te gebeuren. Omdat de projecten inhoudelijk vaak zeer moeilijk door EG officials te beoordelen zijn (ook na het inwinnen van advies van externe deskundigen) is het vastleggen en toetsen van een zeer gedetailleerd plan de enige mogelijkheid tot controle. Gezien vanuit de aanvragers wordt, om opportunistische redenen, de fictie instand gehouden dat een dergelijke planning mogelijk is. Tot overmaat van ramp is het bijstellen van deze gedetailleerde plannen een kostbare zaak.

*De samenwerking.* Samenwerken valt niet mee, zeker niet als de partijen waarmee moet worden samengewerkt een andere culturele, technische of vakma-

tige achtergrond hebben, en daarbij vaak ook nog verschillende doelstellingen nastreven.

Ook juridisch schept samenwerking problemen. Immers, *wie mag wat doen met de in het project behaalde resultaten?* Hoe zit het met het eigendom van eigen producten die tijdens een ESPRIT project (misschien wel door anderen) verbeterd zijn? Wat te denken van verbeteringen en eigen ontwikkelingen die na afloop van het ESPRIT project op de gemeenschappelijke resultaten zijn aangebracht? Wat te doen als resultaten voor een deel binnen verschillende projecten behaald zijn? De amateur waagt zich natuurlijk al helemaal niet aan deze vragen maar ook professionele juristen komen er lang niet altijd uit.

*De resultaten.* De resultaten van ESPRIT projecten zijn vaak indrukwekkend en divers: producten, prototypes, wetenschappelijke publicaties, boeken, en conferenties. Een van de minder positieve resultaten is de verdere groei van het *grijze publicatiecircuit*. Hiermee bedoel ik de altijd dikke en soms interessante “deliverables” die geproduceerd worden. Deze verdwijnen steevast in de Brusselse archieven, er wordt soms door de auteurs zelf nog wel eens naar gerefereerd, maar ze dringen niet door tot het normale wetenschappelijke publicatiecircuit en zijn daardoor effectief onvindbaar. Dit “schrijven voor ESPRIT” is verspilling van energie.

## 5 LESSEN

*De samenwerking.* Essentieel voor een goede samenwerking zijn:

- Gemeenschappelijk begrip van de doelstellingen van het project.
- Erkenning van de (mogelijk conflicterende) doelstellingen en belangen van de partners.
- Gemeenschappelijke (maar ook complementaire) kennis en vocabulair om de doelstellingen te benaderen.
- Uitwisseling van door de partners gebruikte methoden en technieken, en meer in het algemeen, kennistransfer tussen de partners.
- Selectie van methoden en technieken die in het project gebruikt zullen worden.
- Taakverdeling.

Hoewel deze punten voor zichzelf spreken, is voor elk ervan een voorbeeld te geven van een ESPRIT project dat op dat specifieke punt gefaald heeft.

*Een theorie.* Op grond van het voorafgaande, kom ik tot een theorie over ESPRIT projecten. Er zijn allerlei aspecten te onderscheiden die men in elk project in meerdere of mindere mate aantreft, b.v.:

- Was de doelstelling al voor de aanvang van het project gegeven, of is deze pas bij de opzet van het project geformuleerd?
- Is de doelstelling precies geformuleerd?
- Is de doelstelling realistisch?
- Hebben de partners een gemeenschappelijk visie op de manier waarop het doel gerealiseerd moet worden?
- Werkten de partners voor het project al met elkaar samen?
- Hoe groot is de wens om subsidie te ontvangen?

Verdeel ESPRIT projecten nu in twee categorieën. In *idee-georiënteerde* projecten werken partijen samen die ook al vóór dat project met elkaar samenwerkten, die allen de doelstellingen van het project onderschrijven, en die bovendien een gemeenschappelijke ideologie bezitten over de vraag hoe deze doelstellingen gerealiseerd moeten worden. Het ESPRIT project is, kortom, een goede gelegenheid om een al bestaande gemeenschappelijke doelstelling te realiseren.

In *subsidie-georiënteerde* projecten wordt de deelname van partners vooral ingegeven door de wens om subsidie te verwerven middels een ESPRIT project. Daartoe wordt samen met nieuwe partners, een nieuwe doelstelling geformuleerd.

Mijn stelling luidt nu simpelweg:

*idee-georiënteerde projecten slagen,  
subsidie-georiënteerde projecten falen.*

Ik nodig de lezer uit om deze theorie te toetsen aan de eigen ervaringen binnen ESPRIT projecten.

*De andere lessen.* Er zijn nog enkele andere lessen te leren. Allereerst, is het belangrijk om in te zien dat samenwerking vele vormen kan hebben die soms afwijken van wat men daar gewoonlijk onder verstaat. Ik noem:

- Gebruik van elkaars ideeën: de ideeën van andere partners werken als inspiratiebron en stimuleren eigen ontwikkelingen.
- Gebruik van elkaars tools: door import van technologie kan men eigen ontwikkelingen versnellen.

Ten tweede, ik verval in herhalingen, dient “schrijven voor ESPRIT” zoveel mogelijk vermeden te worden.

Ten derde, zijn kleine consortia te prefereren want hoe kleiner de combinatoriek, hoe eenvoudiger het is om consensus te bereiken.

Ten slotte, is het van belang dat elke partner datgene aan een project bijdraagt waarin hij goed is. Dit betekent dat onderzoekers dicht bij hun gebied van expertise moeten blijven (in plaats van modieuze thema’s na te jagen) en

dat bij de taakverdeling tussen onderzoekers en bedrijven ook goed op ieders sterke en zwakke kanten gelet wordt. Het is, bij voorbeeld, een slecht idee om een academicus in te schakelen als productieprogrammeur. Kortom, schoenmaker blijf zoveel mogelijk bij je leest.

## 6 TOT BESLUIT

Laat het duidelijk zijn: ik heb veel aan het ESPRIT programma te danken. Zowel qua ervaring en inhoudelijke samenwerking maar ook, uiteraard, qua financiering van mijn eigen onderzoek. Zonder ESPRIT had ik samen met mijn mede-onderzoekers niet de onderzoeksresultaten kunnen realiseren die nu in internationaal vooraanstaande tijdschriften gepubliceerd zijn maar ook langzaam maar zeker hun weg vinden naar het bedrijfsleven.

De vaak kritische opmerkingen in dit artikel dienen dan ook twee positieve doelen:

- Anderen wijzen op de mogelijke risico's die deelname aan het ESPRIT programma (of aan vergelijkbare andere Europese programma's) met zich mee kan brengen. Om risico's te kunnen vermijden moet men ze immers eerst kennen.
- Dienen als inspiratiebron voor het opzetten van betere vormen van (internationale) samenwerking voor industrie-gericht, strategisch onderzoek.

Dit tweede punt verdient nog enige toelichting. Deelname aan ESPRIT leert mij dat samenwerking gebaseerd moet zijn op projecten met

- Een klein aantal partners (2-4).
- Partners die al bewezen hebben samen te kunnen werken. (Vraag: hoe komt de eerste samenwerking tot stand? Antwoord: het is beter dat deelnemers het samenwerken op eigen kosten leren in plaats van op kosten van de gemeenschap. Deze eis vormt dus een *indirecte* stimulans tot samenwerking.)
- Een doelstelling die innovatief is, doch geen science fiction. (Vraag: wordt het hierdoor niet onmogelijk om revolutionaire vernieuwingen te bereiken? Antwoord: Nee, revolutionaire vernieuwingen zijn meestal het resultaat van fundamenteel onderzoek dat ook als zodanig gefinancierd moet worden. Fundamenteel onderzoek is immers een noodzakelijke voorwaarde voor strategisch onderzoek. Strategische samenwerkingsprojecten met zeer ambitieuze doelstellingen zijn gevaarlijk maar niet onmogelijk. De risico's moeten vooraf alleen beter geanalyseerd worden.)
- Een globale planning en strenge toetsing van de resultaten achteraf.

*Met dank* aan J. Heering en H.R. Walters voor hun commentaar op dit artikel. Deze tekst is eerder gepubliceerd in *Informatie* Nr.4, April 1994 en is met toestemming van de uitgever (Kluwer Bedrijfswetenschappen) in deze bundel opgenomen.



# Special Functions Associated with Root Systems: Recent Progress

Tom H. Koornwinder

*University of Amsterdam, Department of Mathematics,  
Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands  
e-mail: thk@fwi.uva.nl*

## 1. Introduction

It is a pleasure for me to contribute to this farewell bundle for Cor Baayen. During his long period of involvement with the Mathematical Centre he has demonstrated from the beginning a wide interest in all kinds of pure mathematics, see his publications and colloquium contributions on many different topics during the sixties. During his years as a head of the Department of Pure Mathematics (ZW) he will have been aware that some pure math was being done within the MC but outside his department. In the late seventies I joined his department, but quite soon Cor was then called for a higher post: director of the whole institute. Next the departments of Pure and of Applied Mathematics joined into the new Department of Analysis, Algebra and Geometry, headed by Michiel Hazewinkel. In his new position, Cor had to deal with a much bigger world than just pure math. It included all kinds of applied math and a very large amount of computer science. Forced by national science policy and by financial constraints he had to take some unfavorable decisions concerning the heritage of his old ZW department. Fortunately, on a countrywide scale, he could give pure (or theoretical) mathematics some important financial injections via the foundation SMC (mathematics research money flow to the universities), of which he was automatically also a director. In my opinion Cor was a good science manager, showing a real interest in the research being done in his institute and in the people performing this research.

In this paper I present a brief survey of the active area of Special Functions associated with Root Systems. The article is intended for a general mathematical audience. It will not suppose prerequisites on either special functions or root systems. It will also skip many technical details. Some early developments in this area took place at the Mathematical Centre during the seventies ([17], [32]), and some recent developments ([30], [18]) as well. During the last ten years important break-throughs were made by Heckman and Opdam (Leiden; Heckman later in Nijmegen) [11], [27], [28], [12]. Abroad, I. G. Macdonald [23], [24], C. F. Dunkl [8] and I. Cherednik [3], [4], [5] greatly contributed to the subject. A special period at the Stieltjes Institute (physically at the Universities of Leiden and Amsterdam) was devoted to this subject in the spring of 1994. The subject is also an important theme within the four-year country-wide project "Lie Theory and Special Functions", which just started and which is sponsored by SMC.

A lot of the motivation for the subject of this paper comes from analysis on semisimple Lie groups. Spherical functions on Riemannian symmetric spaces of the compact or non-compact type can be written as special functions depending on parameters which assume only special discrete values. In the one-variable cases these special functions are classical, also for parameter values without group theoretic interpretation, but in the more-variable cases they were new. In the case of group theoretic interpretation, many properties of these special functions, as well as associated harmonic analysis, immediately follow by group theoretic arguments. The case of more general parameter values yields the special functions associated with root systems. Properties derived in the group case can still be formulated in the general case, but now as conjectures rather than theorems. This paper describes some of the progress which has been made in proving these conjectures. For convenience, I will restrict to the polynomial (compact) case, with Bessel functions as a sole exception. (For an introduction to the non-polynomial case see [29], [13].) Neither will I discuss the recent work on commuting operators with elliptic functions as coefficients. An important aspect of the whole theory, which will not be discussed very much in this paper, is the connection with completely integrable systems, for instance the generalized Calogero-Moser system.

Special functions associated with root systems have also been developed in the  $q$ -case, where  $q$  is a deformation parameter giving back the earlier cases when  $q = 1$ . Motivation and development of the theory in the  $q$ -case has been quite different from the  $q = 1$  case. Except for the case of Hall polynomials [22], theory was developed [23], [24], [18] without interpretation in group theory. But afterwards quantum groups looked very promising as a natural setting for these polynomials. This had turned out to be true in the one-variable case [19], and very recently some interpretations of more-variable cases on quantum groups were found [25], [10].

In any case, a quantum group interpretation for generic values of the parameters cannot be expected. But, by Cherednik's work [3], [4], [5] we know already another algebraic setting for special functions associated with root systems: affine and graded Hecke algebras [20]. As shown by work of Opdam [29], this new algebraic context also allows harmonic analysis.

## 2. The one-variable case

In this section I will introduce three classical families of special functions, each depending on a real parameter  $k \geq 0$ , and such that the cases  $k = 0$  and  $k = 1$  are elementary. The three families are connected with each other by limit transitions. Later, for each of the families I will discuss generalizations which are associated with root systems.

**2.1. Bessel functions.** Consider *Bessel functions* in a non-standard notation:

$$\mathcal{J}_k(x) := \sum_{j=0}^{\infty} \frac{(-\frac{1}{4}x^2)^j}{(k + \frac{1}{2})_j j!} \quad (x \in \mathbb{R}). \quad (2.1)$$

Here we use the notation for *shifted factorial*:

$$(a)_j := a(a+1)\dots(a+j-1) \quad (j = 1, 2, \dots); \quad (a)_0 := 1.$$

The function  $\mathcal{J}_k$  is related to the Bessel function  $J_\alpha$  in standard notation [9, Ch. VII] by

$$\mathcal{J}_k(x) = \frac{2^{k-\frac{1}{2}} \Gamma(k + \frac{1}{2})}{x^{k-\frac{1}{2}}} J_{k-\frac{1}{2}}(x).$$

Note that

$$\mathcal{J}_k(x) = \mathcal{J}_k(-x), \quad \mathcal{J}_k(0) = 1. \quad (2.2)$$

The cases  $k = 0$  and  $k = 1$  yield elementary functions:

$$\mathcal{J}_0(x) = \cos x, \quad \mathcal{J}_1(x) = \frac{\sin x}{x}. \quad (2.3)$$

The function  $x \mapsto \mathcal{J}_k(\lambda x)$  ( $\lambda \in \mathbb{R}$ ) is eigenfunction of a differential operator:

$$\left( \frac{d^2}{dx^2} + \frac{2k}{x} \frac{d}{dx} \right) \mathcal{J}_k(\lambda x) = -\lambda^2 \mathcal{J}_k(\lambda x).$$

It is the unique  $C^\infty$  solution of this differential equation under conditions (2.2).

**2.2. Ultraspherical polynomials.** Consider *ultraspherical* or *Gegenbauer polynomials* [9, §10.9], i.e. polynomials  $C_n^k$  of degree  $n$  on  $\mathbb{R}$  such that

$$\int_0^\pi C_n^k(\cos x) C_m^k(\cos x) (\sin x)^{2k} dx = 0 \quad (n, m \in \mathbb{Z}_+, n \neq m).$$

Then the  $C_n^k$  are determined up to a constant factor (in general, we will not use the standard normalization for Gegenbauer polynomials). For  $k = 0, 1$  we have:

$$C_n^0(\cos x) = \text{const.} \cos(nx), \quad C_n^1(\cos x) = \text{const.} \frac{\sin((n+1)x)}{\sin x}. \quad (2.4)$$

The function  $x \mapsto C_n^k(\cos x)$  is eigenfunction of a differential operator:

$$\left( \frac{d^2}{dx^2} + 2k \cot x \frac{d}{dx} \right) C_n^k(\cos x) = -n(n+2k) C_n^k(\cos x).$$

For  $(n_N)$  being a sequence of positive integers such that  $n_N/N \rightarrow \lambda$  for some  $\lambda \geq 0$  as  $N \rightarrow \infty$ , we have the limit result

$$\lim_{n \rightarrow \infty} \frac{C_{n_N}^k(\cos(x/N))}{C_{n_N}^k(1)} = \mathcal{J}_k(\lambda x).$$

**2.3.  $q$ -Ultraspherical polynomials.** Let  $0 < q < 1$  and define for any complex  $a$ :

$$(a; q)_\infty := \prod_{j=0}^{\infty} (1 - aq^j).$$

The infinite product converges because of the condition on  $q$ . We will consider  $q$ -ultraspherical polynomials [1] in a non-standard notation. These are polynomials  $C_n^{k,q}$  of degree  $n$  on  $\mathbb{R}$  such that

$$\int_0^\pi C_n^{k,q}(\cos x) C_m^{k,q}(\cos x) \left| \frac{(e^{2ix}; q)_\infty}{(q^k e^{2ix}; q)_\infty} \right|^2 dx = 0 \quad (n, m \in \mathbb{Z}_+, n \neq m).$$

Then the  $C_n^{k,q}$  are determined up to a constant factor. If we put

$$P_n(e^{ix}) := C_n^{k,q}(\cos x)$$

then  $P_n$  is eigenfunction of a  $q$ -difference operator:

$$\frac{1 - q^k e^{2ix}}{1 - e^{2ix}} P_n(q^{\frac{1}{2}} e^{ix}) + \frac{1 - q^k e^{-2ix}}{1 - e^{-2ix}} P_n(q^{-\frac{1}{2}} e^{ix}) = (q^{-\frac{1}{2}n} + q^{\frac{1}{2}n+k}) P_n(e^{ix}).$$

Note that the  $P_n$  on the left hand side have arguments off the unit circle, while orthogonality is on the unit circle. The cases  $k = 0$  and  $k = 1$  are elementary as in (2.4) (not depending on  $q$ ):

$$C_n^{0,q}(\cos x) = \text{const.} \cos(nx), \quad C_n^{1,q}(\cos x) = \text{const.} \frac{\sin((n+1)x)}{\sin x}.$$

With suitable normalization there is the limit relation

$$\lim_{q \uparrow 1} C_n^{k,q}(\cos x) = C_n^k(\cos x).$$

The  $q$ -ultraspherical polynomials form a subclass of the *Askey-Wilson polynomials* [2]: a family of orthogonal polynomials depending, apart from  $q$ , on four non-trivial parameters.

**2.4. Dunkl operators in one variable.** We will now generalize the elementary formulas

$$e^{i\lambda x} = \mathcal{J}_0(\lambda x) + i\lambda x \mathcal{J}_1(\lambda x) \quad \text{and} \quad \frac{d}{dx} e^{i\lambda x} = i\lambda e^{i\lambda x} \quad (2.5)$$

(the first formula follows by (2.3)). Dunkl [8] generalized the operator  $d/dx$  to a mixture of a differential and a reflection operator:

$$(D^{(k)} f)(x) := f'(x) + k \frac{f(x) - f(-x)}{x}. \quad (2.6)$$

Note that this *Dunkl operator* sends smooth functions to smooth functions. Let us define a *generalized exponential function* in terms of Bessel functions (2.1) by

$$\mathcal{E}_k(\lambda x) := \mathcal{J}_k(\lambda x) + \frac{i\lambda x}{2k+1} \mathcal{J}_{k+1}(\lambda x). \quad (2.7)$$

Then it follows immediately from well-known differential recurrence formulas for Bessel functions that

$$D^{(k)} \mathcal{E}_k(\lambda x) = i\lambda \mathcal{E}_k(\lambda x). \quad (2.8)$$

Formulas (2.7) and (2.8) generalize the formulas in (2.5). The function  $x \mapsto \mathcal{E}_k(\lambda x)$  is the unique  $C^\infty$  function which equals 1 in 0 and which is eigenfunction with eigenvalue  $i\lambda$  of  $D^{(k)}$ .

For  $(D^{(k)})^2$  we compute

$$(D^{(k)})^2 f(x) = f''(x) + \frac{2k}{x} f'(x) - k \frac{f(x) - f(-x)}{x^2}.$$

Thus, on even functions  $f$  the square of the Dunkl operator acts as the differential operator  $(d/dx)^2 + 2kx^{-1} d/dx$ . In particular, its action on

$$\mathcal{J}_k(\lambda x) = \frac{1}{2}(\mathcal{E}_k(\lambda x) + \mathcal{E}_k(-\lambda x))$$

yields

$$(D^{(k)})^2 \mathcal{J}_k(\lambda x) = -\lambda^2 \mathcal{J}_k(\lambda x).$$

### 3. Preliminaries about root systems

**3.1. Definition of root system.** Let  $V$  be a  $d$ -dimensional real vector space with inner product  $\langle \cdot, \cdot \rangle$ . For  $\alpha \in V \setminus \{0\}$  let  $s_\alpha$  denote the orthogonal reflection with respect to the hyperplane orthogonal to  $\alpha$  (cf. Fig. 1):

$$s_\alpha(\beta) := \beta - \frac{2\langle \beta, \alpha \rangle}{\langle \alpha, \alpha \rangle} \alpha \quad (\beta \in V).$$

A *root system* in  $V$  (see [15]) is a finite subset  $R$  of  $V \setminus \{0\}$  which spans  $V$  and which satisfies for all  $\alpha, \beta \in R$  the two properties that

$$s_\alpha(\beta) \in R \quad \text{and} \quad \frac{2\langle \beta, \alpha \rangle}{\langle \alpha, \alpha \rangle} \in \mathbb{Z}.$$

Clearly, if  $\alpha \in R$  then  $-\alpha = s_\alpha(\alpha) \in R$ . For convenience, we will restrict ourselves to the case of a *reduced root system*, i.e., a root system  $R$  such that, if  $\alpha, \beta \in R$  and  $\alpha = c\beta$  for some  $c \in \mathbb{R}$ , then  $c = \pm 1$ . The so-called *irreducible root systems* can be classified as four infinite families  $A_n, B_n, C_n, D_n$  of *classical root systems* and five *exceptional root systems*  $G_2, F_4, E_6, E_7, E_8$ . Here the subscript denotes the *rank* of the root system, i.e. the dimension of  $V$ . There is one infinite family of non-reduced irreducible root systems: of type  $BC_n$ .

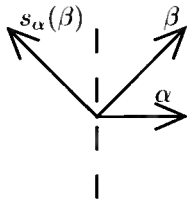


Fig. 1. Reflection  $s_\alpha$



Fig. 2. Root system  $A_1$

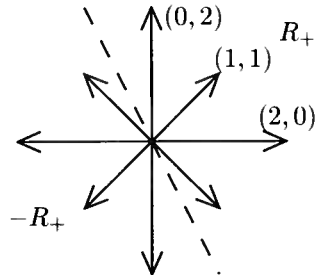


Fig. 3. Root system  $C_2$

An example for  $d = 1$  is the set  $R := \{\pm 2\} \subset \mathbb{R}$  (root system of type  $A_1$ , cf. Fig. 2). An example for  $d = 2$  is the set  $R = R_+ \cup (-R_+)$ , where  $R_+ := \{(1, -1), (2, 0), (1, 1), (0, 2)\} \subset \mathbb{R}^2$  (root system of type  $C_2$ , cf. Fig. 3). In general, when we have a root system  $R$  in  $V$  then we can write it as a disjoint union  $R = R_+ \cup (-R_+)$ , where  $R_+$  and  $-R_+$  are separated from each other by a hyperplane in  $V$  through the origin. The choice for  $R_+$  is not unique. The elements of  $R$  are called *roots* and the elements of  $R_+$  are called *positive roots*.

Let  $GL(V)$  be the group of invertible linear transformations of  $V$ . The *Weyl group*  $W$  of the root system  $R$  is the subgroup of  $GL(V)$  which is generated by the reflections  $s_\alpha$  ( $\alpha \in R$ ). The group  $W$  is finite and it acts on  $R$ . It permutes the possible choices of  $R_+$  in a simply transitive way.

**3.2. Dunkl operators associated with  $R$ .** Let  $R$  be a root system in  $V$ . Let  $k: \alpha \mapsto k_\alpha: R \rightarrow [0, \infty)$  be a function which is  $W$ -invariant, i.e., which satisfies  $k_{w\alpha} = k_\alpha$  for all  $w \in W$  and all  $\alpha \in R$ . If  $R$  is an irreducible (reduced) root system then the Weyl group is transitive on all roots of equal length and there are at most two different root lengths. Thus  $k_\alpha$  then assumes at most two different values. See the above examples: one root length in  $A_1$  and two root lengths in  $C_2$ . The function  $k$  is called a *multiplicity function*. The reason for this name is that root systems naturally arise in the structure theory of real semisimple Lie algebras, where roots have an interpretation as joint eigenvalues of certain operators and the  $k_\alpha$  then are (integer) multiplicities of such eigenvalues.

For  $\xi \in V$  we will denote by  $\partial_\xi$  the corresponding directional derivative. The *Dunkl operators* [8], [16] associated with the root system  $R$  and the multiplicity function  $k$  are defined as the operators  $D_\xi^{(k)}: C^\infty(V) \rightarrow C^\infty(V)$  ( $\xi \in V$ ) given by

$$(D_\xi^{(k)} f)(x) := (\partial_\xi f)(x) + \sum_{\alpha \in R_+} k_\alpha \langle \alpha, \xi \rangle \frac{f(x) - f(s_\alpha x)}{\langle \alpha, x \rangle}. \quad (3.1)$$

This definition is easily seen to be independent of the choice of  $R_+$ . In case of root system  $A_1$  formula (3.1) reduces for  $\xi := 1$  to formula (2.6). Note that the operator (3.1) consists of a term involving a first order derivative and terms involving reflection operators, just as we have seen in (2.6). It is an amazing fact, which can be proved in a rather straightforward way, that the operators  $D_\xi^{(k)}$  commute:

$$[D_\xi^{(k)}, D_\eta^{(k)}] = 0 \quad (\xi, \eta \in V).$$

Let  $\mathbb{D}^{(k)}$  be the algebra generated by the operators  $D_\xi^{(k)}$ . This is a commutative algebra. It can be shown that each  $W$ -invariant operator  $D$  in  $\mathbb{D}^{(k)}$ , when restricted in its action to the  $W$ -invariant  $C^\infty$  functions on  $V$ , coincides with a partial differential operator (so its reflection terms vanish when acting on a  $W$ -invariant function). The joint  $W$ -invariant eigenfunctions of the  $W$ -invariant operators in  $\mathbb{D}^{(k)}$  are called *Bessel functions associated with  $R$* . In the example  $A_1$  things reduce to the one-variable considerations of §2.1 and §2.4. More generally, one may study the joint eigenfunctions of the full algebra  $\mathbb{D}^{(k)}$  and one

may try to do harmonic analysis for these eigenfunctions. A lot of satisfactory results have been obtained, see [16] and the references given there.

**3.3. Weight lattice associated with  $R$ .** We still assume a root system  $R$  in  $V$ . The *weight lattice*  $P$  of  $R$  is defined [15] by

$$P := \{\lambda \in V \mid \frac{2\langle \lambda, \alpha \rangle}{\langle \alpha, \alpha \rangle} \in \mathbb{Z} \text{ for all } \alpha \in R\}.$$

The subset  $P_+$  of *dominant weights* is then given by

$$P_+ := \{\lambda \in P \mid \frac{2\langle \lambda, \alpha \rangle}{\langle \alpha, \alpha \rangle} \geq 0 \text{ for all } \alpha \in R_+\}.$$

It is easily seen that  $w(P) = P$  for  $w \in W$ , so the Weyl group acts on  $P$ . Moreover, it can be shown that each Weyl group orbit in  $P$  has a one-point intersection with  $P_+$ :

$$\forall \lambda \in P \quad \text{Card}(W\lambda \cap P_+) = 1.$$

Thus the dominant weights can be used as a set of representatives for the  $W$ -orbits in  $P$ .

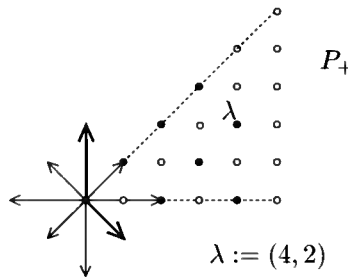


Fig. 4. Root system  $C_2$  with dominant weights and the set  $\{\mu \in P_+ \mid \mu \prec \lambda\}$

We introduce a partial ordering on  $P$  which is induced by the root system: for  $\lambda, \mu \in P$  we say that  $\mu \prec \lambda$  if  $\lambda - \mu = \sum_{\alpha \in R_+} m_\alpha \alpha$  for certain nonnegative integers  $m_\alpha$ . For root system  $C_2$  the concepts of this subsection are illustrated in Fig. 4.

**3.4. Trigonometric polynomials associated with  $R$ .** Let  $P$  be the weight lattice of a root system  $R$  in  $V$ . For  $\lambda \in P$  define the function  $e^\lambda$  on  $V$  by

$$e^\lambda(x) := e^{i\langle \lambda, x \rangle} \quad (x \in V).$$

Note that  $e^\lambda e^\mu = e^{\lambda+\mu}$ . Thus the space

$$\mathcal{A} := \text{Span}\{e^\lambda \mid \lambda \in P\}$$

is an algebra: the algebra of trigonometric functions on  $V$  (with respect to  $R$ ). For a function  $f$  on  $V$  write  $(wf)(x) := f(w^{-1}x)$  ( $w \in W, x \in V$ ). Then  $w e^\lambda = e^{w\lambda}$  ( $w \in W, \lambda \in P$ ). Put

$$m_\lambda := \sum_{\mu \in W\lambda} e^\mu \quad (\lambda \in P_+).$$

Then the functions  $m_\lambda$  are  $W$ -invariant and they form a basis of the space  $\mathcal{A}^W$  of  $W$ -invariant elements in  $\mathcal{A}$ .

Let the dual root lattice  $Q^*$  be defined by

$$Q^* := \{\lambda \in V \mid \langle \lambda, \mu \rangle \in \mathbb{Z} \text{ for all } \mu \in P\}.$$

This lattice gives rise to a torus

$$T := V/(2\pi Q^*).$$

Let  $x \mapsto \dot{x}$  be the natural mapping of  $V$  onto  $T$ . Then each function  $f$  in  $\mathcal{A}$  actually lives on  $T$ :  $f(x) = \tilde{f}(\dot{x})$  for a suitable function  $\tilde{f}$  on  $T$ .

In the example  $A_1$  we have  $P = \mathbb{Z}$ ,  $P_+ = \{0, 1, 2, \dots\}$ , the algebra  $\mathcal{A}$  is spanned by the functions  $x \mapsto e^{inx}$  ( $n \in \mathbb{Z}$ ) and the subalgebra  $\mathcal{A}^W$  by the functions 1 and  $x \mapsto 2 \cos(nx)$  ( $n = 1, 2, \dots$ ). The torus  $T$  equals  $\mathbb{R}/(2\pi\mathbb{Z})$ .

## 4. Jacobi polynomials associated with $R$

**4.1. Definition of Jacobi polynomials for  $R$ .** Let  $R$  be a root system in  $V$  and let  $k: R \rightarrow [0, \infty)$  be a  $W$ -invariant multiplicity function as before. Define a weight function  $\delta_k$  on  $T$  by

$$\delta_k(x) := \prod_{\alpha \in R_+} |2 \sin(\langle \alpha, x \rangle)|^{2k_\alpha}. \quad (4.1)$$

This definition is independent of the choice of  $R_+$ . Define an inner product on the linear space  $\mathcal{A}$  by

$$\langle f, g \rangle_k := \int_T f(x) \overline{g(x)} \delta_k(x) d\dot{x} \quad (f, g \in \mathcal{A}). \quad (4.2)$$

Here  $d\dot{x}$  denotes Lebesgue measure on  $T$ , normalized such that the volume of  $T$  is equal to 1.

The Jacobi polynomial  $P_\lambda^{(k)}$  (cf. [11]) of “degree”  $\lambda \in P_+$  and of “order”  $k$  is an element of  $\mathcal{A}^W$  of the form

$$P_\lambda^{(k)} = \sum_{\substack{\mu \in P_+ \\ \mu \prec \lambda}} c_{\lambda, \mu} m_\mu$$



such that  $c_{\lambda,\lambda} = 1$  and

$$\langle P_\lambda^{(k)}, m_\mu \rangle_k = 0 \quad \text{if } \mu \in P_+ \text{ and } \mu \not\preceq \lambda. \quad (4.3)$$

Instead of (4.3) we can equivalently require that  $P_\lambda^{(k)}$  satisfies the second order differential equation

$$\left( \Delta + \sum_{\alpha \in R_+} k_\alpha \cot(\frac{1}{2} \langle \alpha, x \rangle) \partial_\alpha \right) P_\lambda^{(k)}(x) = - \langle \lambda, \lambda + \sum_{\alpha \in R_+} k_\alpha \alpha \rangle P_\lambda^{(k)}(x). \quad (4.4)$$

In the example  $A_1$  we obtain that  $P_n^{(k)}(x) = \text{const. } C_n^k(\cos x)$ , where  $C_n^k$  is the ultraspherical polynomial of §2.2. The case of the (non-reduced) root system  $BC_1$  would have given us, more generally, the classical one-variable Jacobi polynomials.

**4.2. Three problems and their solutions.** As soon as the above definition of Jacobi polynomials associated with  $R$  is given, three highly nontrivial questions can naturally be posed:

1. It follows immediately from the definition that the orthogonality

$$\langle P_\lambda^{(k)}, P_\mu^{(k)} \rangle_k = 0 \quad (4.5)$$

holds if  $\mu \preceq \lambda$  or  $\lambda \preceq \mu$ . What about (4.5) if  $\lambda$  and  $\mu$  are not related in the partial ordering?

2. Prove the existence of a commutative algebra of differential operators with  $d$  algebraically independent generators, such that the operators in this algebra have the  $P_\lambda^{(k)}$  ( $\lambda \in P_+$ ) as joint eigenfunctions. (Note that the operator in (4.4) can be taken as one of the generators.)
3. Give an explicit expression for  $\langle P_\lambda^{(k)}, P_\lambda^{(k)} \rangle_k$ , or rather for its two factors

$$\frac{\langle P_\lambda^{(k)}, P_\lambda^{(k)} \rangle_k}{\langle P_0^{(k)}, P_0^{(k)} \rangle_k} \quad \text{and} \quad \int_T \delta_k(x) dx. \quad (4.6)$$

In the past few years all these questions have been answered in the positive sense. Let me give some indications.

- If problem 2 can be solved then the answer to 1 follows readily, cf. [11]. Indeed, we need sufficiently many differential operators having the  $P_\lambda^{(k)}$  as eigenfunctions such that the joint eigenvalues, in their dependence on  $\lambda$ , separate the points of  $P_+$ .
- For certain special choices of  $k$  the functions  $P_\lambda^{(k)}$ , renormalized such that  $P_\lambda^{(k)}(0) = 1$ , have an interpretation as spherical functions on compact symmetric spaces  $G/K$ , cf. [14]. (For instance, in case  $A_1$  the ultraspherical polynomial  $C_n^{\frac{1}{2}m-1}$  can be interpreted as spherical function on the  $(m-1)$ -dimensional sphere  $SO(m)/SO(m-1)$ .) Then problems 1, 2 and the first half of problem 3 can be solved by using the group theoretic interpretation.

The orthogonality (4.5) for general  $\lambda, \mu$  follows by Schur's orthogonality relations for matrix elements of irreducible unitary representations of  $G$ . The first expression in (4.6) was explicitly computed by Vretare [33] in terms of Harish-Chandra's  $c$ -function related to the spherical functions on the corresponding non-compact symmetric space. The algebra of differential operators in problem 2 can be obtained by taking the radial parts of the  $G$ -invariant differential operators on  $G/K$ .

- For the classical root systems question 2 could be answered in a positive way by giving explicit expressions for generators of the algebra, see [17] for  $BC_2$  and  $A_2$ , and Olshanetsky & Perelomov [26], Sekiguchi [31] and Debiard [6] for the higher rank cases.
- Heckman and Opdam [11] have given positive answers to 2, and hence to 1, by use of deep transcendental arguments. This also solved part of Problem 3 (the first expression in (4.6)). In 1982 Macdonald [21] had already given conjectures for the explicit evaluation of the second expression in (4.6), which could be proved in a number of special cases.
- Problem 3 for general  $\lambda$  was solved by Opdam [28] by using so-called *shift operators* [27]. The most simple example, for case  $A_1$ , of such operators is the following pair of differential recurrence relations for Gegenbauer polynomials:

$$\begin{aligned} \frac{d}{dx} C_n^k(x) &= \text{const. } C_{n-1}^{k+1}(x), \\ \left( (1-x^2)^{-k+\frac{1}{2}} \frac{d}{dx} \circ (1-x^2)^{k+\frac{1}{2}} \right) C_{n-1}^{k+1}(x) &= \text{const. } C_n^k(x). \end{aligned}$$

By use of these two formulas we can write  $\int_{-1}^1 (C_n^k(x))^2 (1-x^2)^{k-\frac{1}{2}} dx$  as an explicit constant times  $\int_{-1}^1 (C_{n-1}^{k+1}(x))^2 (1-x^2)^{k+\frac{1}{2}} dx$ . Opdam's shift operators in general have a similar structure of lowering  $\lambda$  and raising  $k$ , or conversely. The case of root system  $BC_2$  was already considered in [17], [32].

**4.3. Dunkl type operators.** Some years after Heckman first solved the problems 1 and 2 of the previous subsection he discovered a dramatical simplification [12] for proving these results. For a given root system  $R$  in  $V$  and a given multiplicity function  $k$  he wrote down a trigonometric variant of the Dunkl operators (3.1) for  $\xi \in V$ :

$$\begin{aligned} (D_\xi^{(k)} f)(x) &:= (\partial_\xi f)(x) + \frac{1}{2} \sum_{\alpha \in R_+} k_\alpha \langle \alpha, \xi \rangle \cot(\frac{1}{2} \langle \alpha, x \rangle) (f(x) - f(s_\alpha x)) \\ &(x \in V, f \in C^\infty(V)). \end{aligned} \quad (4.7)$$

Now the operators  $D_\xi^{(k)}$  will no longer commute, in general. However, Heckman showed that the operators  $\sum_{\eta \in W\xi} (D_\eta^{(k)})^j$  ( $\xi \in V, j = 0, 1, 2, \dots$ ), when restricted to the  $W$ -invariant  $C^\infty$  functions on  $V$ , coincide with differential operators which commute with each other and form a commutative algebra. This is the algebra looked for in Problem 2 of the previous subsection. The Jacobi

polynomials  $P_\lambda^{(k)}$  are the joint eigenfunctions of the operators in this algebra. By this approach, Heckman also obtained a quick existence proof for Opdam's shift operators.

Next Cherednik [3] made a slight but significant variation in Heckman's Dunkl type operators (4.7). He put

$$\begin{aligned}
 (\tilde{D}_\xi^{(k)} f)(x) := & (\partial_\xi f)(x) + \sum_{\alpha \in R_+} k_\alpha \langle \alpha, \xi \rangle \frac{1}{1 - e^{-\alpha}(x)} (f(x) - f(s_\alpha x)) \\
 & - \frac{1}{2} \sum_{\alpha \in R_+} k_\alpha \langle \alpha, x \rangle f(x).
 \end{aligned}$$

(Here I took the part of the right hand side on the second line from Opdam [13, p.86]; Cherednik is not very specific about this part of his formula.) Cherednik's operators have the nice property that they mutually commute, without the need of first restricting to  $W$ -invariant functions. On the other hand, they do not share the property  $w D_\xi^{(k)} w^{-1} = D_{w\xi}^{(k)}$  of Heckman's operators. Anyhow, by means of Cherednik's operators one can draw the same conclusions as by Heckman's operators, and in a similar way. Moreover, a structure of graded Hecke algebra can be associated with Cherednik's operators.

## 5. Macdonald polynomials associated with $R$

**5.1. Definition of Macdonald polynomials.** Let  $0 < q < 1$ . We keep the assumptions of §4.1 except that we replace the weight function  $\delta_k$  in (4.1) by

$$\delta_{k,q}(x) := \prod_{\alpha \in R_+} \left| \frac{(e^{i\langle \alpha, x \rangle}; q)_\infty}{(q^{k_\alpha} e^{i\langle \alpha, x \rangle}; q)_\infty} \right|^2.$$

Then the *Macdonald polynomials*  $P_\lambda^{(k,q)}$  were defined by Macdonald [23], [24] just as the Jacobi polynomials  $P_\lambda^{(k)}$ , but with the inner product in (4.2) replaced by

$$\langle f, g \rangle_{k,q} := \int_T f(x) \overline{g(x)} \delta_{k,q}(x) dx \quad (f, g \in \mathcal{A}).$$

In the case of root system  $A_1$  the Macdonald polynomials coincide with the  $q$ -ultraspherical polynomials  $x \mapsto C_n^{k,q}(\cos x)$ . For any root system  $R$ , in the limit for  $q \uparrow 1$ , the Macdonald polynomial  $P_\lambda^{(k,q)}$  tends to the corresponding Jacobi polynomial  $P_\lambda^{(k)}$ .

Macdonald gives some explicit  $q$ -difference operators of which the  $P_\lambda^{(k,q)}$  are eigenfunctions. Although these operators, except for root system  $A_n$  (where they were independently found by Ruijsenaars [30]) do not yet give a full commutative algebra of operators having the  $P_\lambda^{(k,q)}$  as joint eigenfunctions, the additional parameter  $q$  gives enough freedom such that already the eigenvalue of one such operator separates the elements of  $P_+$  for generic  $q$ , by which a

positive answer to question 1 in §4.2 can be given for the case of Macdonald polynomials. Taking limits for  $q \uparrow 1$  then yields the same positive answer for the case of Jacobi polynomials. This is an alternative to Heckman's approach via Problem 2. Macdonald also gives conjectured explicit expressions for the squared norms  $\langle P_\lambda^{(k,q)}, P_\lambda^{(k,q)} \rangle_{k,q}$ .

**5.2. Askey-Wilson polynomials for root system  $BC_n$ .** The author [18] introduced for the non-reduced root system  $BC_n$  a class of polynomials having two more parameters than Macdonald's class for  $BC_n$ . This extended class reduces for  $n = 1$  to the Askey-Wilson polynomials [2]. In [18] only one explicit  $q$ -difference operator was given having the  $BC_n$ -polynomials as eigenfunctions, but this was sufficient for establishing orthogonality. Later, van Diejen [7] gave explicit expressions for the generators of a full commutative algebra of operators having the  $BC_n$  polynomials as joint eigenfunctions.

**5.3. Cherednik's approach to Macdonald polynomials.** Cherednik [4], [5] succeeded to give positive answers to questions 2 and 3 in §4.2. In the context of certain representations of affine Hecke algebras he could realize a commutative algebra of operators which have the Macdonald polynomials as joint eigenfunctions. In the same context he could realize  $q$ -analogues of Opdam's shift operators and next, by the same technique as in Opdam, prove Macdonald's conjectures in the  $q$ -case.

It is beyond the scope of this short survey to explain Cherednik's approach in any detail. In May 1994 I. G. Macdonald delivered some very helpful lectures in Leiden in order to explain Cherednik's approach. Let me here only give a few indications. Just as a Hecke algebra is a deformation of the group algebra of a Weyl group, an affine Hecke algebra (cf. [20]) deforms the group algebra of an affine Weyl group. If  $R$  is an irreducible root system in  $V$  with Weyl group  $W$  then the (extended) affine Weyl group is the semidirect product  $\widetilde{W} := W \ltimes P^-$ , where the dual weight lattice  $P^-$  is defined as  $P^- := \{\lambda \in V \mid \langle \lambda, \alpha \rangle \in \mathbb{Z}\}$ , an abelian group under addition. Then  $\widetilde{W}$  acts as a group of motions on  $V$ , with  $P^-$  acting as a group of translations. The group  $\widetilde{W}$  also acts on  $\mathcal{A}$ , with  $W$  acting as before and with the action of  $P^-$  still depending on a parameter  $q$ .

The affine Hecke algebra  $H$  can be defined in terms of generators and relations which still depend on the values of a  $W$ -invariant function  $\alpha \mapsto t_\alpha$  on  $R$ . Corresponding to a choice of  $R_+$  we can define  $P_+^-$ . The embedding of  $P_+^-$  in  $H$  then generates a commutative subalgebra  $\mathcal{Y}$  of  $H$ .

For given  $q$  we can use the action of  $\widetilde{W}$  on  $\mathcal{A}$  in order to define an action of  $H$  on  $\mathcal{A}$ , by specifying the action for a set of generators of  $H$  (Demazure operators). This action depends on  $q$  and the  $t_\alpha$ . Put  $t_\alpha = q^{-k_\alpha/2}$ , where  $k: R \rightarrow [0, \infty)$  is a multiplicity function. Then Cherednik proves that the Macdonald polynomials  $P_\lambda^{(k,q)}$  are the joint eigenfunctions of the  $W$ -invariant elements in the commutative algebra  $\mathcal{Y}$ . This answers question 2 in §4.2.

At the moment it is still an open problem to extend Cherednik's approach to the  $BC_n$  polynomials. As van Diejen [7] already answered question 2 in §4.2 in a positive way for this case, it would be nice to complement van Diejen's

constructive approach with the deep conceptual approach via affine Hecke algebras.

## References

- [1] R. Askey & M. E. H. Ismail, *A generalization of ultraspherical polynomials*, in *Studies in Pure Mathematics*, P. Erdős (ed.), Birkhäuser, 1983, pp. 55–78.
- [2] R. Askey & J. Wilson, *Some basic hypergeometric orthogonal polynomials that generalize Jacobi polynomials*, *Memoirs Amer. Math. Soc.* (1985), no. 319.
- [3] I. Cherednik, *A unification of Knizhnik-Zamolodchikov and Dunkl operators via affine Hecke algebras*, *Invent. Math.* 106 (1991), 411–431.
- [4] I. Cherednik, *Double affine Hecke algebras, Knizhnik-Zamolodchikov equations, and Macdonald's operators*, *Duke Math. J.*, *Internat. Math. Res. Notices* (1992), no. 9, 171–180.
- [5] I. Cherednik, *The Macdonald constant-term conjecture*, *Duke Math. J.*, *Internat. Math. Res. Notices* (1993), no. 6, 165–177.
- [6] A. Debiard, *Système différentiel hypergéométrique et parties radiales des opérateurs invariants des espaces symétriques de type  $BC_p$* , in *Séminaire d'Algèbre Paul Dubreil et Marie-Paule Malliavin*, M.-P. Malliavin (ed.), *Lecture Notes in Math.* 1296, Springer, 1988, pp. 42–124.
- [7] J. F. van Diejen, *Commuting difference operators with polynomial eigenfunctions*, *Compositio Math.*, to appear.
- [8] C. F. Dunkl, *Differential-difference operators associated to reflection groups*, *Trans. Amer. Math. Soc.* 311 (1989), 167–183.
- [9] A. Erdélyi, W. Magnus, F. Oberhettinger & F. G. Tricomi, *Higher transcendental functions, Vol. II*, McGraw-Hill, 1953.
- [10] P. I. Etingof & A. A. Kirillov, Jr., *Macdonald's polynomials and representations of quantum groups*, *Math. Res. Lett.* 1 (1994), no. 3, 279–296.
- [11] G. J. Heckman (part I with E. M. Opdam), *Root systems and hypergeometric functions I, II*, *Compositio Math.* 64 (1987), 329–352, 353–373.
- [12] G. J. Heckman, *An elementary approach to the hypergeometric shift operators of Opdam*, *Invent. Math.* 103 (1991), 341–350.
- [13] G. J. Heckman (part I, II) and E. M. Opdam (part III), *Hypergeometric functions and representation theory*, in “Seminar Hypergeometric Functions”, *Communications of the Mathematical Institute* 18, Utrecht University, 1994, pp. 72–93.
- [14] S. Helgason, *Groups and geometric analysis*, Academic Press, 1984.
- [15] J. E. Humphreys, *Introduction to Lie algebras and representation theory*, Springer, 1972.
- [16] M. F. E. de Jeu, *Dunkl operators*, Dissertation, University of Leiden, 1994.

- [17] T. H. Koornwinder, *Orthogonal polynomials in two variables which are eigenfunctions of two algebraically independent partial differential operators I, II, III, IV*, Nederl. Akad. Wetensch. Proc. Ser. A 77 (1974), 48–58, 59–66, 357–369, 370–381.
- [18] T. H. Koornwinder, *Askey-Wilson polynomials for root systems of type BC*, in *Hypergeometric functions on domains of positivity, Jack polynomials, and applications*, D. St. P. Richards (ed.), Contemp. Math. 138, Amer. Math. Soc., 1992, pp. 189–204.
- [19] T. H. Koornwinder, *Askey-Wilson polynomials as zonal spherical functions on the  $SU(2)$  quantum group*, SIAM J. Math. Anal. 24 (1993), 795–813.
- [20] G. Lusztig, *Affine Hecke algebras and their graded version*, J. Amer. Math. Soc. 2 (1989), 599–635.
- [21] I. G. Macdonald, *Some conjectures for root systems*, SIAM J. Math. Anal. 13 (1982), 988–1007.
- [22] I. G. Macdonald, *Symmetric functions and Hall polynomials*, Oxford University Press, 1979; second ed. 1994.
- [23] I. G. Macdonald, *A new class of symmetric functions*, in *Séminaire Lotharingien de Combinatoire, 20e Session*, L. Cerlienco & D. Foata (eds.), Publication de l'Institut de Recherche Mathématique Avancée 372/S-20, Strasbourg, 1988, pp. 131–171.
- [24] I. G. Macdonald, *Orthogonal polynomials associated with root systems*, preprint, 1988.
- [25] M. Noumi, *Macdonald's symmetric polynomials as zonal spherical functions on some quantum homogeneous spaces*, Adv. in Math., to appear.
- [26] M. A. Olshauetsky & A. M. Perelomov, *Quantum integrable systems related to Lie algebras*, Phys. Rep. 94 (1983), 313–404.
- [27] E. M. Opdam, *Root systems and hypergeometric functions III, IV*, Compositio Math. 67 (1988), 21–49, 191–209.
- [28] E. M. Opdam, *Some applications of hypergeometric shift operators*, Invent. Math. 98 (1989), 1–18.
- [29] E. M. Opdam, *Harmonic analysis for certain representations of graded Hecke algebras*, Report W 93-18, Dept. of Math., Univ. of Leiden, 1993.
- [30] S. N. M. Ruijsenaars, *Complete integrability of relativistic Calogero-Moser systems and elliptic function identities*, Comm. Math. Phys. 110 (1987), 191–213.
- [31] J. Sekiguchi, *Zonal spherical functions on some symmetric spaces*, Publ. Res. Inst. Math. Sci. 12 Suppl. (1977), 455–459.
- [32] I. G. Sprinkhuizen, *Orthogonal polynomials in two variables. A further analysis of the polynomials orthogonal on a region bounded by two lines and a parabola*, SIAM J. Math. Anal. 7 (1976), 501–518.
- [33] L. Vretare *Elementary spherical functions on symmetric spaces*, Math. Scand. 39 (1976), 343–358.

# Adaptive Spline-Wavelet Image Encoding and Real-Time Synthesis on a VLSI Difference Engine for Image Generation

*To Cor Baayen, at the occasion of his retirement*

A.A.M. Kuijk, P.C. Marais and E.H. Blake

The low level components of a new raster graphics architecture developed at the CWI have proven to have novel uses in image reconstruction. The display hardware can be regarded as a very fast (11ns per operation) Difference Engine that works in two-dimensions. The speed is partly achieved by the use of custom VLSI components for the most primitive operations and this permits the video rate reconstruction of images and other signals compressed by encoding them on various polynomial bases. A wavelet-based image-encoding is described which, when used in conjunction with the Difference Engine allows us to reconstruct an image in real-time without the need to set each pixel explicitly. The image is compressed using a quadratic spline-wavelet transform; when reconstructing, an image-adaptive instruction generator attempts to produce the minimal instruction stream to give a good reproduction. The wavelet coefficients are used to decide which regions of the detail images should be retained in the multi-resolution analysis (MRA). A decision is made for each scanline as to whether it is more economical, in terms of rendering time, to use the 'truncated MRA' or to set the pixels directly. The above approach provides a significant gain over standard image reconstruction/rendering schemes.

## 1 INTRODUCTION

A radical reappraisal of the three-dimensional (3-D) interactive raster graphics pipeline has resulted in an experimental architecture for a graphics workstation which is currently being evaluated at the CWI. Some of the novel uses of parts of the hardware were not foreseen when the research project was initiated.

Principal features of the design for the new raster graphics architecture are:

1. Emphasis on real-time interactive shaded 3-D graphics.
2. Object space methods rather than image space methods are used where possible.
3. Avoids the use of a frame buffer.
4. Uses custom VLSI only at the lowest, most primitive, levels where commercial products are unlikely to suffice in the near term.



It was these design decisions that lead to a number of interesting consequences that have made parts of the architecture eminently suited to a far wider range of problems in computer graphics and image processing. The initial top-down design produced an architecture for raster graphics (only). The bottom-up design that followed concentrated on extracting the lowest common denominator of primitive operations for synthesizing pixels — a language for manipulating related pixels. This vocabulary can be used for expressing other facts about images. For example, the custom VLSI development that was a major part of the project produced what is essentially a very *fast Difference Engine* (to borrow a term from the 19th century history of computation). This engine can compute forward differences in parallel over the whole width of a typical image, taking about 11ns per operation (90 Mhz clock) independently of the length of the forward difference spans. It was recognized that this feature would be useful for image reconstruction as well.

Studies have shown that for image reconstruction the *wavelet transform* [3] offers a better compression/fidelity tradeoff than the Discrete Cosine Transform (DCT)[4]. The complexity of the blocked DCT is of the same order as that of an (unblocked) fast wavelet transform — consequently, blocking is not required and blocking artifacts are no longer a problem. Furthermore, the *multi-resolution* structure of the transform allows for resolution-dependent coding techniques.

The ‘standard’ approach to image synthesis, after such transform coding, is to perform an inverse transform, thus producing the data required for each pixel. However, by requiring that our image be expressible on a suitably defined (quadratic) spline basis, and using the properties of the Difference Engine, it is possible to regenerate the image, progressively, if this is desired, from a *subset* of the full MRA, by examining the transform coefficients which underlie the analysis. This synthesis procedure allows one to reduce the number of instructions required to render an image, when compared with the direct approach.

## 2 THE WAVELET TRANSFORM

A *wavelet*,  $\Psi(x, y)$ , is an  $L^2(\mathbb{R}^2)$  function which satisfies

$$\iint \Psi(x, y) dx dy = 0 \tag{1}$$

This condition ensures that the wavelet is localized in both time and frequency and exhibits a measure of oscillation — hence the name. The *discrete (dyadic) wavelet transform*,  $(W_\Psi I)(j; i, l)$  of an  $L^2(\mathbb{R}^2)$  function,  $I(x, y)$ , with respect to the wavelet  $\Psi$  is defined as

$$(W_\Psi I)(j; i, l) = \langle \Psi_{j;i,l}, I \rangle, \quad i, j, l \in \mathbb{Z} \tag{2}$$

where  $\langle \cdot, \cdot \rangle$  denotes the  $L^2$  inner product and  $\Psi_{j;i,l}(x, y) \equiv 2^j \Psi(2x - i, 2y - l)$ . For non-orthogonal wavelets, there is a corresponding *dual wavelet*,  $\tilde{\Psi}$ , which



satisfies the relationship

$$\langle \Psi_{k;i,p}, \tilde{\Psi}_{l;j,q} \rangle = \delta_{kl} \delta_{ij} \delta_{pq}. \quad (3)$$

It can be shown that the functions  $\{\Psi_{j;i,l}; j, k, l \in \mathbb{Z}\}$  span the space  $L^2(\mathbb{R}^2)$  [3]. Hence, any function,  $I(x, y)$ , in this space can be written as a linear combination of such scaled and translated wavelets:

$$I(x, y) = \cdots + g_{-1}(x, y) + g_0(x, y) + g_1(x, y) + \cdots \quad (4)$$

where

$$g_j(x, y) = \sum_{i,l} d_{j;i,l} \Psi_{j;i,l}(x, y), \quad j \in \mathbb{Z}. \quad (5)$$

Because of the *bi-orthogonality relation*, Equation (3), one may write  $d_{j;i,l} = \langle I, \tilde{\Psi}_{j;i,l} \rangle$ ,  $i, j, l \in \mathbb{Z}$ .

### 3 MULTI-RESOLUTION ANALYSIS

The concept of a *Multi-Resolution Analysis* (MRA) is already familiar to those who have dealt with pyramidal image decompositions; it serves to formalize such a decomposition. Firstly, one must define the term “resolution”. The intuitive interpretation, viz., that it serves to quantify the amount of permissible variation in a region, is formalized. Hence, a high resolution image has a large amount of detail in a region, whereas a low resolution image is much smoother over this same region. One may further quantify this concept with a statement such as: “a  $k$ th resolution image contains  $k \times k$  samples per unit square”. The idea here is that we can capture more detail if we are able to sample at a higher rate.

To develop the theory of such an analysis, we first consider the case of one dimensional signals.

Our signal,  $f(x)$ , must be an elements of the space  $L^2(\mathbb{R})$ , that is, it must contain finite energy. We seek a decomposition of this signal which will reveal its structure on different ‘resolution’ levels. Such an analysis can provide invaluable information about the relative importance of variations in the signal.

Each of these *multi-resolution approximations* resides in a space which contains all possible approximations at that resolution of every  $L^2(\mathbb{R})$  function. These spaces are denoted  $V_j$ ; the parameter  $j$  indicates the resolution level: the “resolution” of the  $j$ th level is given by  $r = 2^j$ . Thus, level 0 has  $r = 1$ . By convention, this is the input level.

Just as the wavelet spaces<sup>1</sup>  $W_j$  are spanned by the scaled translates of a single kernel function,  $\psi$ , we seek a single function,  $\phi$ , the so-called *scaling function*, which will span the spaces  $V_j$  in the same way. If this is the case, then we may define a Multi-Resolution Analysis of  $L^2(\mathbb{R})$ . Since we desire that this analysis be complete, the MRA must encode the detail that is sacrificed

<sup>1</sup> $W_j \equiv \text{clos}_{L^2} \text{span}\{\psi_{jk} : k \in \mathbb{Z}\}$ ; the operation of CLOSure essentially adds all the limit points to a space, thus ‘closing’ it up.

when we go from a higher to a lower resolution. This detail is stored in the complementary *wavelet* spaces,  $W_j$ . We have the following relationship for any resolution level  $j$

$$V_{j+1} = V_j \dot{+} W_j \tag{6}$$

This states that the higher resolution approximation may be resynthesized from the next lower approximation by adding the detail that we sacrificed to achieve that lower approximation. One can deduce the following properties:

1.  $\dots \subset V_{-1} \subset V_0 \subset V_1 \subset \dots$ ;
2.  $\text{clos}_{L^2} \left( \bigcup_j V_j \right) = L^2(\mathcal{R})$ ;
3.  $\bigcap_j V_j = \{0\}$ ;
4.  $V_{j+1} = V_j \dot{+} W_j, \quad j \in \mathcal{Z}$ ;
5.  $f(x) \in V_j \iff f(2x) \in V_{j+1}, \quad j \in \mathcal{Z}$ .

For a more detailed discussion and alternative formulation of these properties, see [1].

The space  $W_j$  is the the orthogonal complement of the space  $V_j$  in  $V_{j+1}$ . The spaces  $W_j$  are spanned by  $\psi_{j,i}(x) \equiv 2^{j/2}\psi(2^j x - i)$ , where  $\psi(x)$  is a 1-D wavelet, satisfying the 1-D analogue of Equation (1). The spaces  $V_j$  are spanned by scaled and translated versions of a so-called *scaling function*,  $\phi(x)$ . The approximation spaces  $V_j$  contains the  $j$ th resolution approximation,  $f_j(x)$ , of the input function,  $f(x)$ , while the *detail* spaces,  $W_j$ , contain the information lost when going from a  $(j+1)$ th level approximation to the  $j$ th level approximation.

A common method used to generate a 2-D MRA, is to take the tensor product of the corresponding 1-D multi-resolution analysis with itself [3]. This provides one with *three* wavelets,  $\Psi^{[p]}(x, y)$ ,  $p = 1, 2, 3$  and a scaling function,  $\Phi(x, y)$ , all of which are separable 2-D functions:

$$\Psi^{[1]}(x, y) = \phi(x)\psi(y) \tag{7}$$

$$\Psi^{[2]}(x, y) = \psi(x)\phi(y) \tag{8}$$

$$\Psi^{[3]}(x, y) = \psi(x)\psi(y) \tag{9}$$

$$\Phi(x, y) = \phi(x)\phi(y) \tag{10}$$

These wavelets are essentially orientated, resolution-dependent band-pass filters; the scaling function may be viewed as a low-pass filter. The detail spaces, spanned by each wavelet type, thus contain difference information with a specific orientation only: vertical, horizontal and diagonal.

The multi-resolution pyramid goes off to infinity in both directions. However, realisable signals are band-limited. Thus, we truncate the representation, discarding all higher level information, by ‘projecting’ our input function into a space which has sufficient detail to represent the sampled signal —  $V_0$  by convention. Similarly, since signals do not always contain arbitrarily low frequencies, it may be unnecessary to decompose one’s signal beyond a certain

level. Thus, one has a  $J$ th level multi-resolution decomposition

$$\begin{aligned}
 I(x, y) &\equiv I_0(x, y) \\
 &= g_{-1}(x, y) + \cdots + g_{-J}(x, y) + I_{-J}(x, y) \\
 &= \sum_{l=-1}^{-J} \sum_{i,j} \sum_{p=1}^3 d_{[p]l;i,j} \Psi_{l;i,j}^{[p]}(x, y) + \\
 &\quad \sum_{i,j} c_{-J;i,j} \Phi_{-J;i,j}(x, y). \tag{11}
 \end{aligned}$$

The wavelet transform is also truncated; the  $J$ th level discrete wavelet transform provides the set of coefficients

$$\{\{d_{[p]l;i,j}\}, \{c_{-J;i,j}\}, i, j \in \mathbb{Z}, l = -1, -2, \dots - J; p = 1, 2, 3\} \tag{12}$$

where the *detail coefficients* are obtained as follows

$$d_{[p]l;i,j} = \langle \tilde{\Psi}_{l;i,j}^{[p]}, I \rangle, i, j \in \mathbb{Z}. \tag{13}$$

Formally, the *approximation coefficients* are given by

$$c_{l;i,j} = \langle \tilde{\Phi}_{l;i,j}, I \rangle, i, j \in \mathbb{Z} \tag{14}$$

where  $\tilde{\Phi}(x, y)$  is the *dual scaling function*. The approximation coefficients,  $c_{l;i,j}$ , encode the present in the lower levels of the multi-resolution pyramid.

#### *Semi-Orthogonal Cardinal Spline MRA*

The space of cardinal splines of order  $m$ ,  $S_m$ , contains all those functions expressible as a weighted sum of  $m$ th order *cardinal B-splines*,  $N_m(x)$ :

$$f(x) = \sum c_k N_m(x - k), f \in S_m. \tag{15}$$

The values of  $N_m(x)$  may be found using the following identity:

$$N_m(x) = \frac{x}{m-1} N_{m-1}(x) + \frac{m-x}{m-1} N_{m-1}(x-1). \tag{16}$$

$$N_m(x) \equiv (N_{m-1} * N_1)(x) = \int_0^1 N_{m-1}(x-t) dt, m \geq 2, \tag{17}$$

where

$$N_1(x) = \chi_{[0,1)}(x) = \begin{cases} 1 & \text{if } x \in [0, 1); \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

The cardinal B-splines are thus generated by repeatedly convolving the unit box with itself. Figure 1 shows some of these functions.

Cardinal B-splines satisfy the following identity, which enables one to compute their values without resorting to integral formulations:

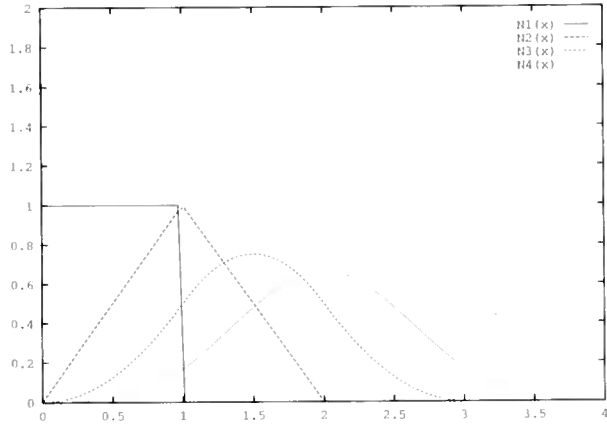


FIGURE 1. Spline scaling functions. The cardinal spline scaling functions are generated by repeatedly convolving  $N_1(x)$  with itself.

The spline-based MRA introduced in [5, 6] has  $N_m(x)$  as its scaling function.

The corresponding  $m$ th order spline wavelet,  $\psi_m(x)$ , has support on the interval  $[0, 2m - 1]$ . This wavelet is *semi-orthogonal*, meaning that it is orthogonal to scaled versions of itself, but not to translates on the same resolution level. These functions satisfy the following *two-scale* relationships

$$N_m(x) = \sum_{k=0}^m p_k N_m(2x - k), \quad (19)$$

$$\psi_m(x) = \sum_{k=0}^{3m-2} q_k N_m(2x - k) \quad (20)$$

The values of these sequences, for the quadratic case, can be found in [5].

#### 4 CALCULATION OF THE WAVELET COEFFICIENTS

Before one can use the MRA, a means must be found to compute the coefficients of the wavelet transform. To this end we use the filtering scheme proposed in [7]. In the context of this work, this gives us the following set of separable convolutional equations for computing the detail and approximation coefficients (from the approximation coefficients of the previous level):

$$c_{j-1:kl} = \sum_m \sum_n a_{m-2k} a_{n-2l} c_{j:mn} \quad (21)$$

$$d_{[1]j-1:kl} = \sum_m \sum_n a_{m-2k} b_{n-2l} c_{j:mn} \quad (22)$$

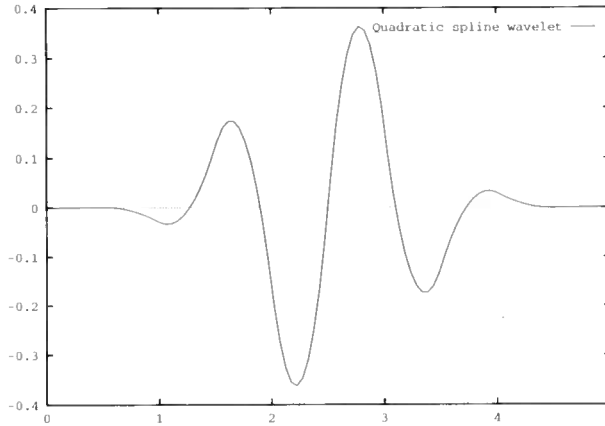


FIGURE 2. A quadratic spline wavelet.

$$d_{[2]_{j-1};kl} = \sum_m \sum_n b_{m-2k} a_{n-2l} c_{j;mn} \quad (23)$$

$$d_{[3]_{j-1};kl} = \sum_m \sum_n b_{m-2k} b_{n-2l} c_{j;mn}. \quad (24)$$

To reconstruct the approximation coefficients (from those lower down in the analysis), one has the following *reconstruction relation*:

$$\begin{aligned} c_{j;km} = & \sum_l \sum_t p_{k-2l} p_{m-2l} c_{j-1;lt} + \\ & \sum_l \sum_t p_{k-2l} q_{m-2l} d_{[1]_{j-1};lt} + \\ & \sum_l \sum_t q_{k-2l} p_{m-2l} d_{[2]_{j-1};lt} + \\ & \sum_l \sum_t q_{k-2l} q_{m-2l} d_{[3]_{j-1};lt}. \end{aligned} \quad (25)$$

The  $\{a_k\}$  and  $\{b_k\}$  sequences can be found in [8].

#### *Calculation of $\{c_{0;i,j}\}$*

In order that we can use the filtering scheme above, one must first generate the initial set of approximation coefficients,  $\{c_{0;i,j}\}$  — which are the basis coefficients of the B-spline representation of the input image. If one just wants to achieve compression, the image samples may be used as the initial coefficient values. If, however, one wishes to evaluate the MRA, then these values must be properly computed.

We use *quasi-interpolation* [9] to obtain these coefficients. Quasi-interpolation is a local interpolation scheme, in which the amount of data used to determine the approximating quasi-interpolant can be limited. In this work a 3x3 convo-

lution mask ( $k = 1$ , below) was used to determine the required coefficients:

$$c_{0:ij} = (\lambda_k I)(i, j), \quad i, j \in \mathbb{Z}, \quad I \in L^2(\mathbb{R}^2). \quad (26)$$

This sequence is computed as

$$\{(\lambda_k I)(i)\} = (\delta - m + \dots + (-1)^k \underbrace{m * \dots * m}_{k \text{ times}}) * I^0(i), \quad i \in \mathbb{Z}^2, \quad (27)$$

where  $\delta \equiv \delta_{i,j;0} = 1$  if  $i, j = 0$ , and 0 otherwise and

$$m_{i,j} = \begin{cases} \Phi(0, 0) - 1 & \text{for } i, j = 0; \\ \Phi(i, j) & \text{for } i, j \neq 0. \end{cases} \quad (28)$$

Because the B-splines must be centred [9],  $\Phi(x, y) = N_3(x + 3/2)N_3(y + 3/2)$ , and the coefficient values actually represent the shifted image  $I_0(x + 3/2, y + 3/2)$ . It is important to remember this shift when evaluating image functions in the MRA.

## 5 QUANTIZATION

We used *vector quantization* to compress the wavelet encoded image. The approach of [10] was used: the various wavelet sub-bands were sub-divided into 2x2 or 4x4 blocks (as determined by the desired compression ratio) and these blocks were quantized with the previously trained codebooks to yield 8-bit indices (thus permitting 256 reproduction levels per sub-band). The LBG algorithm with a minimum mean-squared error measure was used [11]. The codebook was trained with a collection of disparate images, so as not to introduce any kind of image bias; the test images were not in the training sequence. As is done elsewhere, for example [10, 12], the entropy of the coefficient sequence is used as a measure of compression i.e., we assume that the quantization is followed by a perfect entropy coding.

## 6 THE DIFFERENCE ENGINE

The Difference Engine is the final component in the rendering pipeline of a new display architecture developed at CWI [13]. This display processor has the ability to interpolate an arbitrary length polynomial span with a single instruction, in time proportional to the degree of the polynomial. The forward difference interpolatory logic is implemented as a systolic array — each new cycle produces the complete set of difference values for the specified span. An  $n$ th degree polynomial span may be specified by a starting point, a set of  $n$  forward differences and the width of the span. The  $p$ th order forward difference of  $I(x)$  is

$$(\Delta_p I)(x) = (\Delta_{p-1} I)(x + 1) - (\Delta_{p-1} I)(x), \quad (29)$$

where

$$(\Delta_0 I)(x) = I(x). \quad (30)$$

Once the required differences are computed, using the simple recursive scheme presented above, the polynomial values at uniformly spaced intervals ( $\mathbb{Z}$ , in this case) may be obtained by using the following simple update rule

$$(\Delta_p I)(x+1) = (\Delta_p I)(x) + (\Delta_{p+1} I)(x), \quad p = 0, \dots, n-1. \quad (31)$$

for consecutive values of  $x$ . The 11ns cycle time of this processor means that one can perform these calculations with sufficient speed to ensure pixel production at the display refresh rate.

The proposed architecture does not employ a framebuffer. Instead, the image is represented as a list of primitives and the objects selected from this list are converted into Difference Engine instructions by customized hardware, at a sufficient rate to provide real-time video display. The complexity of the image determines the size of the list and consequently the number of instructions which are produced.

There are two important points which should be noted:

- the Difference Engine can interpolate arbitrary order polynomials, in time proportional to the degree (currently  $n + 2$  cycles for a polynomial of degree  $n - 1$ ).
- the Difference Engine provides a scanline accumulator.

The Difference Engine can interpolate polynomial spans accurately up to a length dependent on the degree of the polynomial — currently about 4096 pixels for a quadratic and 512 pixels for a cubic. This limit poses no problems, since the image data can be segmented into several spans if the need arises, which is unlikely if one uses the quadratic scheme.

The existence of an intensity accumulator is essential if one wishes to use the Difference Engine for multi-resolution image synthesis, since one then needs to accumulate several levels of detail for each scanline.

## 7 MULTI-RESOLUTION IMAGE SYNTHESIS

The various images in the quadratic cardinal spline MRA satisfy certain very stringent conditions:

- They are elements of  $C^1(\mathbb{R}^2)$
- The approximation images consist of quadratic patches, with support on

$$[2^j k, 2^j(k+1)] \times [2^j k, 2^j(k+1)], \quad k \in \mathbb{Z}$$

- The detail images also have this property, but over squares half the size on the resolution level  $j$ .

These conditions are a consequence of the tensor product used to generate the MRA and the properties possessed by the prototype 1-D MRA. Thus, the

image data along a scanline (on each level) is composed of adjacent quadratic segments of the same length. It is a simple matter to compute the differences for any such polynomial (using the shifted image functions), and to compose the Difference Engine instructions which will interpolate the polynomial scanline data.

If used without care, multi-resolution synthesis can be far more expensive (in terms of Difference Engine instruction cycles) than just setting each pixel directly, since many instructions must be issued to accumulate all the detail information for each scanline. If however, only ‘busy’ regions of the detail images are added back to the approximation image, this ‘truncated’ MRA can provide significant gains over direct reconstruction (i.e., IWT and setting each pixel directly). Wavelet compression should maintain only the most important coefficients viz. those which will ensure good reconstruction fidelity. These retained coefficients can be used as an indication of ‘busy’ image areas, and the bases which they weight can be used to build the truncated MRA. We determine the extents of these bases which intersect the current scanline — this information is recorded and used to determine whether it is more economical (in terms of Difference Engine instruction cycles required) to simply set the pixels in the current scanline or to render the truncated MRA. If the latter option is selected, the function evaluations are done and the tiers of detail are accumulated on top of the approximation signal. If it is less economical (as will be the case in highly detailed regions), the scanline pixels are set directly.

Due to the continuity constraints, and the architecture of the chip, we need only issue one quadratic interpolation instruction to interpolate the entire approximation scanline: only the second order differences need be changed as we cross each new span boundary. These can be computed and set before the interpolation instruction is issued, by using a low cost set-difference instruction. A similar strategy can be used for detail scanline segments consisting of several adjacent spans.

To improve performance, neighbouring quadratic spans are merged if their differences are the same; this reduces the number of instructions required to interpolate a multi-span segment. However, since this kind of redundancy is only likely to occur in the approximation image, merging is not applied to detail scanline segments. Furthermore, for reasons of efficiency, the merging procedure is not applied prior to deciding what kind of synthesis method to employ. Doing so would require additional calculations which would be wasted if direct synthesis were used.

## 8 RESULTS

### 8.1 *Wavelet Compression*

It was apparent that the fidelity of the reconstructed images left something to be desired, even at modest bit-rates (around 1 bpp) — Figure 4. There are a number of reasons for this lack of performance, in particular, the use of a MMSE distortion metric, which takes no account of edge information and does





FIGURE 3. **Test Images.** The (8-bit greyscale) images are Lenna, House and Sugarbowl.



FIGURE 4. Typical VQ compression result — 0.82 bpp.

not guarantee simultaneous minimization of reconstruction error and transform domain quantization error (since Parseval's identity does not hold in a semi-orthogonal framework). Simple thresholding tests revealed that MMSE VQ was not exploiting the redundancy provided by the wavelet transform effectively.

### 8.2 *Image Synthesis*

The results given below are based on a three level wavelet decomposition in which, rather than applying VQ, the wavelet coefficients were thresholded and those retained were used in the MR synthesis calculations. This was done to decouple the compression implementation from the synthesis algorithms, since the former retained too many (unrepresentative) coefficients to illustrate the concepts referred to earlier. The thresholding used is adapted to orientation and resolution level and forms part of the new compression scheme we are investigating. To enable us to quantify the gains produced by MR synthesis, we introduce the Gain Factor (GF) — the ratio of the instruction cycles required to render the image directly to the number of cycles required if adaptation is used. The GF is always  $\geq 1.0$ .

Table 1 summarizes the results of this preliminary work. Observe that two sets of data are given: the first uses the current cycle costs for the relevant instruction<sup>2</sup> while the second uses the cycle costs which will be used in subsequent implementations of the Difference Engine.

	Cycles	Lenna	House	Sugarbowl
Setddi	2	0	4	62
Eval0	1	100	95	33
Eval1	3	0	0.5	0.2
Eval3	5	0	0.5	4.8
GF	-	1.23	1.63	3.43
MR	-	No	Yes	Yes

	Cycles	Lenna	House	Sugarbowl
Setddi	1	1.4	7.6	65.7
Eval0	1	97.5	87.0	13.9
Eval1	1	1.0	4.7	14.7
Eval3	3	0.1	0.7	5.7
GF	-	1.22	1.57	3.86
MR	-	Yes	Yes	Yes

TABLE 1. **Synthesis Results.** The first four rows of each table give the percentages each of the instruction types contributed to the final rendering cost. The final row indicates whether multi-resolution synthesis was invoked or not. The same threshold was employed with all images. The second table gives the figures when the proposed lower cost instructions are used.

There are several things which were evident from our experiments. Firstly, the smoothness of an image is directly related to the gains obtainable when using MR synthesis: the more texture the image possess, the less likely MR synthesis is to yield any benefit, unless the texture is highly localised. In the latter case, the non-textured scanlines can still be rendered more cheaply. Secondly, image detail is expensive to render, because a) it is present on multiple levels of the MRA and b) the quadratic spans are smaller and consequently more instructions are required to interpolate a scanline. This is the motivation for truncating the MRA.

Images which are themselves composed of splines (such as the Phong shaded images in [13], of which ‘Sugarbowl’ is an example) will experience greater gains than other (smooth) images. However, the extent of this reduction will depend on the size of the spline patches of which the image is composed and for most images these are fairly small. The Difference Engine is ideally suited

<sup>2</sup>The interpolation instructions are of the form ‘eval $n$ ’, where  $n$  is the order of the polynomial to be interpolated; ‘eval0’ switches off accumulation of subsequent pixel values at the given location, otherwise acting like an ‘eval1’ — since it is cheaper, it is used for direct reconstruction. The ‘setddi’ instruction can be used to set the second difference at a specified point; subsequent interpolations, passing through this point, will use this value rather than the one they had been propagating.

to rendering such images.

The images 'Sugarbowl' and 'House' were both able to derive varying degrees of benefit from MR synthesis, since there were regions in which the intensity data varied slowly. 'Lenna' contains a lot of texture; but with lower instruction costs, it becomes economical to use the MRA on some scanlines. In highly uniform or smooth images, span merging on the approximation level can become significant, boosting rendering efficiency substantially. An extreme example of this would be an object on a uniform background; the background would only be present in the approximation image and could be generated very quickly and efficiently.

For highly textured images, when we are forced to chose direct reconstruction, we can still gain by merging neighbouring pixels; this saves one having to set each pixel individually. Since pixels are usually correlated, even the most chaotic of images may benefit (albeit marginally) from such merging. In the examples given above, Lenna experienced a GF of 1.23 from such pixel merging: all neighbouring pixels along a scanline which are within one gray scale of the first pixel considered are approximated by this initial value, and a zero-degree polynomial (*eval0*) of the appropriate length is emitted. When using MR synthesis, smooth images can yield very large gains (a GF of  $> 3$  for non-trivial images like Sugarbowl). The *nature* of the smoothness plays an important role in determining the magnitude of these gains i.e., is the image actually a spline, or just smoothish? True spline images can be approximated with fewer resolution levels and coefficients.

Although not explicitly indicated in the tables above, the level of the decomposition has a very definite affect on the rendering gains one can achieve. If the number of levels is too low, then one gains nothing in rendering time, since short pixel spans (less than the order of the polynomial) must be set directly. If, on the other hand, the number of levels is too high, then too much information must be accumulated from the detail tiers and the rendering efficiency drops. A three level decomposition appears to be optimal.

The Difference Engine is able to produce low resolution approximation images very efficiently, since the spline patches are then quite large (the 3rd level approximation of Lenna can be rendered in a quarter of the time required to render the full image, using the old instruction costs). Progressive transmission is possible if the receiver is equipped with a screen buffer in which incoming information can be accumulated.

## 9 CONCLUSION & FUTURE WORK

Although the implementation of the quantization algorithm was inadequate, the compression potential of the spline WT can be exploited by a better algorithm. Smooth images can be rendered more rapidly using MR synthesis than by direct reconstruction. Even heavily textured images can be rendered more efficiently if zero-degree pixel merging is applied to exploit pixel correlation.

A better quantization system is currently under development. Work can be

done to improve the usability of the Difference Engine w.r.t. MR synthesis — the Difference Engine was not specifically designed to render this kind of structure. One of the modifications that can be made, is the addition of a screen-wide accumulator which the Difference Engine can access to enable efficient rendering of progressively transmitted images. Work can also be done to improve the simple efficiency measures used — the emphasis here was on rendering performance, which assumes that the MR data can be produced at an adequate rate. All the required information can be computed using parallelised FFT hardware — so on the face of it, this assumption is a reasonable one. Nonetheless, one may desire a different measure of efficiency.

#### ACKNOWLEDGMENT

We would like to thank CWI (Centre for Mathematics and Computer Science) for supporting this research as well as the South African Foundation for Research Development (FRD).

#### REFERENCES

1. H. J. Heijman. Discrete wavelets and multiresolution analysis. In Koornwinder [2], pages 49–79. This article originally appeared in CWI Quarterly, Vol. 5, No. 1, March 1992.
2. T. H. Koornwinder, editor. *Wavelets: An Elementary Treatment of Theory and Applications*, volume 1 of *Approximations and Decompositions*. World Scientific, 1993.
3. I. Daubechies, *Ten Lectures on Wavelets*, SIAM, vol 61 in CBMS-NSF series in applied mathematics, 1992.
4. P. Desarte, B. Macq, D. Slock, *Signal-adapted Multiresolution Transform for Image Coding*, IEEE Trans. on Info. Theory, vol 38, no 2, 1992, pp719–746.
5. C.K. Chui, *An Introduction to Wavelets: Wavelet Analysis and its Applications*, Academic Press, 1992, Boston.
6. M. Unser, A. Aldroubi, M. Eden, *A family of polynomial spline wavelet transforms*, Signal Processing, vol 30, 1993, pp 141–162.
7. S. Mallat, *A Theory of Multiresolution Signal Decomposition: The Wavelet Representation*, IEEE Trans. on Patt. Ana. and Mach Intell, vol 11, 1989, pp 674–693.
8. P. Marais, E. Blake, A. Kuijk, *A Cardinal-Spline Image Decomposition On A Systolic Array Display Processor*, CWI Quarterly, vol 4, 1993.
9. C.K. Chui and H. Diamond, *A Natural Formulation of Quasi-Interpolation by Multi-Variate Splines*, Proceedings of the American Mathematical Society, vol 99, no 4, 1987.
10. M. Antonini and M. Barlaud and P. Mathieu and I. Daubechies, *Image Coding using Wavelet Transforms*, IEEE trans. on image processing, vol 1, no 2, 1992, pp 205–220.

11. Y. Linde, A. Buzo, R. M. Gray, *An algorithm for vector quantizer design*, IEEE trans. on comm, vol com-28, no 1, 1980, pp 84–95.
12. P. Westerink, D. Boekee, J. Biemond, *Sub-band coding of images using vector quantization*, IEEE trans on comm, vol 36, no 6, 1988, pp 713–719.
13. E. H. Blake and A.A.M. Kuijk, *A Difference Engine For Images With Applications To Wavelet Decomposition*, Proceedings of the Second International Conference on Image Communications (IMAGE'COM), 1993, pp 309–314.



# Systematic Computations on Mertens' Conjecture and Dirichlet's Divisor Problem by Vectorized Sieving

*Dedicated to Cor Baayen, at the occasion of his retirement  
as scientific director of SMC and its CWI*

Walter M. Lioen

*CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands*

and

Jan van de Lune

*Noordermiedweg 31, 9074 LM Hallum, The Netherlands*

In this paper we present two vectorized numerical sieve algorithms for the number theoretical functions  $\mu(n)$  and  $\tau(n)$ . These sieve algorithms are generalizations of Eratosthenes' sieve for finding prime numbers. We show algorithms for fast systematic computations on Mertens' conjecture and Dirichlet's divisor problem. We have implemented the algorithm for Mertens' conjecture on a Cray C90 and performed a systematic computation of extremes of  $M(x)/\sqrt{x}$  up to  $10^{13}$ . We established the bounds  $-0.513 < M(x)/\sqrt{x} < 0.571$ , valid for  $200 < x \leq 10^{13}$ .

## 1 INTRODUCTION

Eratosthenes' sieve is one of the oldest algorithms in number theory (3rd century B.C.). The ultimate sieving device for Eratosthenes' sieve and its generalizations is a (parallel) vector computer or a massively parallel computer. Our generalizations of Eratosthenes' sieve are devised with large (parallel) vector computers in mind. They are virtually 100 percent vectorizable and they become more and more efficient when the amount of memory increases.

We start by introducing Mertens' conjecture in Section 2. Section 3 is devoted to a completely vectorized algorithm for a systematic computation of  $M(x)$  and analysis of  $M(x)/\sqrt{x}$ . In Section 4 we describe Dirichlet's divisor problem. The corresponding algorithm is given in Section 5. This algorithm in its turn is a generalization of the algorithm described in Section 3. In Section 6 a selection of the numerical results for  $M(x)/\sqrt{x}$ ,  $x = 1, \dots, 10^{13}$ , is presented. Finally, in the last section we give some concluding remarks.

## 2 MERTENS' CONJECTURE

The Möbius function  $\mu(n)$  is defined as follows

$$\mu(n) = \begin{cases} 1, & n = 1, \\ 0, & \text{if } n \text{ is divisible by a prime square,} \\ (-1)^k, & \text{if } n \text{ is the product of } k \text{ distinct primes.} \end{cases}$$

We consider  $M(x)$ , the first summatory function of  $\mu(n)$ ,

$$M(x) = \sum_{n \leq x} \mu(n).$$

$M(x)$  describes the difference between the number of squarefree positive integers  $n \leq x$  with an even number of prime factors and those with an odd number of prime factors.

Based on a table of  $M(x)$  for  $x = 1, \dots, 10000$  Mertens [11] conjectured that

$$|M(x)| < \sqrt{x}, \quad x > 1.$$

Later, based on more extensive numerical 'evidence', Von Sterneck [17] even conjectured that

$$|M(x)| < \frac{1}{2}\sqrt{x}, \quad x > 200.$$

The Möbius function is related to the Riemann zeta function by

$$\frac{1}{\zeta(s)} = \sum_{n=1}^{\infty} \frac{\mu(n)}{n^s}, \quad \Re(s) > 1.$$

Boundedness of  $M(x)/\sqrt{x}$  implies the truth of the Riemann hypothesis. However, the converse does not hold.

For the history of the function  $M(x)/\sqrt{x}$  and the disproofs of Von Sterneck's conjecture and later Mertens' conjecture—both first theoretical and later effective—we refer to [16]. A comprehensive bibliography may be found in the paper by Odlyzko and Te Riele [13] in which they disprove Mertens' conjecture.

Although it is known that  $M(x)/x \rightarrow 0$  as  $x \rightarrow \infty$  (and even more than this), the best known effective asymptotic upper bound on  $|M(x)|$  to date [4] is

$$|M(x)| \leq \frac{1}{2360}x, \quad x \geq 617973.$$

## 3 A VECTORIZED ALGORITHM FOR $M(x)/\sqrt{x}$

### 3.1 Eratosthenes' sieve

Eratosthenes indicated the following method of obtaining all the primes in the range  $2, \dots, N$ : put all numbers between 2 and  $N$  into a 'sieve'; as long as the sieve is not empty, select the smallest number remaining in the sieve, and strike out all multiples of this prime number. The complexity of both the



sieve initialization and the prime number selection is  $\mathcal{O}(N)$ . The complexity of striking out all multiples of the prime numbers found and therewith the complexity of Eratosthenes' sieve is

$$\sum_{i=1}^{\pi(\sqrt{N})} \left\lfloor \frac{N}{p_i} \right\rfloor \sim N \log \log \sqrt{N},$$

where  $\pi(x)$  denotes the number of prime numbers not exceeding  $x$ . Usually, one only sieves the odd numbers. Moreover, if  $N$  becomes large one has to partition the sieve interval. Even for  $N$  large,  $10^{13}$ , say,  $\log \log \sqrt{N}$  is fairly small. This gives an almost linear complexity  $\mathcal{O}(N)$ . For the sake of completeness: the best (sub)linear prime number sieve has complexity  $\mathcal{O}(N/\log \log N)$ , cf. [10, 15].

### 3.2 Sieving $\mu(n)$

The following algorithm yields the Möbius function  $\mu(n)$  for  $n = 1, \dots, N$ .

```

for  $n = 1$  to  $N$ 
     $\mu(n) = 1$ 
for all  $p \leq \sqrt{N}$ 
    for all  $n, p \mid n$ 
         $\mu(n) = -p \cdot \mu(n)$ 
for all  $p \leq \sqrt{N}$ 
    for all  $n, p^2 \mid n$ 
         $\mu(n) = 0$ 
for  $n = 1$  to  $N$ 
    if  $|\mu(n)| \neq n$  then
         $\mu(n) = -\mu(n)$ 
for  $n = 1$  to  $N$ 
     $\mu(n) = \text{sign}(\mu(n))$ 

```

This algorithm starts initializing a sieve array  $\mu$  with the value 1. Besides the sieve array we also keep a list of all primes not exceeding  $\sqrt{N}$ . Next, for all prime numbers  $p$  not greater than  $\sqrt{N}$  we multiply  $\mu(n)$  by  $-p$  for every  $n$  a multiple of  $p$ . By multiplying with  $-p$  we achieve two things: 1. we multiply by  $p$  in order to see if we end up having handled all prime factors of  $n$ ; 2. by multiplying with  $-p$  instead of  $p$ , we keep track of the parity of the number of different prime factors of  $n$  handled so far. For all prime numbers  $p$  not greater than  $\sqrt{N}$  we set  $\mu(n)$  to 0 for every  $n$  a multiple of  $p^2$ . After this step we check whether  $|\mu(n)| = n$  holds. If  $|\mu(n)| = n$  holds,  $n$  is squarefree and none of its prime factors is greater than  $\sqrt{N}$ . If  $|\mu(n)| = n$  does not hold, we have two possibilities: either  $\mu(n) = 0$ , in which case  $n$  is not squarefree, or  $n$  is squarefree and has exactly one prime divisor  $p > \sqrt{N}$ . Anyhow, if  $|\mu(n)| = n$  does not hold, we just change the sign of  $\mu(n)$ , taking care of the parity for this last prime factor, or a no-op if  $\mu(n) = 0$ . At this point we have three possibilities: 1.  $\mu(n) = 0$ , if  $n$  is not squarefree; 2.  $\mu(n) < 0$ , if  $n$  is the product

of an odd number of distinct primes; 3.  $\mu(n) > 0$ , if  $n$  is the product of an even number of distinct primes. With the obvious definition of *sign*, the last loop in the algorithm above completes the computation of the Möbius function  $\mu(n)$  for  $n = 1, \dots, N$ . It is easy to see that the complexity for the above algorithm is the same as for Eratosthenes' sieve:  $\mathcal{O}(N \log \log \sqrt{N})$ .

As we already mentioned for Eratosthenes' sieve, we have to partition the sieve interval if  $N$  becomes large. The determination of whether a prime hits a partition and, if so, the first index it hits, is a non-vectorizable process. In order not to lose vector speed one should choose the partition size considerably (10–100 times, say) larger than the number of sieve primes. In our computations  $N$  equals  $10^{13}$ , so the number of sieve primes becomes  $\pi(\sqrt{N}) = 227,647$ . We chose our partition size equal to  $10^7$ .

### 3.3 Small prime variation

Since we want to compute all values of  $M(x)$  systematically, we can not halve the amount of work by only sieving the odd numbers, as we can for Eratosthenes' sieve. For the same reason we can not apply a 'small prime variation' as in MPQS [14]. However, it is possible to apply a different kind of small prime variation: replace the initialization  $\mu(n) = 1$  by a 'block-initialization'. Using the small primes 2, 3, 5, 7, 11, say, and also the small prime squares 4, 9, we get a pattern-length of

$$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 2 \cdot 3 = 13,860.$$

Sieving with only these few primes and prime squares requires

$$\left\lfloor \frac{N}{2} \right\rfloor + \left\lfloor \frac{N}{3} \right\rfloor + \left\lfloor \frac{N}{5} \right\rfloor + \left\lfloor \frac{N}{7} \right\rfloor + \left\lfloor \frac{N}{11} \right\rfloor + \left\lfloor \frac{N}{4} \right\rfloor + \left\lfloor \frac{N}{9} \right\rfloor \approx 1.6N$$

sieve updates. If we store the initial pattern of length 13,860 and do a periodic block initialization of the sieve array with this pattern (instead of the total initialization with 1), we get about  $1.6N$  sieve updates, for these small primes and prime powers, for free.

We did not tell the full story by stating that one can not manage sieving only the odd numbers. As pointed out by Tijdeman [18] one may use the identity  $\mu(n) + \mu(2n) = 0$ , for  $n$  odd, together with the identity  $\mu(4n) = 0$ , to avoid the computation of  $\mu(n)$  for  $n$  even. However, for  $N$  large, so that we have to partition the sieve array to get the job done, this becomes impractical because one would have to store some  $N/4$  intermediate  $\mu(n)$ -values for  $n$  even.

### 3.4 Vectorizing the partial summation

Thus far we described a vectorized algorithm for the systematic computation of the Möbius function. Eventually, however, we are interested in the extremes of  $M(x)/\sqrt{x}$ , so that first of all we have to compute the partial sums

$$M(x) = \sum_{n \leq x} \mu(n), \quad x = 1, \dots, N.$$

Phrasing this as an algorithm one might compute the partial sums as follows.

```
M(1) = μ(1)
for x = 2 to N
    M(x) = μ(x) + M(x - 1)
```

The previous loop is a classical example of a non-vectorizable loop because of its recursion on  $M(x - 1)$ .

Assuming that the array  $M$  initially contains the values of  $\mu$  (we do this computation in-place anyhow), and partitioning the array in chunks of length  $s$ ,  $1 \leq s \leq N$  we can compute the partial sums using the following algorithm.

```
for y = 2 to s
    for x = y to N by s
        M(x) = M(x) + M(x - 1)

l = [N/s] s
for y = s + 1 to l by s
    for x = y to y + s - 1
        M(x) = M(x) + M(y - 1)

for x = l + 1 to N
    M(x) = M(x) + M(l)
```

Here, the first loop nest solves  $\lceil N/s \rceil$  independent partial summation problems. The inner loop of the first loop nest performs the same operation simultaneously on all chunks. Because of the increment  $s$ , this inner loop is not recursive, therefore vectorizable. After executing the first loop nest, the original partial summation problem is only solved for the first chunk  $M(x), x = 1, \dots, s$ . The second loop nest takes care of the other chunks in turn by adding  $M(y - 1)$ , the end-point-value of the previous chunk, to all values in the current chunk. Here, the inner loop is vectorizable, since trivially  $y - 1 < y, \dots, y + s - 1$ . After executing the second loop nest, all chunks except for possibly the last one also contain the correct values for the original partial summation problem. Finally, the last loop handles the last chunk in case  $s$  does not evenly divide  $N$ .

Using this algorithm, also the partial summation is vectorizable albeit at the price of doing twice as many additions but, at a performance gain of an order of magnitude, because it now readily vectorizes. On a Cray C90 we measured a speed-up factor of 5–9 depending on the values of  $N$  and  $s$ .

We still have not chosen the chunk size  $s$ . In order to perform both loop nests at vector speed,  $s$  should be chosen such that the iteration counts of the respective inner loops (being  $N/s$  and  $s$ ) are not too small. Moreover,  $s$ , being the increment of the first inner loop, should not be a multiple of the number of memory banks. The latter would cause memory bank conflicts resulting in a measured performance degradation by a factor 4 in CPU time on a Cray C90. Finally, the choice of  $s$  also depends on other optimization techniques in the actual implementation.

Had our prototype not been written using Fortran `INTEGERS`, we probably would have opted for Cray's `SCILIB` (`SCientific LIBrary`) routine `RECPS`. Our implementation and `RECPS` perform comparably. In Section 5 we can not do without the partial summation algorithm described above, since there is no `SCILIB` routine with the same functionality for `INTEGERS`.

### 3.5 Gathering the statistics

Having a completely vectorized algorithm for the systematic computation of  $M(x)$  we are still not completely done. One not entirely minor point remains: we want to study the local extremes of  $M(x)/\sqrt{x}$ . Clearly, we do not want—neither have to—compute  $\sqrt{x}$  for all  $x$ .  $M(x)/\sqrt{x}$  can only reach a new extreme value if  $M(x)$  does. Searching for new extremes can only be done at vector speed if the number of extremes is small with respect to the number of elements we are considering. If in the interval we are investigating Mertens' conjecture

$$-\sqrt{x} < M(x) < \sqrt{x}, \quad 1 < x \leq N,$$

holds, it guarantees at most  $\sqrt{N}$  local maxima and minima. On the other hand, if Mertens' conjecture would not hold in the interval we are investigating, we would find the smallest argument value  $x$  giving a counterexample for Mertens' conjecture.

We search  $M(x)$  for new extremes in either direction using the highly efficient Cray `SCILIB` routines `ISRCHFGT` and `ISRCHFLT`.

We refrain from describing the actual bookkeeping process, since bookkeeping of the extremes gets rather complicated by the sieve partitioning, our search for extremes in two directions (positive/negative), and minimization of the printed output.

### 3.6 Comparison to Neubauer's algorithm and Dress' version

Te Riele drew our attention to the work of Neubauer [12], who used a similar algorithm for computing  $M(x)$ ,  $x = 1, \dots, 10^8$ . Neubauer also had to partition his sieve interval. However, he used three sieve arrays. His algorithm [12, p. 2] reads as follows: for  $n = 0, 1, \dots$

$$1000n < m \leq 1000(n + 1)$$

$$p_i^2 \mid m \Rightarrow \mu(m) = 0$$

$$\varrho(m) = \sum_{p_i \mid m} \log p_i \quad \nu(m) = \sum_{p_i \mid m} 1$$

Neubauer builds up  $\varrho(m)$  to check whether there is a prime factor  $p > \sqrt{N}$  and he counts the number of different prime factors in  $\nu(m)$ . Neubauer does not use multiplications, nor divisions. However, he must take care of precision because of the inherently inexact  $\log p_i$  values.

Recently, Dress used a variant of Neubauer's algorithm using only two sieve arrays ( $a$  and  $\mu$  in [3, Algorithm 2]) and—at least in the description—a division step.

Our algorithm only uses one sieve array, computing  $\mu(n)$  in-place. Moreover, on current vector computers a vector-multiplication is just as expensive as a vector-addition. Because of the overhead involved in partitioning the sieve interval, there is a certain trade-off between memory usage and CPU usage. Using only one sieve array, and, of course, the unavoidable prime table, it is possible to use the available memory as efficient as possible. Moreover, we have added a small prime variation.

#### 4 DIRICHLET'S DIVISOR PROBLEM

We consider  $D(x)$ , the first summatory function of  $\tau(n)$ ,

$$D(x) = \sum_{n \leq x} \tau(n),$$

where  $\tau(n)$  denotes the number of divisors of  $n$ . Dirichlet [2] proved that

$$D(x) = x \log x + (2\gamma - 1)x + E(x),$$

where  $\gamma$  is Euler's constant, and  $E(x) = \mathcal{O}(\sqrt{x})$ . This may be considered as a lattice point problem, counting the number of lattice points in the first quadrant between the axes and the hyperbola  $qd = x$ , including those on the hyperbola

$$D(x) = \sum_{n \leq x} \sum_{d|n} 1 = \sum_{\substack{q,d \\ qd \leq x}} 1.$$

Compare FIGURE 1. An unsolved problem in analytic number theory is the estimation of the order of the error term  $E(x)$ . TABLE 1 shows the historical development of Dirichlet's divisor problem. For a more complete table, further references, and much more about lattice point problems in general, we refer to [5, 7].

#### 5 A VECTORIZED ALGORITHM FOR DIRICHLET'S DIVISOR PROBLEM

Given the unique prime factorization of  $n$

$$n = \prod_{i=1}^k p_i^{e_i}$$

we have the following formula for  $\tau(n)$

$$\tau(n) = \prod_{i=1}^k (e_i + 1).$$

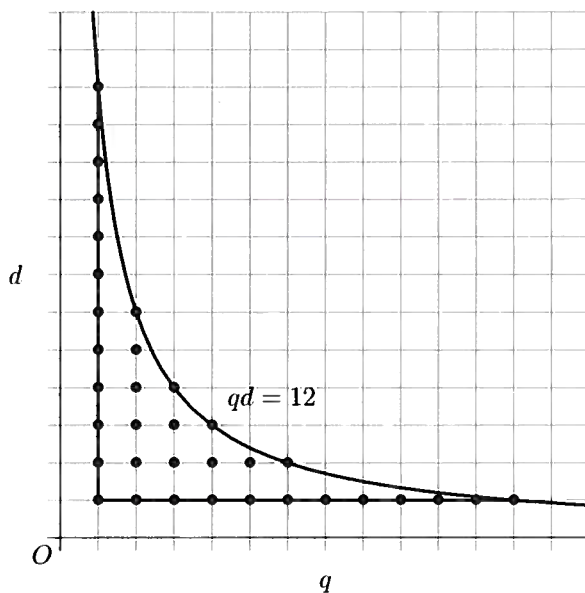


FIGURE 1. Dirichlet's divisor problem

Using two sieve arrays instead of one, and exploiting the above mentioned formula we can sieve  $\tau(n)$  similarly as  $\mu(n)$ . The following algorithm computes the number of divisors function  $\tau(n)$  for  $n = 1, \dots, N$ .

```

for  $n = 1$  to  $N$ 
   $I(n) = 1$ 
   $\tau(n) = 1$ 
for all  $p \leq \sqrt{N}$ 
  for all  $n, p \mid n$ 
     $I(n) = p \cdot I(n)$ 
     $\tau(n) = 2 \cdot \tau(n)$ 
for  $e = 2$  to  $\lfloor \log_2 N \rfloor$ 
  for all  $p \leq \sqrt[e]{N}$ 
    for all  $n, p^e \mid n$ 
       $I(n) = p \cdot I(n)$ 
       $\tau(n) = (e + 1) \cdot \frac{\tau(n)}{e}$ 
for  $n = 1$  to  $N$ 
  if  $I(n) \neq n$  then
     $\tau(n) = 2 \cdot \tau(n)$ 

```

We need two sieve arrays because keeping track of a parity as for  $\mu(n)$  does not suffice. In the  $I$ -array we multiply all prime factors encountered during sieving.

TABLE 1. The order of the error term in  $D(x)$

	year	$E(x)$	
Dirichlet	1849	$\mathcal{O}(x^{1/2})$	
Voronoi	1903	$\mathcal{O}(x^{1/3} \log x)$	
Van der Corput	1922	$\mathcal{O}(x^{33/100})$	
Kolesnik	1969	$\mathcal{O}(x^{(12/37)+\varepsilon})$	$\forall \varepsilon > 0$
Iwaniec & Mozzochi [6]	1988	$\mathcal{O}(x^{7/22})$	
Van de Lune and Wattel conjecture [9]	1990	$\mathcal{O}(x^{1/4} \log x)$	
Hardy and Landau	1915	$\Omega_{\pm}(x^{1/4})$	

This way only a single prime factor  $p > \sqrt{N}$  can remain which is taken care of by the last loop nest. In the  $\tau$ -array we maintain the number of divisors using the above mentioned formula: when sieving with a prime factor we multiply  $\tau(n)$  with 2 (since  $e = 1$ ); sieving with a prime square we divide the current value of  $\tau(n)$  by 2 and multiply with 3; when sieving with higher prime powers, exponent  $e$ , say, we divide by  $e$  and multiply with  $e + 1$ .

Similarly as for the  $\mu(n)$  we can use a small prime variation by creating patterns for both the  $I$ -array and the  $\tau$ -array.

The partial summation, and gathering of the statistics can all be performed analogous to the procedures for  $M(x)/\sqrt{x}$ .

For an actual implementation on the Cray C90 one should use an INTEGER  $\tau$ -array, because of the very fast but inexact floating point division (resulting e.g. in  $3.0/3.0 \neq 1$ ).

## 6 NUMERICAL RESULTS FOR $M(x)/\sqrt{x}$

We verified the results of Neubauer [12], Cohen & Dress [1], and Dress [3]. Furthermore, we established the bounds  $-0.513 < M(x)/\sqrt{x} < 0.571$ , valid for  $200 < x \leq 10^{13}$ . See TABLE 2 for some selected values of  $M(x)$  and  $M(x)/\sqrt{x}$  for  $x = 1, \dots, 10^{13}$ .

The computation of Cohen and Dress [1] in 1979 up to  $7.8 \cdot 10^9$  took a week on a TI980B minicomputer. The computation of Dress [3] up to  $10^{12}$  in 1992 took 4000 hours on three Sun SPARCstations 2.

Our results were all obtained using one processor. A test run up to  $10^{10}$  of our prototype implementation took 32 minutes on a Cray Y-MP. The same run of our final implementation took 9 minutes on a Cray C90. The speed-up was due to the faster machine and the improved implementation. The verification of [3] (up to  $10^{12}$ ) took some 17 hours on a Cray C90. Finally, the computation up to  $10^{13}$  took a little less than 200 hours on a Cray C90.

TABLE 2.  $M(x)$  and  $M(x)/\sqrt{x}$  for some selected<sup>a</sup>  $x < 10^{13}$

$x$	$M(x)$	$\frac{M(x)}{\sqrt{x}}$	$x$	$M(x)$	$\frac{M(x)}{\sqrt{x}}$
30,095,923	-1,448	-0.264	9,826,066,363	-31,207	-0.315
30,919,091	-2,573	-0.463	15,578,669,387	-51,116	-0.410
34,750,986	1,420	0.241	18,835,808,417	50,287	0.366
61,913,863	2,845	0.362	19,890,188,718	60,442	0.429
70,497,103	-2,574	-0.307	22,745,271,553	-51,117	-0.339
76,015,339	-3,448	-0.395	38,066,335,279	-81,220	-0.416
90,702,782	2,846	0.299	48,201,938,615	60,443	0.275
92,418,127	3,290	0.342	48,638,777,062	76,946	0.349
109,528,655	-3,449	-0.330	56,794,153,135	-81,221	-0.341
110,103,729	-4,610	-0.439	101,246,135,617	-129,332	-0.406
141,244,329	3,291	0.277	106,512,264,731	76,947	0.236
152,353,222	4,279	0.347	108,924,543,546	170,358	0.516
179,545,614	-4,611	-0.344	148,449,169,741	-129,333	-0.336
179,919,749	-6,226	-0.464	217,309,283,735	-190,936	-0.410
216,794,087	4,280	0.291	295,766,642,409	170,359	0.313
360,718,458	6,695	0.353	297,193,839,495	207,478	0.381
455,297,339	-6,227	-0.292	325,813,026,298	-190,937	-0.335
456,877,618	-8,565	-0.401	330,138,494,149	-271,317	-0.472
514,440,542	6,696	0.295	330,486,258,610 <sup>c</sup>	-287,440	-0.500
903,087,703	10,246	0.341	330,508,686,218 <sup>c</sup>	-294,816	-0.513
1,029,223,105	-8,566	-0.267	400,005,203,086	207,479	0.328
1,109,331,447	-15,335	-0.460	661,066,575,037	331,302	0.407
1,228,644,631	10,247	0.292	1,246,597,697,210	-294,817	-0.264
2,218,670,635	15,182	0.322	1,440,355,022,306	-368,527	-0.307
2,586,387,614	-15,336	-0.302	1,600,597,184,945	331,303	0.262
2,597,217,086	-17,334	-0.340	1,653,435,193,541	546,666	0.425
3,061,169,989	15,183	0.274	2,008,701,330,005	-368,528	-0.260
3,314,385,678	21,777	0.378	2,087,416,003,490	-625,681	-0.433
3,724,183,273	-17,335	-0.284	2,319,251,110,865	546,667	0.359
3,773,166,681	-25,071	-0.408	2,343,412,610,499	594,442	0.388
5,439,294,226	21,778	0.295	3,268,855,616,262	-625,682	-0.346
5,439,294,781	21,791	0.295	3,270,926,424,607	-635,558	-0.351
6,600,456,626	-25,072	-0.309	3,754,810,967,055	594,443	0.307
6,631,245,058	-31,206	-0.383	4,098,484,181,477	780,932	0.386
7,544,459,107	21,792	0.251	5,184,088,665,413	-635,559	-0.279
7,660,684,541	38,317	0.438	5,197,159,385,733	-689,688	-0.303
7,725,038,629 <sup>b</sup>	43,947	0.500	6,202,507,744,370	780,933	0.314
7,766,842,813 <sup>b</sup>	50,286	0.571	9,784,334,467,058	889,948	0.285

<sup>a</sup>A listed  $M(x)$ -value guarantees the corresponding  $x$  to be the smallest argument value for which  $M(x)$  assumes this value. Consecutive  $M(x)$ -column-entries of the same sign guarantees absence of new extremal  $M(x)$ -values of the opposite sign in between. A framed  $M(x)/\sqrt{x}$  value guarantees the corresponding  $x$  to be the smallest argument value greater than 200 for which  $M(x)/\sqrt{x}$  assumes this value.

<sup>b</sup>This verifies a result of Cohen and Dress [1].

<sup>c</sup>This verifies a result of Dress [3].



## 7 CONCLUDING REMARKS

We showed two vectorized algorithms: one for fast systematic computations on Mertens' conjecture, and one for fast systematic computations on Dirichlet's divisor problem. In an update of this paper we will extend Section 6 with numerical results for Dirichlet's divisor problem.

The algorithms we described are generalizable to arbitrary arithmetical functions  $f : \mathbf{N} \rightarrow \mathbf{Z}$  as long as we have a fairly simple relation between  $f(p^e q)$  and  $f(p^{e-1} q)$ , where  $e, p, q \in \mathbf{N}$ ,  $p$  prime,  $p \nmid q$ . For example  $\tau(p^e q) = \frac{e+1}{e} \tau(p^{e-1} q)$ . In particular, we have devised similar algorithms for Gauß' lattice point problem and amicable numbers, to name just two [8].

## ACKNOWLEDGEMENTS

This work was sponsored by the Stichting Nationale Computerfaciliteiten (National Computing Facilities Foundation, NCF) for the use of supercomputer facilities, with financial support from the Nederlandse Organisatie voor Wetenschappelijk Onderzoek (Netherlands Organization for Scientific Research, NWO).

## REFERENCES

1. H. Cohen. Arithmétique et informatique. *Astérisque*, 61:57–61, 1979.
2. G.L. Dirichlet. Über die Bestimmung der mittleren Werthe in der Zahlentheorie. *Abhandlungen der Königlich Preussischen Akademie der Wissenschaften*, pages 69–83, 1849.
3. F. Dress. Fonction sommatoire de la fonction de Möbius, 1. Majorations expérimentales. *Experiment. Math.*, 2(2):89–98, 1993.
4. F. Dress and M. El Marraki. Fonction sommatoire de la fonction de Möbius, 2. Majorations asymptotiques élémentaires. *Experiment. Math.*, 2(2):99–112, 1993.
5. F. Fricker. *Einführung in die Gitterpunktlehre*. Number 73 in LMW/MA. Birkhäuser Verlag, 1982.
6. H. Iwaniec and C.J. Mozzochi. On the divisor and circle problems. *J. Number Theory*, 29(1):60–93, 1988.
7. E. Krätzel. *Lattice Points*. Number 22 in Mathematische Monographien. VEB Deutscher Verlag der Wissenschaften, 1988.
8. W.M. Lioen and J. van de Lune. Vectorized algorithms for certain arithmetical functions. Work in progress, 1995.
9. J. van de Lune and E. Wattel. Systematic computations on Dirichlet's divisor problem. To appear.
10. H.G. Mairson. Some new upper bounds on the generation of prime numbers. *Comm. ACM*, 20(9):664–669, September 1977.
11. F. Mertens. Über eine zahlentheoretische Function. *Sitzungsber. Akad. Wiss. Wien*, 106(IIa):761–830, 1897.
12. G. Neubauer. Eine empirische Untersuchung zur Mertensschen Funktion. *Numer. Math.*, 5:1–13, 1963.

13. A.M. Odlyzko and H.J.J. te Riele. Disproof of the Mertens conjecture. *J. Reine Angew. Math.*, 357:138–160, 1985.
14. C. Pomerance, J.W. Smith, and R. Tuler. A pipeline architecture for factoring large integers with the quadratic sieve algorithm. *SIAM J. Comput.*, 17(2):387–403, April 1988.
15. P. Pritchard. Linear prime-number sieves: A family tree. *Sci. Comput. Programming*, 9:17–35, 1987.
16. H.J.J. te Riele. On the history of the function  $M(x)/\sqrt{x}$  since Stieltjes. In G. van Dijk, editor, *Thomas Jan Stieltjes – Collected Papers*, volume 1, pages 69–79. Springer-Verlag, 1993.
17. R.D. von Sterneck. Neue empirische Daten über die zahlentheoretische Funktion  $\sigma(n)$ . In E.W. Hobson and A.E.H. Love, editors, *Proc. of the fifth International Congress of Mathematicians (Cambridge, 22–28 August 1912)*, volume 1, pages 341–343. Cambridge, 1913.
18. R. Tijdeman. Private communication, April 2, 1993.

# Actions on the Hilbert cube

*To Cor Baayen, at the occasion of his retirement.*

Jan van Mill

*Faculteit Wiskunde en Informatica, Vrije Universiteit  
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands*

*(email: vanmill@cs.vu.nl)*

We provide a negative answer to Problem 933 in the "Open Problems in Topology Book".

*Key words & phrases:* Lie group, action, conjugate

*AMS Subject Classification:* 57N50

## 1 INTRODUCTION

Let  $Q$  denote the Hilbert cube  $\prod_{i=1}^{\infty} [-1, 1]_i$ . In the "Open Problems in Topology Book", WEST [2] asks the following (Problem #933):

*Let the compact Lie group  $G$  act semifreely on  $Q$  in two ways such that their fixed point sets are identical. If the orbit spaces are ANR's, are the actions conjugate?*

The aim of this note is to present a counterexample to this problem. For all undefined notions we refer to [1].

## 2 THE EXAMPLE

Let  $G$  be a group and let  $\pi: G \times X \rightarrow X$  be an action from  $G$  on  $X$ . Define  $\mathcal{F}ix(G) = \{x \in X : (\forall g \in G)(\pi(g, x) = x)\}$ . It is clear that  $\mathcal{F}ix(G)$  is a closed subset of  $X$ : it is called the *fixed-point set* of  $G$ . The action  $\pi$  is called *semifree* if it is free off  $\mathcal{F}ix(G)$ , i.e., if  $x \in X \setminus \mathcal{F}ix(G)$  and  $\pi(g, x) = x$  for some  $g \in G$  then  $g$  is the identity element of  $G$ . The space of orbits of the action  $\pi$  will be denoted by  $X/G$ . Let  $I$  denote the interval  $[0, 1]$ .

Let  $G$  denote the compact Lie group  $\mathbb{T} \times \mathbb{Z}_2$ , where  $\mathbb{T}$  denotes the circle group. We identify  $\mathbb{Z}_2$  and the subgroup  $\{-1, 1\}$  of  $\mathbb{T}$ . In addition,  $D$  denotes  $\{z \in \mathbb{C} : |z| \leq 1\}$ . We let  $G$  act on  $D \times D$  in the obvious way:

$$((g, \varepsilon), (x, y)) \mapsto (g \cdot x, \varepsilon \cdot y) \quad (g \in \mathbb{T}, \varepsilon \in \{-1, 1\}, x, y \in D),$$

where “ $\cdot$ ” means complex multiplication. Observe that this action is semifree, and that its fixed-point set contains the point  $(0, 0)$  only. Also, observe that  $(D \times D)/G \approx \mathbb{I} \times D$ .

**LEMMA 2.1** *Let  $H$  denote either  $G$  or  $\mathbb{T}$ . There is a semifree action of  $H$  on  $Q \times \mathbb{I}$  having  $Q \times \{0\}$  as its fixed-point set. Moreover,  $(Q \times \mathbb{I})/G$  and  $Q$  are homeomorphic.*

**PROOF.** We will only prove the lemma for  $G$  since the proof for  $\mathbb{T}$  is entirely similar. We first let  $G$  act on  $X = D \times D \times Q$  as follows:

$$((g, \varepsilon), (x, y, z)) \mapsto (g \cdot x, \varepsilon \cdot y, z) \quad (g \in \mathbb{T}, \varepsilon \in \{-1, 1\}, x, y \in D, z \in Q).$$

This action is semifree and its fixed-point set is equal to  $\{(0, 0)\} \times Q$ . Also observe that  $X/G \approx \mathbb{I} \times D \times Q$ .

We now let  $G$  act coordinatewise on the infinite product  $X^\infty$ . This action is again semifree, having the diagonal  $\Delta$  of  $\{(0, 0)\} \times Q$  in  $X^\infty$  as its fixed-point set. Also,  $X^\infty/G$  is homeomorphic to  $(\mathbb{I} \times D \times Q)^\infty \approx Q$ . Since  $\Delta$  projects onto a proper subset of  $X$  in every coordinate direction of  $X^\infty$ , it is a  $Z$ -set. Since  $X^\infty \approx Q$  there consequently is a homeomorphism of pairs  $(X^\infty, \Delta) \rightarrow (Q \times \mathbb{I}, Q \times \{0\})$ . We are done.

We will now describe two actions of  $G$  on  $Q \times [-1, 1]$ . By Lemma 2.1 there is a semifree action  $\alpha_r: \mathbb{T} \times Q \times \mathbb{I} \rightarrow Q \times \mathbb{I}$  having  $Q \times \{0\}$  as its fixed point set, while moreover  $Q \times \mathbb{I}/G \approx Q$ . We let  $\mathbb{T}$  act on  $Q \times [-1, 0]$  as follows:

$$(z, (q, t)) \mapsto (\bar{q}, s) \quad \text{iff} \quad \alpha_r(z, (q, -t)) = (\bar{q}, -s).$$

We will denote this action by  $\alpha_l$ . So  $\alpha = \alpha_l \cup \alpha_r$  is an action of  $\mathbb{T}$  onto  $Q \times [-1, 1]$ , having  $Q \times \{0\}$  as its fixed-point set. Now define  $\bar{\alpha}: G \times (Q \times [-1, 1]) \rightarrow Q \times [-1, 1]$  as follows:

$$\bar{\alpha}((z, \varepsilon), (q, t)) = \begin{cases} \alpha(z, (q, t)), & (\varepsilon = 1), \\ \alpha(z, (q, -t)), & (\varepsilon = -1). \end{cases}$$

Then  $\bar{\alpha}$  is a semifree action of  $G$  onto  $Q \times [-1, 1]$  having  $Q \times \{0\}$  as its fixed-point set, while moreover  $(Q \times [-1, 1])/\bar{\alpha} \approx Q$ . Observe the following triviality.

**LEMMA 2.2** *If  $A \subseteq Q \times [-1, 1]$  is  $\bar{\alpha}$ -invariant such that  $A$  is not contained in  $Q \times \{0\}$ , then  $A$  intersects  $Q \times (0, 1]$  as well as  $Q \times [-1, 0)$ .*

We will now describe the second action on  $Q \times [-1, 1]$ . By Lemma 2.1 there is a semifree action  $\beta_r: G \times Q \times \mathbb{I} \rightarrow Q \times \mathbb{I}$  having  $Q \times \{0\}$  as its fixed point set, while moreover  $Q \times \mathbb{I}/G \approx Q$ . Construct  $\beta_l$  from  $\beta_r$  in the same way we constructed  $\alpha_l$  from  $\alpha_r$ . Then  $\beta = \beta_l \cup \beta_r$  is a semifree action from  $G$  onto  $Q \times [-1, 1]$  having  $Q \times \{0\}$  as its fixed-point set. Moreover,  $(Q \times \mathbb{I})/\beta$  is the union of two Hilbert cubes, meeting in a third Hilbert cube, hence is an **AR**. (It can be shown that  $(Q \times \mathbb{I})/\beta \approx Q$ .)

Now assume that the two axioms  $\bar{\alpha}$  and  $\beta$  are conjugate. Let  $\tau: Q \times [-1, 1] \rightarrow Q \times [-1, 1]$  be a homeomorphism such that for every  $g \in G$ ,  $\beta(g) = \tau^{-1} \circ \bar{\alpha}(g) \circ \tau$ . Then  $\tau(Q \times (0, 1])$  is a connected  $\bar{\alpha}$ -invariant subset of  $Q \times [-1, 1]$  which misses  $Q \times \{0\}$ . This contradicts Lemma 2.2.

#### REFERENCES

1. J. van Mill. *Infinite-Dimensional Topology: prerequisites and introduction*. North-Holland Publishing Company, Amsterdam, 1989.
2. J. E. West. Open Problems in Infinite Dimensional Topology. In J. van Mill and G. M. Reed, editors, *Open Problems in Topology*, pages 523–597, North-Holland Publishing Company, Amsterdam, 1990.



# The Expansion Theorem for Median Graphs

Henry Martyn Mulder

*To Cor Baayen, at the occasion of his retirement*

## 1. Introduction

This paper deals with the adventures of the expansion theorem for median graphs [Nu 78, Mu 80b]. It was the first theorem I ever proved (on a walk during the Fifth Hungarian Combinatorial Conference in Keszthély, Hungary, I was 'struck' by the idea ultimately leading to this theorem), and it was the starting point for my Ph. D. thesis written under the inspiring guidance of Cor Baayen. Median graphs existed already in the literature [Av 61, Ne 71], but they were independently introduced by LEX SCHIJVER and me [MS 79] in the context of a problem in finite topology posed by JAN VAN MILL [vM 77] early 1976. All three of us were at that time Ph. D. students of Cor Baayen.

Loosely speaking the idea of expansion is the following. Let  $G$  be covered by a number of subgraphs, which two by two intersect in the same subgraph  $G_0$  of  $G$ . Now we take disjoint copies of the covering subgraphs and join the respective copies of  $G_0$  in these subgraphs by new edges.

By imposing conditions on the covering subgraphs and on how to insert the new edges we get specific instances of expansion. Some types of expansion may not be sensible to study, but others seem to be quite promising in producing interesting problems and results. In [Mu 90] a 'masterplan' was formulated for studying various expansion problems.

To show how fruitful this approach can be, we discuss a number of results on median graphs. These all have elegant and straightforward proofs using a specific instance of expansion, by which median graphs can be characterized. A median graph is a graph such that, for any triple of vertices  $u, v, w$ , there exists a unique vertex minimizing the sum of the distances to  $u$ ,  $v$  and  $w$ .

## 2. Median graphs and expansion

In this section, we will give some results and introduce terminology found in [Mu 78, Mu 80b, Mu 90, MMR 94]. All graphs considered in this paper will be finite, and we use the standard notation  $G=(V,E)$  to denote a graph with vertex set  $V$  and edge set  $E$ . We will often simply write only  $G$  and leave

$V$  and  $E$  understood. Also, we will not distinguish between a subset  $W$  of  $V$  and the subgraph induced by  $W$ . In a connected graph, the *distance*  $d(x,y)$  between two vertices  $x$  and  $y$  is the length of a shortest  $x,y$ -path, or an  $x,y$ -*geodesic*. The star of our show is the *median graph*  $G$ : a connected graph such that for every three vertices  $x,y,z$ , of  $G$ , there is a unique vertex  $w$  on a geodesic between each pair of  $x,y,z$ . This vertex  $w$  is called the *median* of the triple  $x,y,z$ . The *interval* between the vertices  $x$  and  $y$  is the set  $I(x,y)$  of all vertices on  $x,y$ -geodesics, i.e.,

$$I(x,y) = \{w \in V : d(x,w) + d(w,y) = d(x,y)\}.$$

The *interval function*  $I$  of a graph  $G$  was extensively studied in [Mu 80b]. It is an easy observation that a graph  $G$  is a median graph if and only if  $|I(x,y) \cap I(y,z)| = 1$  for all vertices  $x,y,z$  of  $G$ . Median graphs were first studied in 1961 by AVANN [Av 61], and independently introduced by NEBESKY [Ne 71] and MULDER and SCHRIJVER [MS 79]. Trees are the simplest examples of median graphs. Another prime example is the  $n$ -cube  $Q_n$ . Recall that  $Q_n$  has  $\{0,1\}^n$  as vertex set, and two vertices are adjacent whenever they differ in exactly one place. For three vertices  $x = x_1x_2\dots x_n, y = y_1y_2\dots y_n, z = z_1z_2\dots z_n$  of  $Q_n$  the median  $w = w_1w_2\dots w_n$  of  $x,y,z$  is determined by the majority rule:  $w_i = \delta$  if  $\delta$  occurs at least twice among  $x_i, y_i, z_i$ , for  $i = 1, \dots, n$ . Other examples of median graphs are the grids and the covering graphs of distributive lattices. It is also an easy observation that median graphs are bipartite, for if  $x_0\dots x_kx_{k+1}\dots x_{2k}x_0$  is a shortest cycle of odd length, then  $x_0, x_k, x_{k+1}$  would have  $x_k$  and  $x_{k+1}$  as two distinct medians. The smallest bipartite graph that is not a median graph is  $K_{2,3}$ : the profile consisting of three independent vertices has two medians.

A set  $W$  of vertices of a graph  $G$  is *convex* if  $I(x,y) \subseteq W$  for every  $x,y \in W$ , and a *convex subgraph* of  $G$  is a subgraph induced by a convex set of vertices of  $G$ . Clearly, a convex subgraph of a connected graph is also connected. Moreover, the intersection of convex sets (subgraphs) is convex. The *convex hull*  $Con(U)$  of a set of vertices  $U$  is the intersection of all the convex sets containing  $U$ . It was proved in [Mu 80b] that intervals in median graphs are convex, so that  $Con(\{x,y\}) = I(x,y)$ . Also, in median graphs, convex sets can be viewed in another useful way through the notion of a gate. For  $W \subseteq V$  and  $x \in V$ , the vertex  $z \in W$  is a *gate* for  $x$  in  $W$  if  $z \in I(x,w)$  for all  $w \in W$ . Note that a vertex  $x$  has at most one gate in any set  $W$ , and if  $x$  has a gate  $z$  in  $W$ , then  $z$  is the unique nearest vertex to  $x$  in  $W$ . The set  $W$  is *gated* if every vertex has a gate in  $W$  and a *gated subgraph* is a subgraph induced by a gated set. It



is not difficult to see that in any graph, a gated set is convex and that in a median graph a set is gated if and only if it is convex. (This last fact follows immediately from results in [Mu 80b].)

Recall that for two graphs  $G_1=(V_1,E_1)$  and  $G_2=(V_2,E_2)$ , the *union*  $G_1\cup G_2$  is the graph with vertex set  $V_1\cup V_2$  and edge set  $E_1\cup E_2$ , and the *intersection*  $G_1\cap G_2$  is the graph with vertex set  $V_1\cap V_2$  and edge set  $E_1\cap E_2$ . We write  $G_1\cap G_2=\emptyset$  ( $\neq\emptyset$ ) when  $V_1\cap V_2=\emptyset$  ( $\neq\emptyset$ ). A *proper cover* of  $G$  consists of two convex subgraphs  $G_1$  and  $G_2$  of  $G$  such that  $G=G_1\cup G_2, G_1\cap G_2\neq\emptyset$ . Every graph  $G$  admits the *trivial proper cover*  $G_1, G_2$  with  $G_1=G_2=G$ . On the other hand a cycle does not have a proper cover with two proper subgraphs.

We are now able to give the definition of the operation which will help yield a characterization of median graphs. Let  $G'=(V',E')$  be properly covered by the convex subgraphs  $G_1'=(V_1',E_1')$  and  $G_2'=(V_2',E_2')$  and set  $G_0'=G_1'\cap G_2'$ . For  $i=1,2$ , let  $G_i$  be an isomorphic copy of  $G_i'$ , and let  $\lambda_i$  be an isomorphism from  $G_i'$  onto  $G_i$ . We set  $G_{0i}=\lambda_i[G_0']$  and  $\lambda_i(u')=u_i$ , for  $u'$  in  $G_0'$ . The *expansion* of  $G'$  with respect to the proper cover  $G_1', G_2'$  is the graph  $G$  obtained from the disjoint union of  $G_1$  and  $G_2$  by inserting an edge between  $u_1$  in  $G_{10}$  and  $u_2$  in  $G_{20}$ , for each  $u'$  in  $G_0'$ . Denote the set of edges between  $G_{10}$  and  $G_{20}$  by  $F_{12}$ . This is illustrated in Figure 1. We say that  $\lambda_i$  *lifts*  $G_i'$  up to  $G_i$ . For any subgraph  $H'$  of  $G'$  we abuse the notation and write  $\lambda_i[H']$  for  $\lambda_i[H'\cap G_i']$ . So  $\lambda_i$  lifts the part of  $H'$  lying in  $G_i'$  up to  $G_i$ .

This type of expansion was called a "convex expansion" in [Mu 78], [Mu 80b], and a "convex Cartesian expansion" in [Mu90] for a more general setting. We are now able to state the following fundamental result on median graphs first proved in [Mu 78] and [Mu 80b]. This result is the basis of a recent  $O(|V|^2\log|V|)$  algorithm for recognizing median graphs found in [JS].

**Theorem 1.** A graph  $G$  is a median graph if and only if  $G$  can be obtained by successive expansions from the one vertex graph  $K_1$ .

Using this theorem, trees can be obtained from  $K_1$  by restricting the expansions to those of the following type:  $G_1$  is always the whole graph  $G$  and  $G_2$  is a single vertex. Expansion with respect to such a cover amounts to adding a new vertex adjacent to the one in  $G_2$ . The  $n$ -cubes can be obtained from  $K_1$  by using only trivial proper covers. Note that  $K_{2,3}$  can not be obtained from a smaller graph by expansion with respect to a proper cover.

In order to make full use of Theorem 1 and to develop additional techniques, we give a very brief sketch of the proof. Along the way we introduce some extra terminology.

The basic ideas used for the proof of Theorem 1 are the following. Take an arbitrary edge  $v_1v_2$  in a median graph  $G$ . Let  $G_1$  be the subgraph of  $G$  induced by all vertices nearer to  $v_1$  than to  $v_2$ , and let  $G_2$  be the subgraph induced by all vertices nearer to  $v_2$  than  $v_1$ . Since  $G$  is bipartite, it follows that  $G_1, G_2$  partition  $G$ . We call such a partition a *split*. Let  $F_{12}$  be the set of edges between  $G_1$  and  $G_2$ , and let  $G_{i0}$  be the subgraph induced by the endvertices in  $G_i$  of the edges in  $F_{12}$ , for  $i=1,2$ . Then one proceeds to prove the following facts (not necessarily in this order):

- (i)  $F_{12}$  is a matching as well as a cutset (minimal disconnecting edge-set).
- (ii) The subgraphs  $G_1, G_2, G_{10}, G_{20}$  are convex subgraphs of  $G$ .
- (iii) The obvious mapping of  $G_{10}$  onto  $G_{20}$  defined by the edges in  $F_{12}(u_1 \rightarrow u_2)$ , for any edge  $u_1u_2$  in  $F_{12}$  with  $u_i$  in  $G_{i0}$ , for  $i=1,2$ ) is an isomorphism.
- (iv) For every edge  $u_1u_2$  of  $F_{12}$  with  $u_i$  in  $G_{i0}$  ( $i=1,2$ ), the subgraph  $G_1$  consists of all the vertices of  $G$  nearer to  $u_1$  than to  $u_2$ , so that  $u_1$  is the gate in  $G_1$  for  $u_2$ . A similar statement holds for  $G_2$ .

Now the *contraction*  $G'$  of  $G$  with respect to the split  $G_1, G_2$  is obtained from  $G$  by contracting the edges of  $F_{12}$ . To illustrate this in Figure 1, move from right to left. Clearly expansion and contraction are inverse operations. The *contraction map*  $\kappa$ , of  $G$  onto  $G'$ , associated with  $F_{12}$  is thus defined by  $\kappa|_{G_i} = \lambda_i^{-1}$ , for  $i=1,2$ . Finally one shows that  $G'$  is a median graph and so, by induction on the number of vertices, Theorem 1 is proved.

We present another feature of median graphs that helps in getting the right mental picture of how to operate with them in the rest of the paper. A *cutset coloring* of a connected graph is a proper colouring of the edges (adjacent edges have different colours) such that each colour class is a cutset (a minimal disconnecting edge set). Of course, most graphs will not have a cutset colouring, whereas even cycles of length at least six have more than one. If we want to cutset colour the edges of a graph, then in an induced 4-cycle  $wxyzw$ , opposite edges must have the same colour. So,  $w, z$  are on one side and  $x, y$  on the other side of the cutset colour of  $wx$ , and thus  $yz$  gets the same colour as  $wx$ . We call this the *4-cycle property* of cutset colourings. It follows from (i), (ii) and (iii) that in any cutset colouring of the median graph  $G$ , the set  $F_{12}$  must be a colour class. Using induction on the number of colours gives the next corollary [Mu 78, Mu 80b].

**Corollary 2.** A median graph is uniquely cutset colourable up to the labeling of the colours.

For a split  $G_1, G_2$ , we call the set  $F_{12}$  a *colour*, and  $G_1$  and  $G_2$  the *colourhalves* of  $F_{12}$ . Thus it follows that any colour in the cutset colouring of the median graph  $G$  defines a split into two convex colourhalves, as in the case of  $F_{12}$  with all the properties listed above. Hence the 4-cycle property, one can determine the colour class of an arbitrary edge  $xy$ . This colour class splits  $G$  into the convex subgraph of all vertices nearer to  $x$  than to  $y$  and the convex subgraph of all vertices nearer to  $y$  than to  $x$ , etc. There is yet another important feature of median graphs that we need in the sequel, and which follows from (the proof of) Theorem 1 [Mu 80b]. If we consider any two colours in the cutset colouring of the median graph  $G$ , and we contract them in any order, then we get the same median graph  $G''$ . Hence we can apply the corresponding expansions to obtain  $G$  from  $G''$  in any order. This means that in obtaining  $G$  from a median graph  $H$  by a succession of expansions, we can apply these expansions in any order. This is easily seen in the case for trees: every expansion corresponds to an edge in the tree, and it does not matter in what order we introduce the edges in forming the tree.

The basic technique that will be used in proofs found in the next section is as follows: One or more contractions on the median graph  $G$  are performed to obtain a smaller median graph  $G'$ , on which we apply the appropriate induction hypothesis. Then we perform the corresponding expansions in reverse order on  $G'$  so that we regain  $G$ . During this process, a vertex  $x$  of  $G$  is contracted to a unique vertex  $x'$  in  $G'$ . When we recover  $G$  from  $G'$  by expansions, then  $x'$  is lifted up in each expansion to the appropriate colourhalf until we regain  $x$ . The sequences of vertices and expansions that we obtain in this way from  $x'$  up to  $x$  is called the *history* of  $x$  (with respect to the expansions involved). Similarly, if  $\pi = (x_1, \dots, x_k)$  is a sequence of vertices of  $G$ , a *profile* for short, then  $\pi$  is contracted to a profile  $\pi' = (x'_1, \dots, x'_k)$  on  $G'$ , where  $x'_i$  is the contraction of  $x_i$ , for  $i = 1, \dots, k$ . We thus define the *history* of  $\pi$  in the obvious way. If  $x'$  is a vertex of  $G'$  and we lift  $x'$  up to a vertex  $x$  in an expansion of  $G'$ , then we call  $x$  a *descendant* of  $x'$ . Hence if we know which lifts are applied on  $x'$  in the expansions to regain  $G$  from  $G'$ , then we know the history of all the descendants of  $x'$ .

Having now introduced the basic techniques and results on median graphs, we will use them frequently without specific mention in the sequel.

In a sense median graphs are the appropriate common generalization of trees and hypercubes. This as well as many other results on median graphs suggest the following 'meta'conjecture.

**Metaconjecture.** Any ‘reasonable’ property shared by trees and hypercubes is shared by all median graphs.

In the rest of the paper, we use the standard notation developed above:  $G$  is a median graph with split  $G_1, G_2$  with colour  $F_{12}$ , contraction  $G'$  etcetera.

### 3. Median sets

Let  $G$  be a connected graph. A *profile* on  $G$  is  $W$  is a vertex sequence  $\pi = (v_1, v_2, \dots, v_k)$  in  $G$ . Note that multiple occurrences in  $\pi$  are allowed. The *length*  $k$  of the profile is denoted by  $|\pi|$ . A profile is *even* or *odd* depending on whether  $k$  is even or odd. The (*simultaneous*) *distance*  $D(u, \pi)$  of a vertex  $u$  to  $\pi$  is defined by

$$D(u, \pi) = \sum_{i=1}^k d(u, v_i).$$

A *median* of  $\pi$  is a vertex  $x$  minimizing the distance  $D(x, \pi)$ , and the *median set*  $M(\pi)$  of  $\pi$  consists of the medians of  $\pi$ . Since  $G$  is assumed to be connected, a median set is always non-empty. The median set of two vertices  $u, v$  is the interval  $I(u, v)$ . In general not much is known about the structure of median sets, but not so for median graphs. Clearly here every triple of vertices has a unique median. For longer profiles the situation is equally plain. After one has made the effort to develop the expansion technique, one can sit down in the armchair and let the expansions do the work. In [MMR 94] the expansion technique is exploited in its full richness to study median sets in median graphs. We present here the main results and prove one Lemma to give an idea how one could proceed to prove the theorems.

If  $\pi$  is a profile in a median graph  $G$  with split  $G_1, G_2$ , then let  $\pi_i$  be the subprofile of  $\pi$  consisting of all elements of  $\pi$  in  $G_i$ . For each subset  $W$  of  $V$ , we set  $W' = \kappa[W]$  and  $x' = \kappa(x)$ . Note that if for some  $u'$  in  $G'_0$ , both  $u_1$  and  $u_2$  are elements of  $W$ , then  $u'$  is in  $W'$  and  $|W'| < |W|$ . If  $\pi$  is a profile on  $G$ , then we have  $\pi'_i = \kappa(\pi_i)$  and  $\pi_i = \lambda_i(\pi'_i)$ , where  $\kappa$  and  $\lambda_i$  are applied componentwise.

**Lemma 3.** With the above notation, if  $\pi$  is a profile in the median graph  $G$  with  $|\pi_1| > |\pi_2|$ , then  $M(\pi')$  is contained in  $G'_1$ , and  $M(\pi)$  is contained in  $G_1$ , and  $M(\pi') = M(\pi)'$ , and  $|M(\pi)| = |M(\pi')|$ .

**Proof.** Let  $w'$  be a vertex in  $G'_2 - G'_0$  and let  $x'$  be the gate of  $w'$  in  $G'_1$ . Then we have

$$D(w', \pi'_1) = D(x', \pi'_1) + |\pi'_1| d(x', w').$$

The triangle inequality for  $d$  yields

$$D(w', \pi'_2) \geq D(x', \pi'_2) - |\pi'_2| d(x', w').$$

Hence we have

$$\begin{aligned} D(w', \pi') &= D(w', \pi'_1) + D(w', \pi'_2) \\ &\geq D(x', \pi'_1) + d(x', w') (|\pi'_1| - |\pi'_2|) \\ &> D(x', \pi'). \end{aligned}$$

So  $M(\pi')$  lies in  $G'_1$ .

Now choose a vertex  $w$  in  $G_2$  and a vertex  $v$  in  $G_1$  with  $v'$  in  $M(\pi')$ . Then we have

$$D(v, \pi) = D(v', \pi') + |\pi_2|,$$

$$D(w, \pi) = D(w', \pi') + |\pi_1|,$$

whence  $D(w, \pi) > D(v, \pi)$ . So  $M(\pi)$  lies in  $G_1$ . Finally, for each vertex  $v$  in  $G$ , we have

$$D(v, \pi) = D(v', \pi') + |\pi_2|,$$

so that  $M(\pi) = M(\pi')$ . Since  $M(\pi)$  lies in  $G_1$ , it follows that  $|M(\pi)| = |M(\pi')|$ .  $\square$

Using this Lemma, we can relate  $M(\pi)$  to the median set of  $\pi'$  in  $G'$ .

**Theorem 4.** If  $\pi$  is a profile in a median graph  $G$  with  $|\pi_1| > |\pi_2|$ , then  $M(\pi) = \lambda_1[M(\pi')]$ . Furthermore, if  $\pi$  is odd, then  $|M(\pi)| = 1$ . If  $\pi$  is even, then  $M(\pi)$  is an interval, and if  $|\pi_1| = |\pi_2|$ , then  $M(\pi) = \lambda_1[M(\pi')] \cup \lambda_2[M(\pi')]$

Using expansions, we can also relate  $M(\pi)$  to the median sets of its vertex-deleted subprofiles.

**Theorem 5.** Let  $\pi = (v_1, v_2, \dots, v_k)$  a profile in a median graph  $G$  with  $k > 1$ . If  $\pi$  is odd, then  $M(\pi) = \bigcap_i M(\pi - v_i)$ , and if  $\pi$  is even, then  $M(\pi) = \text{Con}(\bigcap_i M(\pi - v_i))$ .

For proofs the reader is referred to [MMR 94].

#### 4. Dynamic search

In [CGS 87] and [CGS 89] CHUNG, GRAHAM and SAKS considered the following intriguing problem and proved some important results.

Let  $G = (V, E)$  be a connected graph, where on each vertex some piece of information is located. A retriever is located at some vertex  $u$  of  $G$ , his *position*. A *quest* for a piece of information comes in the form of a quest for the vertex where this information is located. The retriever has two options:

- (i) to retrieve from  $u$  the information at  $v$ , which costs  $d(u, v)$ ;
- (ii) to move from  $u$  to some vertex  $v$ , which also costs  $d(u, v)$ .

If the retriever is at an *initial position*  $p_0$ , then his goal is, given a sequence of quests  $Q = q_1, q_2, \dots, q_n$  to find a sequence of positions  $P = p_0, p_1, \dots, p_n$  such that the following distance sum is minimized:

$$(*) \quad \sum_{i=1}^n d(p_{i-1}, p_i) + d(p_i, q_i).$$

We can read this sum as follows: being at  $p_{i-1}$ , the retriever first moves to  $p_i$  and then retrieves  $q_i$ , for  $i = 1, \dots, n$ .

With each *quest sequence*  $Q$  and each *position sequence*  $P$  we can associate a *caterpillar*  $R(P, Q)$  consisting of  $P$ ,  $Q$  and a  $p_{i-1}, p_i$ -geodesic and a  $p_i, q_i$ -geodesic, for  $i = 1, \dots, n$ . Note that in  $R(P, Q)$  we may have multiple occurrences of vertices as well as edges. The  $p_{i-1}, p_i$ -geodesics with  $i = 1, \dots, n$  form the *spine* of the caterpillar, the  $p_i, q_i$ -geodesics are the *legs* (note that mathematics is capable of creating new biological species). The *length*  $\ell(P, Q)$  of the caterpillar  $R(P, Q)$  is the sum of the lengths of all geodesics involved in constructing the caterpillar, and thus  $\ell(P, Q)$  equals sum (\*) above. In these terms, given a quest sequence  $Q$  and initial position  $p_0$  we want to find a *shortest* caterpillar  $R(P, Q)$ .

If the retriever being at the initial position knows all the quests in quest sequence  $Q$ , then he can always find a shortest caterpillar  $R(P, Q)$

minimizing his total costs. How to find  $P$  is another story. But if he has only partial knowledge at some position  $p_{i-1}$ , he can only optimize  $p_i$  with respect to, say, the next  $k$  quests  $q_i, q_{i+1}, \dots, q_{i+k-1}$ . When finally all quests have come in and he has completed his caterpillar  $R(P, Q)$ , then it is generally not the shortest possible caterpillar.

If at any position  $p_{i-1}$  we have only foreknowledge of the next two quests  $q_i$  and  $q_{i+1}$ , then the best thing we can do is choosing a median point of  $p_{i-1}, q_i, q_{i+1}$  as our next position  $p_i$ . This is the *median strategy*. In [CGS 87] the problem was posed and settled on which graphs the median strategy, with always foreknowledge of the next two quests at each position, will produce a shortest caterpillar for each initial position  $p_0$  and each quest sequence  $Q$ , cf. [Wr 87].

**Theorem 6.** Let  $G$  be a connected graph. The median strategy with foreknowledge of the next two quests at each position produces a shortest caterpillar for each initial position  $p_0$  and each quest sequence  $Q$  if and only if  $G$  is a median graph.

If the median strategy is optimal, then CHUNG, GRAHAM and SAKS proceed in the following way. Assume that there are vertices  $u, v, w$  having two distinct median points. Choose such a triple with  $d(u, v) + d(v, w) + d(w, u)$  as small as possible. Now, with initial position  $u$ , by choosing quest sequences of length at most 6 of the type  $u, u, v, w, q, q$  and varying  $q$ , a contradiction can be derived. For full details of this proof the reader is referred to [CGS 87]. To prove the converse they make use of BANDELT's theorem [Ba 84] that the median graphs are precisely the retracts of hypercubes (see the next subsection). Here we give an alternative proof for the 'if part' using our expansion approach.

**Proof of the 'if part' of Theorem 6.** We use induction on the number of expansions, so let  $F, G_1, G_2, G', \pi', W'$  etcetera be as above. Let  $P$  be the position sequence obtained via the median strategy with respect to initial position  $p_0$  and quest sequence  $Q$ . Note that, because of unicity of medians,  $P$  is uniquely determined.

Assume that there is a position sequence  $T$  with  $\ell(T, Q) < \ell(P, Q)$ . Note that  $P'$  is the position sequence obtained via the median strategy in  $G'$  with respect to  $p'_0$  and  $Q'$ . By induction hypothesis, we know that  $\ell(T', Q') \geq \ell(P', Q')$ . Note that, for any caterpillar  $R(S, Q)$  in  $G$ , it follows from the expansion procedure that

$$\ell(S,Q) = \ell(S',Q') + \alpha(S,Q),$$

where  $\alpha(S,Q)$  is the number of edges from  $F$  lying on  $R(S,Q)$ .

Without loss of generality we may assume that  $p_0$  lies in  $G_1$ . Put  $q_0 = p_0$ . The spine of  $R(P,Q)$  starts in  $G_1$ . Beginning in  $p_0$  we walk along the spine of  $R(P,Q)$  and check where the caterpillar crosses the cut  $F$ :

- if  $p_{i-1}, q_{i-1}, q_{i+1}$  lie in  $G_1$  and  $q_i$  lies in  $G_2$ , then the crossing is in the  $p_i, q_i$ -leg and the spine remains in  $G_1$ ,
- if  $p_{i-1}, q_{i-1}$  lie in  $G_1$  and  $q_i, q_{i+1}$  lie in  $G_2$ , then the crossing is in the spine between  $p_{i-1}$  and  $p_i$ ; now we exchange the roles of  $G_1$  and  $G_2$  and proceed along the spine.

Note that a crossing only occurs if  $Q$  crosses  $F$ , but not necessarily, for in the first case above  $Q$  crosses  $F$  twice and the caterpillar crosses  $F$  only once. Each caterpillar must cross  $F$  at least once in the above situations. So, for any position sequence  $S$ , we have  $\alpha(S,Q) \geq \alpha(P,Q)$ .

Combined with  $\ell(T',Q') \geq \ell(P',Q')$  we get  $\ell(T,Q) \geq \ell(P,Q)$ , contradicting our assumption that  $R(T,Q)$  was a shorter caterpillar than  $R(P,Q)$ .  $\square$

## 5. Retracts of hypercubes

A *retract* of a graph  $G$  is an isometric subgraph  $H$  of  $G$  such that there is a distance decreasing map of  $G$  onto  $H$ , which restricted to  $H$  is the identity. BANDELIT [Ba 84] proved that the median graphs are precisely the retracts of hypercubes (for further references on retracts see [Ba 84] or [CGS 89]). This result also can be proved using expansions. We only sketch that here using the notation introduced above.

We define an *extremal colour* of a median graph  $G$  to be a colour  $F$  such that, say,  $G_1 = G_0$ . Then  $G_1$  is an *extremal subgraph*. In a tree the end vertices are the extremal subgraphs, and in an  $n$ -dimensional hypercube ( $n$ -cube, for short) the  $(n-1)$ -subcubes are the extremal subgraphs. Note that the edges on a geodesic in a median graph all have different colours.

**Lemma 7.** Let  $G$  be a median graph with split  $G_1, G_2$ . Then  $G_1$  as well as  $G_2$  contain an extremal subgraph.

**Proof.** Assume that  $G_1 \neq G_{10}$ . Let  $x$  be a vertex in  $G_1 - G_{10}$  adjacent to a vertex  $y$  in  $G_{10}$ , and let  $z$  be the neighbour of  $y$  in  $G_{20}$ . Recall that  $z$  is the gate for



$y$  in  $G_2$ . Let  $A$  be the colour of  $xy$ , and let  $F$  be the colour of  $yz$  (i.e. the colour between  $G_1$  and  $G_2$ ). We will show that colour  $A$  does not occur in  $G_2$ . Note that, if  $A$  occurs in  $G_{10}$ , then it occurs in  $G_{20}$  as well.

Assume the contrary, and let  $pq$  be an edge of  $A$  in  $G_2$  with, say,  $d(y,p)+1=d(y,q)$ . Then  $u$  and  $p$  are on one side of  $A$ , so that  $x$  and  $q$  are on the other side. Let  $P=y \rightarrow z \rightarrow \dots \rightarrow p$  be a  $y,p$ -geodesic. Then there is an  $x,q$ -geodesic  $Q=x \rightarrow t \rightarrow \dots \rightarrow q$  with  $t$  adjacent to  $z$ . Then  $xt$  and  $yz$  have the same colour, so  $xt$  is in  $F$ . This implies that  $x$  is in  $G_{10}$  contradicting the choice of  $x$ . So the colour  $A$  is fully contained in  $G_1$ , and  $G_{10} \cup G_2$  is on one side of  $A$  and  $x$  on the other side.

Repeating this argument, if necessary, we arrive at an extremal subgraph of  $G$  fully contained in  $G_1$ . Similarly there is an extremal subgraph contained in  $G_2$ .  $\square$

Using Theorem 3.2.7 from [Mu 80b], we can easily verify that a retract of a hypercube is a median graph. To prove that each median graph can be realized as a retract of a hypercube we use induction on the number of colours.

Let  $G$  be a median graph, and let  $F$  be an extremal colour with extremal subgraph  $G_1 = G_{10}$ . We embed  $G$  in an  $n$ -cube  $Q$  as in Theorem 3. Then  $F$  splits  $Q$  into two  $(n-1)$ -cubes  $Q_1$  and  $Q_2$  with  $G_i$  in  $Q_i$ ,  $i=1,2$ . By induction there is a retraction of  $Q_2$  onto  $G_2$ . Apply the corresponding retraction on  $Q_1$ . Then it maps  $Q_1$  onto a copy  $H_1$  of  $G_2$  matched isomorphically via  $F$  to  $G_2$ . This map preserves  $G_1$ . Now we only have to map  $H_1 - G_1$  into  $G_2$  in the right way. If  $u_1$  in  $H_1 - G_1$  has neighbour  $u_2$  in  $G_2 - G_{20}$ , then we map  $u_1$  on a neighbour of  $u_2$ , which is nearer to  $G_{20}$  than  $u_2$ . This is possible whenever we have a distance decreasing map of  $G_2$  into itself, which preserves  $G_{20}$  and maps vertices of  $G_2 - G_{20}$  on neighbours nearer to  $G_{20}$ .

The existence of such a map can again be proved by induction on the number of colours. We omit the details here.

Actually this is precisely the way how WILKEIT [Wi 86] proved that the so-called quasi-median graphs are the retracts of the Cartesian products of arbitrary complete graphs (see Section 4).

## 6. Crossing splits

Two splits  $G_1, G_2$  and  $H_1, H_2$  of a median graph  $G$ , or their associated colours, are said to be *crossing* if  $G_i \cap H_j \neq \emptyset$ , for  $i, j = 1, 2$ . Note that, for a split  $G_1, G_2$  of  $G$ , the subgraph  $G_1$  is extremal if and only if each

colour occurring in  $G_1$  crosses  $F_{12}$  (see [Mu 90]). We use this fact in the following theorem, which has a very simple proof due to the expansion technique.

**Theorem 8.** Let  $G$  be a median graph. Then  $G$  contains  $n$  pairwise crossing splits if and only if  $G$  contains an  $n$ -cube as an induced subgraph.

**Proof.** If  $G$  contains an  $n$ -cube, then the  $n$  colours of this cube extend to pairwise crossing splits in  $G$  because of the 4-cycle property.

Assume  $G$  contains  $n$  pairwise crossing splits  $G_1^k, G_2^k$  for  $k=1, \dots, n$ . Without loss of generality, we may assume that  $G$  has no other splits. Otherwise we could contract these, and the contraction would still contain  $n$  pairwise crossing splits, and the existence of an  $n$ -cube in contraction yields an  $n$ -cube in any expansion by its history.

Note that now every colourhalf  $G_i^k$  is an extremal subgraph of  $G$ , i.e., for  $k=1, \dots, n$ , colour  $F_{12}^k$  yields an isomorphism between  $G_1^k = G_{10}^k$  and  $G_2^k = G_{20}^k$ . Using induction on the number of colours  $n$  in  $G$ , we may conclude that both  $G_1^k$  and  $G_2^k$  are  $(n-1)$ -cubes, so that  $G$  is an  $n$ -cube.  $\square$

## 7. The hull number of a median graph

The intersection of convex sets in a graph is again convex. This gives rise to the following definition. Let  $W$  be a subset of vertices in a graph  $G=(V,E)$ . The *convex hull* of  $W$ , denoted by  $Con(W)$ , is the smallest convex subgraph of  $G$  containing  $W$  (see [Mu 80b], where it was termed the convex closure). A set  $S \subseteq V$  *generates*  $G$  if  $Con(S)=G$ . In [ES 85] EVERETT and SEIDMAN introduced the *hull number*  $h(G)$  of a graph  $G$  to be the size of a *minimum generating set*. Here of course, minimum means that there is no generating set with fewer vertices.

Any two diametrical vertices (vertices at largest distance) generate a hypercube. So  $h(Q)=2$ , for any hypercube  $Q$  except  $K_1$ . In a tree  $T$  we need all end vertices to generate  $T$ . By convention an end vertex will be the vertex of degree zero if  $T=K_1$ , and a vertex of degree one otherwise. Clearly,  $h(T)$  is the number of end vertices in  $T$ .

In this subsection we consider (minimum) generating sets of median graphs. We say that a set  $W$  *touches* a subgraph  $H$  of  $G$  if  $H$  contains a vertex of  $W$ . The following three results are obvious (we use the above notations).

**Lemma 9.** If  $S$  generates the median graph  $G$  and  $G'$  is a contraction of  $G$ , then  $S'$  generates  $G'$ .

**Corollary 10.** If  $G'$  is a contraction of the median graph  $G$ , then  $h(G) \geq h(G')$ .

**Lemma 11.** If  $G_1, G_2$  is a split in a median graph  $G$  generated by  $S$ , then  $S$  touches  $G_1$  as well as  $G_2$ .

The main result of this subsection is the following theorem.

**Theorem 12.** Let  $S$  be a set of vertices touching each extremal subgraph of a median graph  $G$ . Then  $S$  generates  $G$ .

**Proof.** We use induction on the number of expansions. Let  $F$  be an extremal colour with split  $G_1, G_2$  and  $G_1 = G_{10}$ . We may take  $G_2$  as the contraction of  $G$  with respect to  $F$ . Note that every colour in  $G_1$  occurs in  $G_{20}$  as well, and vice versa.

Every extremal colour of  $G$  distinct from  $F$  is an extremal colour of  $G_2$ . So all extremal subgraphs of  $G_2$  associated with these colours are touched by  $S'$ . If  $A$  is a non-extremal colour in  $G_1$ , then it is also non-extremal in  $G_{20}$  as well as in  $G_2$ .

Assume that  $B$  is an extremal colour in  $G_2$  that is not extremal in  $G$ . Then  $G_{20}$  must be contained in the extremal subgraph of  $B$ . Since  $S$  touches  $G_1$ , it follows that  $S'$  touches  $G_{20}$ , so it touches the extremal subgraph of  $B$  in  $G_2$  as well. Hence  $S'$  touches all extremal subgraphs of  $G_2$ .

By induction  $S'$  generates  $G_2$ . Let  $w_1, x_1, \dots, z_1$  be the vertices of  $S$  in  $G_1$ , and let  $w_2, x_2, \dots, z_2$  be their respective neighbouring gates in  $G_{20}$ . Since  $S$  generates  $G$ , it touches  $G_2$ , say in  $v$ . Then  $w_2$  lies in  $I(w_1, v)$ , etcetera. So  $Con(S)$  contains  $w_2, x_2, \dots, z_2$ . Therefore  $Con(S)$  contains  $Con(S') = G_2$ , in particular  $Con(S)$  contains  $G_{20}$ . Take any vertex  $p_2$  in  $G_{20}$  with neighbouring gate  $p_1$  in  $G_1$ . Then  $I(w_1, p_2)$  contains  $p_1$ . So  $G_1 = G_{10}$  is contained in  $Con(S)$  as well, and we are done.  $\square$

The following theorem is an immediate consequence.

**Theorem 13.** Let  $G$  be a median graph. Then  $h(G)$  is equal to the minimum number of vertices touching all extremal subgraphs of  $G$ .

It is easily seen that one can actually decrease the hull number by contractions. But what are the contractions that preserve the hull number? In a tree one can contract all internal edges, thus obtaining a *star* (a  $K_{1,n}$ ) with the same number of end vertices. Contracting any further edge decreases the hull number. In a hypercube we can contract all colours but one, thus obtaining the star  $K_{1,1}$  with the same hull number. By convention we will consider  $K_1$  also to be a star.

A *star contraction* of a median graph  $G$  is a star obtained by successive contractions of  $G$ . Let  $T$  be a star contraction of  $G$  with the maximum possible number of end vertices. We define  $\tau(G)$  to be the number of end vertices of this star  $T$ . Then we get the following problem.

**Question.** For which median graphs  $G$  do we have  $h(G) = \tau(G)$ ?

## 8. Quasimedial graphs

Almost all of the above results can be generalized to *quasimedial graphs*, which generalize median graphs. These graphs were introduced and characterized by another expansion procedure in [Mu80b]. For the relevant theorems on retracts see [CGS 89] and [Wi 86], and for the generalization of the dynamic search problem, see [CGS 89].

## References

- [Av 61] S.P. AVANN, Metric ternary distributive semi-lattices, Proc. Amer. Math. Soc. **12** (1961) 407–414.
- [Ba 84] H.-J. BANDELT, Retracts of hypercubes, J. Graph Theory **8** (1984) 501–510.
- [BB 84] H.-J. BANDELT and J.P. BARTHÉLEMY, Medians in median graphs, Discrete Appl. Math. **8** (1984) 131–142.
- [BH 83] H.-J. BANDELT and J. HEDLIKOVÁ, Median algebras, Discrete Math. **45** (1983) 1–30.
- [CGS 87] F.R.K. CHUNG, R.L. GRAHAM and M.E. SAKS, Dynamic search in graphs, in Discrete Algorithms and Complexity, Academic Press (1987) 351–388.
- [CGS 89] F.R.K. CHUNG, R.L. GRAHAM and M.E. SAKS, A dynamic location problem for graphs, Combinatorica **9** (1989) 111–131.

- [Dr 87] H. DRÁŠKOVICOVÁ, Weak direct product decompositions of algebras, *Contributions to General Algebra* **4**, Hölder-Pichler-Tempsky (1987), Wien.
- [Es 85] M.G. EVERETT and S.B. SEIDMAN, The hull number of a graph, *Discrete Math.* **57** (1985) 217–223.
- [Is 80] J.R. ISBELL, Median algebra, *Trans. Amer. Math. Soc.* **260** (1980) 319–362.
- [JS] P.K. JHA and G. SLUTZKI, Convex-expansions algorithms for recognition and isometric embedding of median graphs, *Ars Comb.* **34** (1992) 75–92.
- [vM 77] Supercompactness and Wallman spaces, Ph. D. Thesis, Vrije Universiteit Amsterdam, 1977.
- [MMR 94] F.R. McMorris, H.M. Mulder and F.R. Roberts, The median procedure on median graphs, Report 9413/B, Econometrisch Instituut EUR, 1994, submitted.
- [Mu 78] H.M. MULDER, The structure of median graphs, *Discrete Math.* **24** (1978) 197–204.
- [Mu 80a] H.M. MULDER,  $n$ -cubes and median graphs, *J. Graph Theory* **4** (1980) 107–110.
- [Mu 80b] H.M. MULDER, The interval function of a graph, *Math. Centre Tract* **132**, Math. Centre Amsterdam (1980).
- [Mu 90] H.M. MULDER, The expansion procedure for graphs, in: R. Bodendiek ed., *Contemporary methods in graph theory*, Wissenschaftsverlag, Mannheim etc., 1990, pp. 459–477.
- [Ne 71] L. NEBESKÝ, Median graphs, *Comment. Math. Univ. Carolinae* **12** (1971) 317–325.
- [Wi 86] E. WILKEIT, Isometrische Untergraphen von Hamming-Graphen, Doctoral thesis, Universität Oldenburg (1986).
- [Wr 87] P. WINKLER, The metric structure of graphs: theory and applications, in: *Surveys in Combinatorics 1987*, C. Whitehead ed., Cambridge University Press (1987) 197–221.



# Control of Discrete Event Systems – Research at the Interface of Control Theory and Computer Science

Ard Overkamp  
Jan H. van Schuppen

CWI

*P.O. Box 94079, 1090 GB Amsterdam, The Netherlands*

This expository paper is directed to a general audience of engineers, mathematicians, and computer scientists. A discrete event system is a mathematical model (in the form of an automaton, Petri nets, or process algebra) of, for example, a computer controlled engineering system such as a communication network. Control theory for discrete event systems aims at synthesis procedures for a supervisor that forces a discrete event system such that it satisfies prespecified control objectives. As an example it is discussed how the control problem of blocking prevention for nondeterministic systems may be solved by the use of failure semantics.

*AMS Subject Classification (1991):* 93B50, 93C30, 68B20, 68D45.

*Keywords and Phrases:* Discrete event systems, automata, supervisory control, failure semantics.

This paper is dedicated to P.C. Baayen for his stimulation of the interaction between mathematics and computer science.

## 1 INTRODUCTION

The purpose of this paper is to introduce the reader to the research topic of control of discrete event systems. This expository paper is written for a general audience of engineers, mathematicians, and computer scientists. No specific background is needed neither of system and control theory nor of computer science. Only subsection 4.2 contains results derived at CWI.

The motivation for control of discrete event systems comes from control of engineering systems, manufacturing processes, and computer systems. Examples are online scheduling of transactions in databases, control of a rapid thermal

processor, and design of a communication protocol. The control objectives in such problems are, for example, liveness, safety, and prevention of blocking.

In modeling of practical control problems use is made of models from computer science: automata, Petri nets, and process algebras. A discrete event system is often taken to be an automaton in which the outside world can influence the occurrence of events.

The control problem for discrete event systems is then often formulated as: Construct a supervisor which observes the events of the system and determines after every event which elements of the set of possible next events must be prevented from occurring. Control objectives are as mentioned above, primarily to guarantee a certain level of liveness and safety.

Control theory for discrete event systems makes use of several subareas of computer science such as automata theory, process algebras, logic, temporal logic, complexity, etc.

A description of the paper by section follows. Section 2 contains motivation and Section 3 models of discrete event systems. Control synthesis problems are discussed in Section 4. Guidelines for further reading may be found in Section 5.

## 2 MOTIVATION

Research in control of discrete event systems is motivated by practical control problems in, for example, communication networks, databases, manufacturing systems, and traffic systems (metro lines, railways, and freeway traffic). See for references Section 5.

**Example 2.1** Consider a telephone network. Subscribers can generate events such as ‘taking the receiver off the hook’, ‘replacing the receiver’, ‘press a button’. The telephone network itself also generates events, such as ‘ring the bell’, ‘start the dialtone’, ‘establish a connection’. Some sequences of events represent unwanted behavior. For instance, a bell should not ring if the receiver is off the hook. Other sequences represent wanted behavior. For instance, a connection should be established if the right protocol is followed by both subscribers. The caller should have taken the receiver off the hook, waited for the dialtone, dialed the correct number, etcetera. Some of these event sequences are enforced by the hardware of the telephone network. A receiver can only be replaced after it is taken off the hook. Some sequences have to be enforced by a supervisor. In the old days a human operator was necessary to guarantee the correct behavior. Nowadays a computer does the job. The challenging task is to automatically synthesize the computer program when provided information only about the uncontrolled behavior of the telephone network and the control objectives.

Practical control problems in the areas mentioned lead to control problems at the level of engineering and computer science. Examples of control objectives for problems at this level are:



1. *Safety*. Prevent the controlled system from reaching states at which a disaster may occur.
2. *Liveness*. Guarantee that the controlled system is able to perform a specified minimum level of performance.

An example of a safety property is *blocking*: Prevent that the controlled system reaches a state from which it cannot proceed to any other state. One speaks of *deadlock* in case of a system with two independently operating supervisors in the situation where both supervisors are waiting for each other [10]. Control problems at the engineering level are transformed into control problems at the control theory level, see section 4 below.

Terminology of systems and control is summarized below. An *event* is the occurrence of an action. A *discrete event system* or *plant* is a mathematical model of an object exhibiting a sequence of events. A *supervisor* is the mathematical object that restricts the operation of a discrete event system. The discrete event system in connection with the supervisor will be called the *controlled discrete event system* or the *closed-loop system*. A *control objective* is a specification on the behavior of the closed-loop system.

Control problems for discrete event systems at the level of control theory lead in general to the following theoretical questions:

- *Existence*. Does there exist a supervisor such that the closed-loop system satisfies the control objectives?
- *Decidability*. Can a supervisor be constructed with an algorithm that terminates in a finite number of steps?
- *Algorithm*. How to construct an algorithm that produces a supervisor meeting the control objectives?
- *Complexity*. How, polynomially or exponentially, does the number of computations of an algorithm for the construction of a supervisor depend on the parameters of the problem?

It is the aim of control theory for discrete event systems to answer questions as these.

The approach to control of a discrete event system taken in the research area of systems and control differs from that taken in computer science. In control theory the approach is control synthesis. In computer science the approach is specification and verification. Thus, in computer science a specification is made that the controlled system is to satisfy, an implementation for the closed-loop system is made, and finally it is verified whether or not the closed-loop system meets the specification. The latter step is called *verification*. Verification can be done by automata theoretic tools, see [26]. Simulation is a popular method to test whether an algorithm or program satisfies the specification but for most practical problems complete testing by simulation is unfeasible. Which

approach to control is to be preferred, that of control theory or that of computer science? The answer to this question must be based on experience with a large number of practical control problems for discrete event systems. It will take several more years to collect such experience.

The area of modeling and control of discrete event systems is intertwined with computer science. Modeling of discrete event systems is based on computer science models. Also logic and temporal logic is used in both areas. The subject of verification is also of interest to systems and control. The use of computers in engineering and data processing is expected to lead to new control problems for which systems and control, and computer science will be needed.

The approach of supervisory control is entirely different from control theory as practised in stochastic control and from control of queues as practised in operations research. In the latter areas the processing time is of major interest. Correctness is the major concern in control of discrete event systems. In timed discrete event systems the concept of time also appears but often in constraints and not as part of the cost function.

### 3 MODELING OF DISCRETE EVENT SYSTEMS

To model practical control problems in terms of computer science concepts the following model classes are currently used: (1) automata; (2) Petri nets; (3) process algebras. Automata will be described in detail below.

Which model class is to be preferred for the practical control problems mentioned in section 2? The choice of a model class must be a trade-off between descriptive power and complexity. In regard to descriptive power it has been proven that the model class of Petri nets strictly contains the automata, while the model class of process algebras strictly contains Petri nets. A control problem in Petri nets or in process algebras may be undecidable, that is, there does not exist an algorithm for that problem which terminates in a finite number of steps. The authors prefer automata theory over Petri nets because the concept of state is more explicit. This makes control synthesis easier. The authors like the model class of process algebras because of its modeling power. In many engineering disciplines the model class of Petri nets is rather popular.

There follows terminology and notation on automata.

**DEFINITION 3.1** *An automaton is a collection*

$$A = (\Sigma, X, d, x_0, X_m),$$

where  $\Sigma$  is a finite set of event labels called the alphabet,  $X$  is set of states,  $x_0 \in X$  is the initial state,  $X_m \subset X$  is the set of marked states, and  $d : \Sigma \times X \rightarrow X$  is said to be the transition function.

A generator is an automaton in which  $d : \Sigma \times X \rightarrow X$  is only a partial function. Thus, for  $x \in X$  there is a subset  $\Sigma(x) \subset \Sigma$  such that  $d(., x) : \Sigma(x) \rightarrow X$  is a function.

A finite state system is a generator in which  $X$  is a finite set. The term discrete event system, or for short system or plant, is defined in this paper to be an automaton.

In an automaton events are modelled as to occur spontaneously. The mechanism that selects an event is not modelled. In a discrete event system there is no model for a clock. Events occur sequentially.

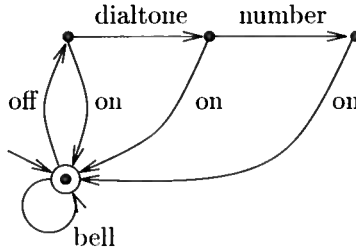


FIGURE 1. Automaton representing a telephone unit.

**Example 3.2** A complete model of a system such as a telephone network consists usually of a number of modules, each representing a part of the total system. In Fig. 1 the automaton representing the behavior of the telephone unit is shown. The off-event represents the lifting of the receiver. The on-event represents putting the receiver back on the hook. Encircled nodes indicate marked states. The small arrow that does not start at a node points to the initial state. The automaton describes that a number can only be chosen after a dialtone is given. This is an abstraction of the fact that buttons can be pressed before a dialtone is given, but that these actions do not result in an event inside the system. It is not represented in the automaton that a dialtone should not be given when the receiver is taken off the hook to answer a call. This behavior has to be enforced by a supervisor.

An automaton generates a language which concept is defined below.

**DEFINITION 3.3** Let  $\Sigma$  be a set which will be called the event alphabet. A string is a sequence of events

$$s = (\sigma_1, \sigma_2, \dots, \sigma_n),$$

where for  $n \in \mathbb{Z}_+$ ,  $i = 1, \dots, n$ ,  $\sigma_i \in \Sigma$ . The empty string, denoted by  $\epsilon$ , is defined to be the string without elements. The set of all strings over  $\Sigma$  including the empty string, is denoted by  $\Sigma^*$ . A language is defined to be a subset  $L \subset \Sigma^*$ .

The prefix closure of a language  $L$ , denoted by  $\bar{L}$ , is defined to be the set

$$\bar{L} = \{s \in \Sigma^* \mid \exists t \in \Sigma^* \text{ such that } st \in L\}.$$

The language  $L \subset \Sigma^*$  is said to be prefix closed if  $L = \bar{L}$ .

DEFINITION 3.4 Let  $G = (\Sigma, X, d, x_0, X_m)$  be a generator. Extend the transition function  $d : \Sigma \times X \rightarrow X$  to  $d : \Sigma^* \times X \rightarrow X$  by

$$\begin{aligned} d(\epsilon, x) &= x, \\ d(s\sigma, x) &= d(\sigma, d(s, x)), \text{ if well defined and for } \sigma \in \Sigma^*, s \in \Sigma. \end{aligned}$$

The behavior of  $G$  is defined to be the language

$$L(G) = \{s \in \Sigma^* \mid d(s, x_0) \text{ is defined}\}, \quad (1)$$

and the marked behavior is defined to be the language

$$L_m(G) = \{s \in L(G) \mid d(s, x_0) \in X_m\}. \quad (2)$$

A string in  $L_m(G)$  is said to be a marked string.

A string in the behavior is a finite string that  $G$  can generate. A string in the marked behavior is a finite string that ends in a marked state. A marked string represents a completed task. If  $G$  is a generator then by definition of  $L(G)$  this set is prefix closed.

Does there exist a generator  $G$  for a given language  $L$  such that the language generated by  $G$  equals  $L$ , or  $L(G) = L$ ? This representation problem is rather fundamental in system theory and automata theory. Not every language has such a representation. A major theorem of automata theory, see [21, Section 2.5], states that any regular language can be represented by a finite-state automaton. The concept of a regular language will not be defined in this paper because of space limitations. Thus, a finite-state generator produces a regular language while a regular language can be represented as being generated by a generator. The formalisms of regular languages and of finite state generators are thus equivalent. In the remainder of the paper the two formalisms are used interchangeably, with most results being formulated in terms of languages.

Control can be enforced by synchronization of the plant with a controller (supervisor). A supervisor can only block events of the plant. It cannot enforce the execution of an event.

DEFINITION 3.5 The synchronous composition of plant  $G = (\Sigma, X, d_g, x_0, X_m)$  and supervisor  $S = (\Sigma, Q, d_s, q_0, Q_m)$  is the automaton  $G||S = (\Sigma, X \times Q, d_{gs}, (x_0, q_0), X_m \times Q_m)$ , where

$$d_{gs}(\sigma, (x_g, q_s)) = \begin{cases} (d_g(\sigma, x_g), d_s(\sigma, q_s)), & \text{if well defined.} \\ \text{undefined,} & \text{otherwise.} \end{cases}$$

Events in the synchronous composition are possible only if they are possible in the plant as well as in the supervisor. Then

$$\begin{aligned} L(G||S) &= L(G) \cap L(S), \\ L_m(G||S) &= L_m(G) \cap L_m(S). \end{aligned}$$

## 4 CONTROL SYNTHESIS

The purpose of this section is to describe how practical control problems are formulated and solved at the level of control theory.

### 4.1 Supervisory control synthesis

In this subsection the basic concepts of supervisory control, as introduced by Ramadge and Wonham [39], will be explained. The general problem of control theory for discrete event systems is to find a controller (supervisor) that influences the behavior of the plant in such a way that it meets the control objectives.

In some applications the supervisor does not have the ability to block all events. For instance if an alarm event is executed when some water level exceeds a threshold, then this event can be observed by the supervisor but it cannot be blocked. If this event has to be prevented from occurring then somewhere else in the system some other events have to be blocked (For instance the closing of a waste gate) such that the alarm event cannot occur anymore. Usually the presence of uncontrollable events is modelled by splitting up the event set into two subsets  $\Sigma_c$  and  $\Sigma_u$ , where  $\Sigma_c$  represents the controllable events, and  $\Sigma_u$  the uncontrollable events. It is required that a supervisor never blocks an uncontrollable event. Such a supervisor is called *complete*.

The main objective of control synthesis for discrete event systems is to find a complete supervisor which allows only legal event sequences. These sequences together form the *legal language*. This language is specified by an automaton, denoted  $E$ , which generates exactly all legal strings. The basic control objective is to find a complete supervisor such that  $L(G||S) = L(E)$ . It was shown by Ramadge and Wonham that this supervisor exists only if the plant cannot go from a legal string to an illegal string by executing only uncontrollable events. This property is formulated in the controllability condition.

**DEFINITION 4.1** *Let  $G$  be a plant and  $\Sigma_u$  the set of uncontrollable events. The language  $K$  is said to be controllable if*

$$\bar{K}\Sigma_u \cap L(G) \subseteq \bar{K},$$

where  $\bar{K}\Sigma_u = \{s\sigma \in \Sigma^* | s \in \bar{K}, \sigma \in \Sigma_u\}$ .

**THEOREM 4.2** *Let  $G$  be a plant and  $E$  a specification of the legal behavior, with  $L(E) \subseteq L(G)$ . There exists a complete supervisor,  $S$ , such that  $L(G||S) = L(E)$  if and only if the language  $L(E)$  is controllable.*

If the language  $L(E)$  is not controllable then there exists no supervisor such that  $G||S$  generates exactly all legal strings. The control objectives may be relaxed such that any system that generates no illegal strings is satisfactory. Thus, a supervisor is looked for such that  $L(G||S) \subseteq L(E)$ .

**THEOREM 4.3** *Let  $G$  be a plant and  $E$  a specification of the legal behavior, with  $L(E) \subseteq L(G)$ . There exists a complete supervisor,  $S$ , such that  $L(G||S) \subseteq L(E)$  if and only if there exist a prefix-closed and controllable language contained in  $L(E)$ .*

Ramadge and Wonham also showed that the set of languages that are prefix-closed, controllable and contained in  $L(E)$  is closed under arbitrary unions. This implies that there is a unique supremal element in this set. That is, there exists a language such that all languages that are controllable and contained in  $L(E)$  are a subset of this language. From lattice theory a fixed point algorithm is known that computes this supremal language. This algorithm has polynomial complexity with respect to the number of states in the state space of the automata  $G$  and  $E$ . The automaton that generates this supremal language can be used as supervisor.

#### 4.2 Blocking

The relaxed control objective in the previous section does not guarantee that the closed-loop system will never block. After the system has generated a certain string, it may happen that no subsequent event is possible. Either events cannot be generated by the plant, or the supervisor blocks the events. The marked behavior, as defined in Section 2, may be used to guarantee that systems are nonblocking. Because we will use another definition of nonblocking later on, we will indicate this form with marking-nonblocking. A system is said to be marking-nonblocking if every string that the system generates can be extended to a marked string.

**DEFINITION 4.4** *System  $E$  is said to be marking-nonblocking if*

$$L(E) = \overline{L_m(E)}.$$

The supervisory control problem for systems with marking is to find a complete supervisor such that  $L_m(G||S) \subseteq L_m(E)$  and  $G||S$  is marking-nonblocking. Note that these two conditions together imply that no illegal string will be generated. That is,  $L(G||S) = \overline{L_m(G||S)} \subseteq \overline{L_m(E)} \subseteq L(E)$ .

**THEOREM 4.5** *Let  $G$  be a plant and  $E$  the specification of the legal behavior, with  $L_m(E) \subseteq L_m(G)$ . There exists a complete supervisor,  $S$ , such that  $L_m(G||S) \subseteq L_m(E)$  and  $L(G||S) = \overline{L_m(G||S)}$ , if and only if there exist a controllable language contained in  $L_m(E)$ .*

Note that the definition of controllability is given for general, not necessarily prefix-closed, languages. As in the previous section, the set of languages that are controllable and contained in  $L_m(E)$  is closed under arbitrary unions. This means that there exists a supremal language in this set. Let  $K$  be this supremal language. Then, the automaton,  $S$ , with  $L(S) = \bar{K}$  and  $L_m(S) = K$  can be used as a supervisor.

If some parts of a system are not completely modelled, or only a part of the system can be observed, then the system may exhibit nondeterministic behavior. That is, the observed sequence of events does not uniquely determine the state of the system.

**DEFINITION 4.6** A nondeterministic automaton is defined to be a automaton in which the transition function  $d$  is of the form  $d : \Sigma \times X \rightarrow 2^X$ . The set  $d(\sigma, x)$  precisely contains all states that can be reached from state  $x$  by event  $\sigma$ .

Consider now the supervisory control problem of blocking prevention for nondeterministic systems. Marking is not sufficient to guarantee that nondeterministic systems are nonblocking. Consider the following example.

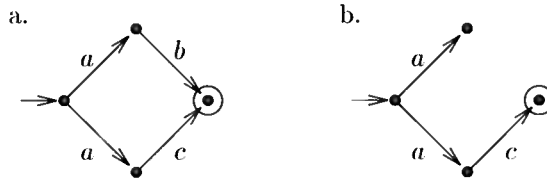


FIGURE 2. Blocking of a nondeterministic automaton.

**Example 4.7** Let  $G$  be an automaton as in Fig. 2.a. Suppose string  $ab$  is illegal. If event  $b$  is blocked, then an automaton as in Fig. 2.b is obtained. It is clear that this system can block after event  $a$ . But this is not detectable by considering the marked language. From  $L(G) = \{a, ac\} = \overline{\{ac\}} = \overline{L_m(G)}$  it follows that  $G$  is marking-nonblocking.

Hoare [19] introduced a different method to deal with blocking in nondeterministic systems. Not only the language of the system is considered but also the events that cannot be executed are taken into account. If a nondeterministic system is offered a set of admissible events, via the synchronous composition, and the system can be in a state in which it cannot execute any of the offered events, then the system is said to refuse all events in this set. Such a set of events is called a refusal.

**DEFINITION 4.8** The set of refusals or the refusal set of the system  $G$  after string  $s$  is defined to be the set

$$\text{ref}(G, s) = \{R \subseteq \Sigma : \exists x \in d(s, x_0) \text{ s.t. } R \cap \lambda(x) = \emptyset\},$$

where  $\lambda(x) = \{\sigma \in \Sigma \mid d(\sigma, x) \text{ is defined}\}.$

Note that a refusal is a set of events. In the definition above  $R$  is a refusal. So a refusal set or set of refusals is a set of sets of events.

The method introduced by Hoare is known as failure semantics. Using this method, blocking of a nondeterministic system can be defined as the situation in which all events can be refused.

DEFINITION 4.9 *The (nondeterministic) system  $G$  is said to be nonblocking if for all  $s$  in  $L(G)$ ,  $\Sigma \notin \text{ref}(G, s)$ .*

The controlled system is guaranteed to be nonblocking if the legal behavior is nonblocking and the controlled system does not refuse more than the system describing the legal behavior. This statement motivates the reduction relation.

DEFINITION 4.10 *One says that system  $G$  reduces system  $E$ , denoted  $G \sqsubseteq E$ , if*

$$L(G) \subseteq L(E), \text{ and} \\ \forall s \in L(G), \text{ref}(G, s) \subseteq \text{ref}(E, s).$$

The supervisory control problem for nondeterministic systems is to find a complete supervisor,  $S$ , such that  $G||S \sqsubseteq E$ . It is shown in [37] that an important condition for the existence of such a supervisor is the reducibility condition.

DEFINITION 4.11 *Language  $K$  is said to be reducible (w.r.t.  $G$  and  $E$ ) if*

$$\forall s \in K, \quad \forall R_g \in \text{ref}(G, s), \quad \rho(K, s) \cup R_g \in \text{ref}(E, s), \\ \text{where } \rho(K, s) = \{\sigma \in \Sigma \mid s\sigma \notin K\}.$$

THEOREM 4.12 *Let  $G$  be a nondeterministic system and  $E$  a specification of the legal behavior. There exists a complete supervisor,  $S$ , such that  $G||S \sqsubseteq E$ , if and only if there exists a controllable and reducible language contained in  $L(E)$ .*

Thus, if  $E$  is nonblocking and if there exists a controllable and reducible language contained in  $L(E)$ , then there exists a supervisor  $S$  such that  $L(G||S) \subseteq L(E)$  and  $G||S$  is nonblocking.

It has been shown that the set of reducible and controllable languages is closed under arbitrary unions. So a unique supremal language is contained in the set and is computable by a fixed point algorithm with polynomial complexity. As with deterministic systems, the (deterministic) automaton that generates this supremal language can be used as supervisor.

## 5 GUIDELINES FOR FURTHER READING

Practical control problems for which control of discrete event systems has been analysed include: database operations [27]; rapid thermal multiprocessor [6]; and protocol design for communication networks [13, 17, 40].

Automata theory at an introductory level may be found in [21] and at an advanced level in [15]. A book on Petri nets is [14] and a book on related models [4]. The theory of process algebras may be found in [5, 18, 19, 31]. For temporal logic see [30].

Supervisory control of discrete event systems was started and mainly developed by W.M. Wonham and his doctoral students, see [38, 39, 41, 43, 45, 47].



An overview paper is [44]. For publications of S. Lafortune and co-workers see [11, 12, 27, 28]. The supervisory control problem with failure semantics is treated in [37]. A large quantity of additional publications remains unmentioned because of space limitations.

Control of infinite string automata was developed by J.G. Thistle [43]. Such strings are used to express liveness conditions. Techniques to do verification for the associated languages were developed by R.P. Kurshan [26]. A book by Kurshan will appear shortly. Modeling for control of discrete event systems by process algebras was considered by K. Inan and P. Varaiya in [22, 23].

Time plays a role in many practical control problems. Examples of such problems are the operation of a railway gate [36] or the operation of a telephone network. Timed discrete event systems are closely related to the computer science area of real-time systems. A stimulating discussion on theoretical concepts for real-time systems is presented in [25, 42]. Modeling of timed discrete event systems brings with it several new issues compared with untimed discrete event systems, such as the role of durations and forcing of events. Models of timed discrete event systems that have been proposed include discrete-time systems [7], timed automata proposed by R. Alur and D. Dill [1, 3], temporal logic [36], and timed process algebras developed by J. Sifakis and co-workers [32, 33, 34]. Control of timed discrete event systems is treated in [7, 20, 29, 36, 46] of which the work by G. Hoffmann and H. Wong-Toi is of particular interest. An application to specification and design of a telephone exchange is presented in [24].

A hybrid system is a mathematical model of a phenomenon in which the model includes logical variables and continuous variables described by differential equations. Many computer controlled engineering systems are hybrid systems, for example a temperature controller for a house or the controller of an air plane. Models for hybrid systems were proposed in [2, 9, 16, 34, 35]. For an approach to control of hybrid systems, see [8].

## 6 CONCLUDING REMARKS

What has been achieved in control of discrete event systems? For practical problems with logical variables discrete event systems have been formulated as mathematical models. These systems are the basic building blocks for control. Control synthesis results yield algorithms that produce supervisors that will satisfy a specified level of performance.

What research directions should be explored in control of discrete event systems? First experience must be gained with realistic and practical problems as they appear in industrial laboratories. Modeling of discrete event systems would benefit from a deeper analysis of the trade-off between modeling power and complexity. Hierarchical decomposition may be a direction to explore. A discrete event system has little mathematical structure hence it is difficult to enlist the aid of parts of traditional mathematics. The developments in theoretical computer science should be watched closely. Control theory of discrete

event systems should also concentrate attention on decentralized control motivated by the use of networks of computers. Faster algorithms for control synthesis would be useful in practice.

Control of timed discrete event systems needs more motivation by realistic and practical problems. Experience must be gained with the model classes of timed discrete event systems and timed process algebras. Control of hybrid systems leads to a diverse set of problems. Research in this area has only recently started.

#### ACKNOWLEDGEMENTS

The authors acknowledge their influence by the pioneering research in control of discrete event systems of W.M. Wonham (University of Toronto) and his doctoral students. They also acknowledge discussions on the subject with R.K. Boel, K. Inan, S. Lafortune, F. Vaandrager, and P. Varaiya.

#### REFERENCES

1. R. Alur, C. Courcoubetis, and D. Dill. Model-checking for real-time systems. In X, editor, *Proc. of the 5th IEEE Symposium on Logic in Computer Science*, pages 414–425, X, 1990. X.
2. R. Alur, C. Courcoubetis, T. Henzinger, P. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. In G. Cohen and J.-P. Quadrat, editors, *11th International Conference on Analysis and Optimization of Systems - Discrete Event Systems*, number 199 in Lecture Notes in Control and Information Sciences, pages 331–351, London, 1994. Springer-Verlag.
3. R. Alur and D. Dill. The theory of timed automata. In J.W. de Bakker, C. Huizing, and W.P. de Roever, editors, *Real-time: Theory and practice. Proceedings of the REX Workshop, Mook, The Netherlands, June 3-7, 1991*, number 600 in Lecture Notes in Computer Science, pages 45–74, Berlin, 1992. Springer.
4. F.L. Baccelli, G. Cohen, G.J. Olsder, and J.-P. Quadrat. *Synchronization and linearity - An algebra for discrete event systems*. John Wiley & Sons, Chichester, 1992.
5. J.C.M. Baeten and W.P. Weijland. *Process algebra*. Cambridge University Press, Cambridge, 1990.
6. S. Balemi, G.J. Hoffmann, P. Gyugyi, H. Wong-Toi, and G.F. Franklin. Supervisory control of a rapid thermal multiprocessor. *IEEE Trans. Automatic Control*, 38:1040–1059, 1993.
7. B.A. Brandin and W.M. Wonham. Supervisory control of times discrete event systems. *IEEE Trans. Automatic Control*, 39:329–342, 1994.
8. M.S. Branicky, V.S. Borkar, and S.K. Mitter. A unified framework for hybrid control: Background, model, and theory. Report LIDS-P-2239, Laboratory for Information and Decision Systems, M.I.T., Cambridge, MA, 1994.

9. R.W. Brockett. Hybrid models for motion control systems. In H.L. Trentelman and J.C. Willems, editors, *Essays on control: Perspectives in the theory and its applications*, pages 29–53. Birkhäuser, New York, 1993.
10. A. Burns and A. Wellings. *Real-time systems and their programming languages*. Addison-Wesley, X, 1990.
11. E. Chen and S. Lafortune. Dealing with blocking in supervisory control of discrete-event systems. *IEEE Trans. Automatic Control*, 36:724–735, 1991.
12. Sheng-Luen Chung, S. Lafortune, and Feng Lin. Limited lookahead policies in supervisory control of discrete event systems. *IEEE Trans. Automatic Control*, 37:1921–1935, 1992.
13. R. Cieslak, C. Desclaux, A.S. Fawaz, and P. Varaiya. Supervisory control of discrete-event processes with partial observations. *IEEE Trans. Automatic Control*, 33:249–260, 1988.
14. R. David and H. Alla. *Petri nets and grafcet*. Prentice Hall, New York, 1992.
15. S. Eilenberg. *Automata, languages, and machines (Volumes A and B)*. Academic Press, New York, 1974, 1976.
16. R.L. Grossman, A. Nerode, A.P. Ravn, and H. Rischel, editors. *Hybrid systems*. Number 736 in Lecture Notes in Computer Science. Springer, New York, 1993.
17. E. Haghverdi and K. Inan. Verification by consecutive projections. In X, editor, *FORTE 92 Proceedings*, pages x–y, X, 1992. X.
18. M. Hennessy. *Algebraic theory of processes*. M.I.T. Press, Cambridge, MA, 1988.
19. C.A.R. Hoare. *Communicating sequential processes*. Prentice/Hall International, Englewood Cliffs, NJ, 1985.
20. G.J. Hoffmann and H. Wong-Toi. The input-output control of real-time discrete event systems. Report ISL/GFF/92-1, Information Systems Laboratory, Stanford University, Stanford, 1992.
21. J.E. Hopcroft and J.D. Ullman. *Introduction to automata theory, languages, and computation*. Addison-Wesley Publishing Company, Reading, MA, 1979.
22. K. Inan and P. Varaiya. Finitely recursive process models for discrete event systems. *IEEE Trans. Automatic Control*, 33:626–639, 1988.
23. K.M. Inan and P.P. Varaiya. Algebras of discrete event models. *Proc. IEEE*, 77:24–38, 1989.
24. A. Kay and J.N. Reed. A rely and guarantee method for times CSP: A specification and design of a telephone exchange. *IEEE Trans. Software Eng.*, 19:625–639, 1993.
25. R. Kurki-Suonio. Real-time: Further misconceptions (or half-truths). *Computer*, 27:71–76, 1994 (June, no. 6).
26. R.P. Kurshan. Automata-theoretic verification of coordinating processes. In G. Cohen and J.-P. Quadrat, editors, *11th International Conference on Analysis and Optimization of Systems - Discrete Event Systems*, number 199 in Lecture Notes in Control and Information Sciences, pages 16–28,

- London, 1994. Springer-Verlag.
27. S. Lafortune. Modeling and analysis of transaction execution in database systems. *IEEE Trans. Automatic Control*, 33:429–447, 1988.
  28. S. Lafortune and Enke Chen. The infimal closed controllable superlanguage and its application in supervisory control. *IEEE Trans. Automatic Control*, 35:398–405, 1990.
  29. Y. Li and W.M. Wonham. Supervisory control of real-time discrete event systems. *Information Sciences*, 46:159–183, 1988.
  30. Z. Manna and A. Pnueli. *The temporal logic of reactive and concurrent systems - Specification*. Springer-Verlag, Berlin, 1992.
  31. R. Milner. *Communication and concurrency*. Prentice-Hall, Englewood Cliffs, NJ, 1989.
  32. X. Nicollin and J. Sifakis. The algebra of timed processes ATP: Theory and applications. *Information and Computation*, 114:131–178, 1994.
  33. X. Nicollin, J. Sifakis, and S. Yovine. Compiling real-time specifications into extended automata. *IEEE Trans. Software Engineering*, 18:794–804, 1992.
  34. X. Nicollin, J. Sifakis, and S. Yovine. From ATP to timed graphs and hybrid systems. In J.W. de Bakker, C. Huizing, and W.P. de Roever, editors, *Real-time: Theory and practice, Proceedings of the REX Workshop, Mook, The Netherlands, June 3-7, 1991*, number 600 in Lecture Notes in Computer Science, pages 45–74, Berlin, 1992. Springer.
  35. X. Nicollin, J. Sifakis, and S. Yovine. From ATP to timed graphs and hybrid systems. *Acta Informatica*, 30:181–202, 1993.
  36. J.S. Ostroff. *Temporal logic for real-time systems*. Research Studies Press Ltd., Taunton, England, 1989.
  37. A. Overkamp. Supervisory control for nondeterministic systems. In G. Cohen and J.-P. Quadrat, editors, *11th International Conference on Analysis and Optimization of Systems - Discrete Event Systems*, number 199 in Lecture Notes in Control and Information Sciences, pages 59–65, London, 1994. Springer-Verlag.
  38. P.J. Ramadge and W.M. Wonham. Supervisory control of a class of discrete event processes. *SIAM J. Control Optim.*, 25:206–230, 1987.
  39. P.J.G. Ramadge and W.M. Wonham. The control of discrete event systems. *Proc. IEEE*, 77:81–98, 1989.
  40. K. Rudie and W.M. Wonham. Protocol verification using discrete-event systems. In *Proceedings of the 31st IEEE Conference on Decision and Control*, pages 3770–3777, New York, 1992. IEEE Press.
  41. K. Rudie and W.M. Wonham. Think globally, act locally: Decentralized supervisory control. *IEEE Trans. Automatic Control*, 37:1692–1708, 1992.
  42. J.A. Stankovic. Misconceptions about real-time computing: A serious problem for next-generation systems. *Computer*, 21, No. 10 (Oct.):10–19, 1988.
  43. J.G. Thistle. *Control of infinite behaviour of discrete-event systems*. PhD thesis, Department of Electrical Engineering, University of Toronto, Toronto, 1991.

44. J.G. Thistle. Logical aspects of control of discrete event systems: A survey of tools and techniques. In G. Cohen and J.-P. Quadrat, editors, *11th International Conference on Analysis and Optimization of Systems - Discrete Event Systems*, number 199 in Lecture Notes in Control and Information Sciences, pages 3–15, London, 1994. Springer-Verlag.
45. K.C. Wong. *Discrete-event control architecture: An algebraic approach*. Systems and control group report, University of Toronto, Toronto, 1994.
46. Howard Wong-Toi and G. Hoffmann. The control of dense real-time discrete event systems. Report STAN-CS-92-1411, Department of Computer Science, Stanford University, Stanford, 1992.
47. W.M. Wonham and P.J. Ramadge. On the supremal controllable sublanguage of a given language. *SIAM J. Control & Opt.*, 25:637–659, 1987.



# Fast, Randomized Join-Order and Join-Method Selection Combined with Transformation Based Optimization <sup>1</sup>

Arjan Pellenkoft  
César Galindo-Legaria  
Martin Kersten

*CWI*

{arjan,cesar,mk}@cwi.nl

Most of the work on randomized query optimization has relied heavily on the use of transformations rules for the generation of execution plans. Recently, however, we gave evidence that for the problem of choosing a join evaluation order, generating alternatives uniformly at random from the space yields solutions comparable to those obtained with transformation-intensive methods, and requires generating fewer candidate plans.

This paper presents a thorough empirical study of the impact of catalogs and join methods on the relative performance of transformation-free and transformation-based randomized optimization. Basically, our previous results remain valid for a wide variety of catalogs and relational profiles. But in contrast with the problem of selecting a join order, selecting join algorithms (e. g. hash, merge, nested-loops) seems better handled by transformations than random picking.

We then propose a two-phase approach that combines the speed of random picking with the quality of solutions of transformation-based optimization, and verify experimentally its superiority over the other algorithms, in all the search spaces considered.

## 1 INTRODUCTION

A major task of relational query optimizers is to select a suitable join evaluation order for which the estimated evaluation cost is minimum [Ul82, CP85, KRB85]. For small queries, exhaustive search is often feasible, but the number

---

<sup>1</sup>To Cor Baayen, at the occasion of his retirement and as a tribute to his choice in 1985 to establish a Database Research group. His visionary goal to improve scientific cooperation is exemplified by the co-author César Galindo-Legaria, one of the few ERCIM fellows.

of join orders increases very fast as the number of relations grow. Heuristics and/or probabilistic algorithms are then a viable alternative. Research on probabilistic algorithms has focused on *Simulated Annealing* (SA) and *Iterative Improvement* (II), and their variations [IW87, SG88, Swa89b, Swa89a, IK90, IK91, LVZ93]. Those optimization algorithms rely heavily on transformation rules to generate alternative join evaluation orders. The transformation rules are usually based on algebraic properties of the join evaluation orders, like *commutativity* and *associativity*, and they impose a particular topology on the search space —namely, evaluation plans are adjacent if they differ by a single application of a transformation rule. But the effect of a given topology on the behavior of search algorithms remains difficult to quantify. This prompted us to examine a transformation free (TF) optimization scheme that generates plans uniformly at random and keeps the best solution generated [GLPK94]. Our finding was that transformations tend to improve solutions “slowly,” and the TF scheme converges faster and finds plans comparable to those found by transformation based optimizers.

The study in [GLPK94] was based on a calibrated cost model for the DBS3 system [ACV91] —a main memory database whose cost model accounts for CPU only— and considered execution plans with only hash-joins. In this paper we extend our previous experiments to assess the stability of the phenomenon observed. We use the same I/O-dominated cost model used at the University of Wisconsin in their randomized optimization work [IK90, Kan91]. We examine the impact of indices, changes on the statistical profiles of the catalogs, and the use of different join algorithms.

For the problem of selecting a join-order, the size of the space is exponential in the number of relations (see [GLPK95] for the exact size). When, in addition, a join algorithm is selected ( $n - 1$   $m$ -ary decisions for a query on  $n$  relations with  $m$  algorithms available), the resulting search space is the product of two exponentially large spaces. So, including the selection of join algorithms has a different effect on the problem than changing the cost model or the catalog profiles. In fact, our current experiments show a qualitative difference in the relative performance of optimization algorithms when different join algorithms are allowed. The “high proportion” of good solutions in the space of evaluation orders is for the most part preserved on different catalogs and cost models, but it decreases in the product space of evaluation orders with method selection. At the same time, the transformations used in this product space seem particularly appropriate and lead to good solutions.

We then study a two-phase approach similar to those of [IK90, LVZ93], using TF in the first phase and then transformations. The behavior of this algorithm combines the fast converge of random picking with the high quality of solutions of transformation-based search, and it is superior to the other algorithms in all the spaces we considered. From the behavior of this hybrid algorithm, it appears that the neighborhood structure around a given plan, from the point of view of the transformation-induced topology, depends mostly on the cost of such plan. That is, a transformation-based search behaves roughly the same



way when started on any two randomly-selected plans of the same cost.

*Road map.* This paper is organized as follows. In Section 2 we give definitions, details on the cost model, and the three basic search algorithms. The testbed for the experiments is described in Section 3. Section 4 describes experiments with various catalogs, and Section 5 examines multiple join algorithms. Finally, Section 6 contains experimental results on hybrid algorithm. Conclusions are given in Section 7.

## 2 DEFINITIONS

This section defines the search space, the basic probabilistic search algorithms used on that space, and the performance measure used for comparing the algorithms.

### 2.1 Search Space

*Query evaluation plans.* We represent a query by means of a *query graph*. Nodes of such graph are labeled by relation names, and edges are labeled by predicates. An edge labeled  $p$  exists between the nodes of two relations, say  $R_i$ ,  $R_j$ , if predicate  $p$  references attributes of  $R_i$ ,  $R_j$ . The *result* of a query graph  $G = (V, E)$  is defined as a Cartesian product followed by relational selection:  $\sigma_{p_1 \wedge \dots \wedge p_n}(R_1 \times \dots \times R_m)$ , where  $\{p_1, \dots, p_n\}$  are the labels of edges  $E$  and  $\{R_1, \dots, R_m\}$  are the labels of nodes  $V$ .

*Query evaluation plans (QEPs)* are used to evaluate queries, instead of the straight definition of product followed by selection given above. A QEP is an operator tree whose inner nodes are labeled by a join operator and whose leaves are labeled by relations. The *result* of a QEP is computed bottom-up in the usual way. QEPs include annotations on the join-algorithm to use — e. g. nested loops, hash, merge, etc. — when several are available.

Not every binary tree on the relations of the query is an appropriate QEP, because some may require the use of Cartesian products. We restrict the search space to those QEPs that do not require products, called *valid* in [SG88]. Some systems restrict the topology of QEPs further, so that each join operates on at least one base relation. Such restriction leads to the space of *linear* QEPs. We do not impose such restriction here, so we work on the more general *bushy* space.

*Tree transformations.* The transformations used to generate new QEPs, where applicable, are the following [IK90, IK91]: Commutativity,  $A \bowtie B \leftrightarrow B \bowtie A$ ; associativity,  $(A \bowtie B) \bowtie C \leftrightarrow A \bowtie (B \bowtie C)$ ; left join exchange,  $(A \bowtie B) \bowtie C \leftrightarrow (A \bowtie C) \bowtie B$ ; right join exchange,  $A \bowtie (B \bowtie C) \leftrightarrow B \bowtie (A \bowtie C)$  and join method selection,  $A \bowtie_{method_i} B \leftrightarrow A \bowtie_{method_j} B$ .

```

PROCEDURE II() {
  minS = infinite; // with cost(infinite) = infinite
  WHILE not (stopping_condition) DO {
    S = random state;
    WHILE not (local_minima(S)) DO {
      S' = random state in neighbors(S);
      if cost(S') < cost(S) THEN S = S';}
    IF cost(S)<cost(minS) then minS = S;}
  return(minS);}

```

FIGURE 1. Iterative Improvement

## 2.2 Search Algorithms

We experiment with three basic search algorithms, the transformation-based Iterative Improvement and Simulated Annealing, and a transformation free algorithm. We summarize them here for completeness. More details on the transformation-based optimizers can be found in a number of references, including [KCV82, NSS86, SG88, IK90, LVZ93].

*Iterative Improvement (II)* performs a large number of *local optimizations*. A local optimization starts at a random QEP, called the *current QEP*. By applying a randomly selected transformation rule to the current QEP a new one is generated. If this is cheaper then it is accepted as current QEP, otherwise it is rejected. A local optimization stops when a local minimum has been reached. The II algorithm stops as soon as a predefined number of plans has been generated. The plan found with the lowest cost is returned as the result. Figure 1 shows the pseudo-code of the II algorithm.

To detect a local minimum the neighbors are not searched exhaustively but a *r-local minimum* is used [Kan91], i.e. a plan is a local minimum if none of *r* randomly selected neighbors has a lower cost. Since the plans are selected at random, and repetitions are possible, a *r-local minimum* is not guaranteed to test all neighbors. In the experiments *r* is set to the number of neighbors of a node.

*Simulated Annealing (SA)*. Sometimes the II algorithm fails to find good plans because it gets stuck in high cost local minima. SA attempts to solve this problem by also accepting new QEPs with a higher cost, with some probability. The SA algorithm starts at a random QEP and randomly generates next QEPs. The probability of accepting QEPs with higher cost decreases as time progresses. When a predefined number of plans has been generated or a “stable condition” has been reached the SA algorithm stops.

```

PROCEDURE SA(){
  S = S0;
  T = T0;
  minS = S;
  WHILE not(frozen) DO {
    WHILE not(equilibrium) DO {
      S' = random state in neighbors(S);
      deltaC = cost(C') - cost(S);
      IF (deltaC <=0) THEN S = S';
      IF (deltaC > 0) THEN S = S' with probability e^(-deltaC/T);
      IF cost(S)<cost(minS) THEN minS = S;}
    T = reduce(T);}
  return(minS)}

```

FIGURE 2. Simulated Annealing

Figure 2 shows the pseudo-code of the SA algorithm. The *frozen* and *equilibrium* conditions used in our experiments are those given in [Kan91].

If time is infinite both transformation based search algorithms will find the global minimum, but in practice the resource available for optimization are limited and must be used as efficiently as possible.

*Transformation Free (TF)*. To remove the reliance on transformation rules, and a potentially slow quality improvement, the TF algorithm was investigated in detail in [GLPK94]. This algorithm generates QEPs uniformly at random, and keeps track of the one with the lowest cost. The algorithm terminates after it has generated a predefined number of QEPs. The QEP with the lowest cost is returned as preferred plan for execution. Like II and SA, if TF is given infinite time it will find the global minimum. But unlike SA and II, if time is finite TFs performance only depends on the cost distribution over the search space and not on the topology imposed on the space by the transformation rules. Figure 3 shows the pseudo-code of the TF algorithm. Note that the random states are chosen *uniformly* from the space. See [GLPK95] for details on how this is achieved.

### 2.3 Performance Measure

The behaviour of an optimization strategy can be represented by a function mapping the number  $n$  of plans explored to the estimated cost of the best plan found. For a given algorithm  $A$ , we call this cost the *solution* after  $n$ , and denote it by  $S_n^A$ . Formally, using  $U_n^A$  as the set of the first  $n$  plans visited by  $A$ , the solution after  $n$  is:

$$S_n^A = \min\{\text{cost}(p) \mid p \in U_n^A\}.$$

```

PROCEDURE TF(){
  minS = random state;
  WHILE not(stop_condition) DO {
    S = random state;
    IF cost(S)<cost(minS) THEN minS = S;}
  return(minS)}

```

FIGURE 3. Transformation Free

For transformation-base algorithm, every valid plan generated by the algorithm is counted as explored, even if it is not accepted by the algorithm (e. g. because its cost is higher than the current plan in II).

Since the algorithms are probabilistic,  $U_n^A$  is a random subset of size  $n$  from the search space, and therefore  $S_n^A$  is a random variable. Based on this, we measure the success of these algorithms using the mean and standard deviation of the solution. As  $n$  increases, the mean of  $S_n^A$  should approach the minimum cost in the search space; while at the same time the standard deviation of  $S_n^A$  approaches zero. The second condition ensures that the algorithm, though probabilistic, behaves in a stable way. Although the number of plans explored does not account for all the resources required by an algorithm, we use this solution after  $n$  as an *implementation-independent* measure of algorithm performance.

### 3 TESTBED

To assess the stability of the TF search algorithm we conducted a large number of experiments with the I/O-based cost model of [Kan91] and queries and catalogs that were also used in our earlier work with the DBS3 cost model [ACV91].

The new cost model is used exhaustive to study the impact of the catalogs and the available join methods on the performance of TF, II, and SA. The queries used in the experiments are randomly generated and acyclic. They range from 4 to 20 joins and all join predicates are equality joins. These queries were optimized for three catalogs with different variance in attribute values and relation size. The queries and catalogs used in [IK91] constitute our starting point and in the sequel of this paper these catalogs will be referred to as the *original* catalogs.

#### 3.1 Cost Model

The cost model called CM2 in [Kan91] is the basis for our experiments. This cost model assumes a disk-based database system. Since the cost of evaluating a QEP is dominated by the I/O, the number of pages that are read or written during the evaluation of a QEP is used as cost metric. A large buffer is assumed

Catalog	Cardinality	Percentage of unique values in attribute
catalog 1	1000	[0.9,1.0]
catalog 2	[1000,100000]	[0.9,1.0]
catalog 3	[1000,100000]	[0.1,1.0]

FIGURE 4. Sizes and selectivities of the *original* catalogs

in the cost model. The major difference with the DBS3 cost model used in our previous work, is that the DBS3 model assumes a main memory database system, in which the CPU cost is the predominant factor.

The CM2 cost model is able to handle three join algorithms, namely *nested-loop*, *merge-scan* and *hash-join*. The cost functions for the nested-loops algorithm are *page-level* nested-loops join and *index-scan* nested-loop. The cost of the cheapest alternative is returned as cost for a nested-loop join. The cost of the merge-scan join consist of sorting the inputs, if they are not already sorted, and by merging the two input streams. The hash-join also has two alternatives of which the one with the cheapest cost is returned. These two alternatives are *simple hash-join* and *hybrid hash-join*. In the computation of the cost for the hash-join it is assumed that the hash table is build on the smallest input.

When an index is available for a join attribute it can be used to reduce the loading cost. The usual assumption is made that the attribute values are uniformly distributed and that the columns values are independent.

### 3.2 Factors Considered

The factors considered in our study are the following:

- Catalog variance (the difference in relation size and join selectivity).
- Relation sizes (original catalogs or enlarged catalogs).
- Indices (present or not).
- Join algorithms (nested-loop, hash-join, merge-scan).

The catalogs used are randomly generated from a profile that specifies an allowed range for relation sizes and uniqueness of attributes. Figure 4 gives the profiles for the three types of catalogs used. For example, a catalog of type 2 (or simply catalog 2) uses relation sizes ranging from 1,000 to 100,000 tuples and the uniqueness of the attribute values range from 90% to 100%. This percentage of unique values is used for the computation of the join selectivity in the cost estimation. The ranges are chosen such that the *variance* in catalog 1 is small, and it is increased in catalogs 2 and 3.

The *enlarged* catalogs are constructed by multiplying the relation sizes in the original catalog (Figure 4) by a hundred. These enlarged catalogs were used to study the impact of the large I/O buffer in the cost model and possible non-linear behaviour of the cost functions. For both the original and the enlarged

catalogs we used two variants; one with many indices and one without indices. They are used to check the hypothesis that indices have a strong effect on the shape of the search space and, therefore, affect the performance of the search algorithms.

In the experiments discussed in Section 4 there is only one join method available for a single QEP. So all of join operators in a QEP are either nested-loop, merge-scan or hash-join. Section 5 and 6 describe experiments in which the plans considered combine different join algorithms.

### 3.3 Performance Characteristics

These graphs showing our results present the average of solutions found by the various algorithms after exploring a given number of plans. As is usual in the work on this subject, the  $y$ -axis is a linear measure of *scaled cost*, with a scaled cost of 1 for the cheapest individual plan found by any algorithm, in the given search space.

These graphs have some properties useful for the comparison of search algorithms. A general description of the graph of TF and II is as follows. Up to a *crossover point* the TF algorithm generates better plans, and after that the II algorithm finds better plans. This crossover point marks the solution that is found by both algorithms after exploring the same number of plans.

After exploring a many plans, the cost of solutions found by probabilistic algorithms improves very slowly. We could say that at some point the optimizer becomes *stable* and call the quality of the plan at that point the *final cost*. The difference in final cost is used to compare algorithms.

Another important characteristic of the graph is the *cost range*. If the difference between the best solution and the worst solution in the search space is small, the optimization has a relatively smaller impact on the execution time of the query. If, on the other hand, the cost range is large, the optimizer can produce a dramatic improvement on query performance.

These three aspects — *crossover point*, *final cost (difference)* and *cost range*— of a performance graph are helpful in analysing the performance of the search strategies. In Figure 5 these aspects are marked in a skeleton performance graph.

## 4 EFFECT OF VARIANCE, INDICES, AND RELATION SIZES

This section discusses the experiments done to determine the circumstances for which the random generation of plans is comparable to the transformation based approach. The sequel of this section describes the behaviour of TF, II and SA for various catalogs and join methods. All graphs shown are averages over a large number of runs.

### 4.1 Catalogs with Indices

The original catalogs with indices are used for our first experiment. As mentioned in Section 3 the optimizers only consider QEPs in which all join algo-

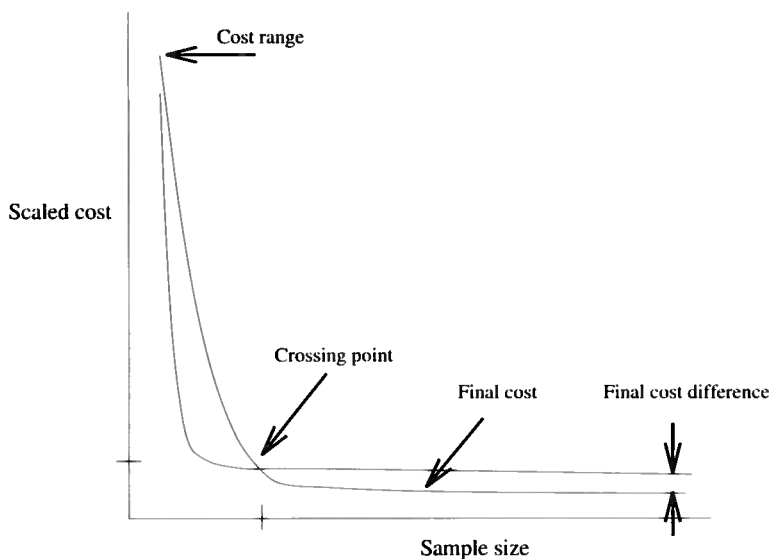


FIGURE 5. Skeleton performance graph

gorithms are either nested-loop, merge-scan or hash-join.

We observed that as the catalogs changed, from low variance to high variance, the *cost range* of the graphs increased and the *crossover point* shifts to the left. The *final cost* of the II and TF algorithm are similar for catalogs 2 and 3. Only for the low-variance catalog 1 the II algorithm is consistently better. For the high-variance catalogs the relative performance TF algorithm is best. Figure 6 illustrates shows the results for a query of 20 joins when only hash-joins are considered (the results for nested-loops and merge-join are similar).

To our surprise the QEPs with only nested-loops join were the cheapest in absolute cost. A closer examination of the QEPs generated showed that the large buffer size, relative to the size of the relations involved, caused this effect. Most of the processing can be done such that the inputs to the join operator fit in the buffer, so the nested loops algorithm does not require any reloads. Due to the overhead cost of the other two algorithms they resulted in more expensive QEPs.

#### 4.2 Catalogs without Indices

We drop all indices in the next round of experiments, to test the assumption that indices reduce the *cost range* and make the search space *smoother*. That is, the *cost difference* with neighbors becomes smaller.

Surprisingly, for the experiments conducted, the performance difference between search algorithms in spaces with indices or without is small. But an interesting change can be observed for the high variance catalogs in Figure 7. Compared to the indexed catalogs the *crossover points* shifted slightly to the

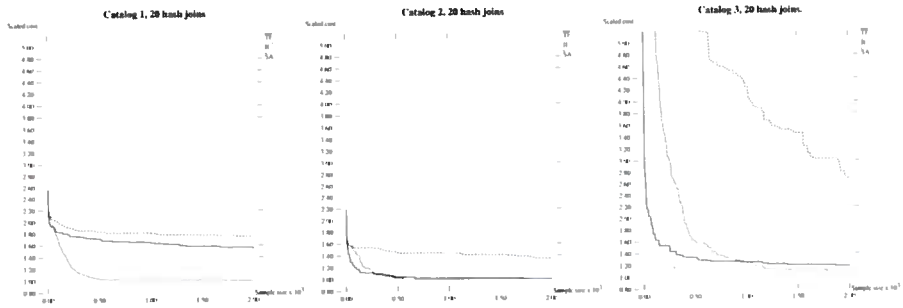


FIGURE 6. Space of hash QEPs for the original catalog with indices

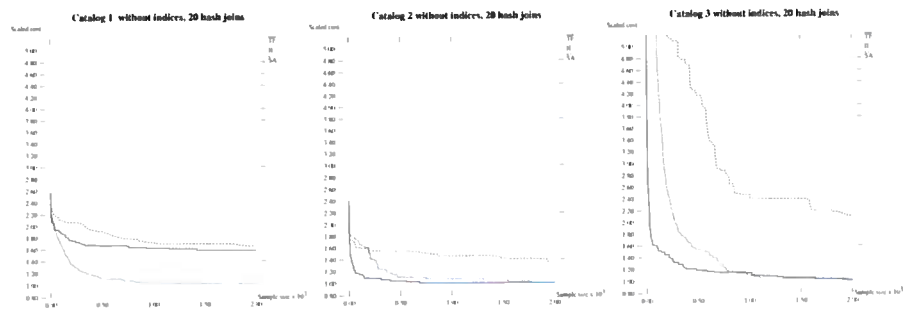


FIGURE 7. Space of hash QEPs for the original catalogs without indices

right for all join methods and the quality of the plans at the *crossover points* is better. The *cost range* of the graphs and the *final costs* are similar to those of the indexed catalogs.

Our experiments, then, lead to the conclusions that although indices have a noticeable impact on optimization performance, it is relatively small compared to the impact of the catalog variances or variance in join selectivity.

### 4.3 Large Catalogs with Indices

To examine the impact of the large buffer on the performance of the search algorithms, we enlarged the relation sizes of the original catalogs. For these big relations the QEPs with only hash-joins were consistently cheaper than QEPs with only merge scan or nested loop. This search space of QEPs, with only hash joins, also showed the biggest change in performance. For catalog 2 TF finds plans much faster than II and also the distance between the graphs has grown in comparison to the original catalog 2. For catalog 3 the TF algorithm



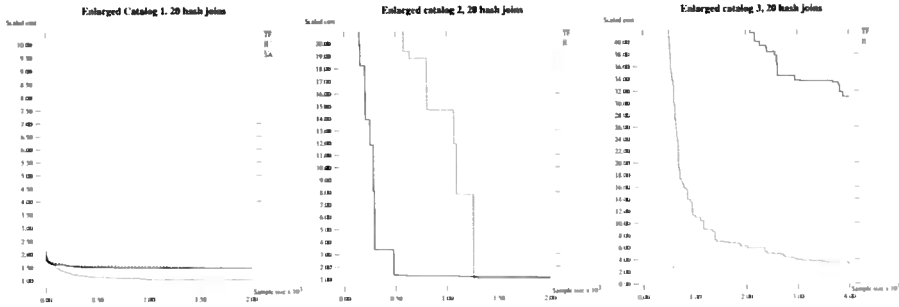


FIGURE 8. Space of hash QEPs for enlarged catalogs with indices

improves faster before the *crossover point*, but this *crossover point* has a high cost.

With the enlarged catalogs both the cost and the difference between final costs has grown. To make the performance graphs of the search algorithms visible, the scale of the  $y$ -axis was enlarged by a factor of ten. In Figure 8 the performance graphs of the search algorithms are given for the tree catalogs when only hash-joins are used.

We also ran experiments for enlarged catalogs *without* indices. The results are basically the same as those presented for the spaces with indices, so they are not shown here.

## 5 EFFECT OF MULTIPLE JOIN ALGORITHMS

We now consider the use of multiple join algorithms in QEPs. To deal with this case in transformation based strategies, a rule is added that changes the algorithm at a specific join operator. Such addition leads to a dramatic growth of the search space. If  $m$  join algorithms are considered and the QEPs joins  $n$  relations, each QEP in the original search space is mapped to  $m^{n-1}$  QEPs with join selection. This big search space seems to contain cheaper QEPs —e. g. a hash-join whose inputs are sorted can be replaced by a merge-scan— but it also introduces many QEPs with higher cost. Important for the performance of all three search algorithms is how the cost distribution changes, and for transformation based optimizers also the modified connectivity of the search space.

Uniformly random generation of elements from the enlarged space is easy —modulo the uniform generation of evaluation orders. Simply select independently and uniformly a join algorithm for each join in the QEP.

Figure 9 shows the performance graphs for the three search methods using all join algorithms. For reference, we also show the result of II and TF on the restricted space of plans that use hash-joins only. The effect of the enlarged

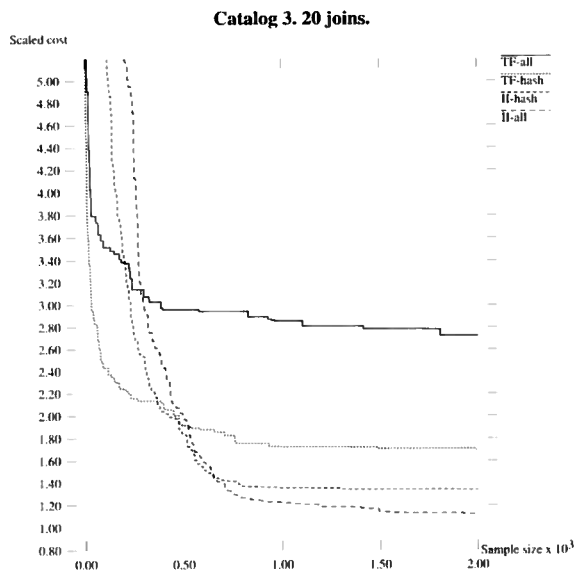


FIGURE 9. Multiple join methods

space is clear from this graph. Initially, both TF and H progress about as quickly in the space restricted to hash-joins as in the more general space. But then TF becomes stable in more costly solutions when it has to select a join algorithm, while H finds better solutions when selecting a join algorithm.

We can conclude that the reduced percentage of good plans in the bigger space has a negative effect on the performance of the TF algorithm. However, the topology imposed by the change-join-algorithm transformations seems particularly appropriate for a transformation-based search.

In the following section we show experiments in which random generation and the use of transformation rules are mixed. Ideally these methods should incorporate the good behaviour of both the TF and H algorithm, fast convergence and good final plans.

## 6 HYBRID ALGORITHMS

Considering all experiments performed, an improvement of transformation based optimizers seems feasible by balancing the generation of random plans with the application of transformations. Other multi-phase optimization schemes have been proposed in [Kan91, LVZ93], but they still rely mainly on transformations to generate alternatives.

It is reasonable to consider starting the search by generating a predefined number of plans (TF-phase), followed by one transformation-based local optimization. During this local optimization phase no new random starting points

```

PROCEDURE SII(n) {
  minS = infinite; // with cost(infinite) = infinite
  WHILE not (stopping_condition) DO {
    S = random state
    FOR i = 1 TO n - 1 DO {
      S' = random state;
      IF cost(S') < cost(S) THEN S = S';}
    WHILE not (local_minima(S)) DO {
      S' = random state in neighbors(S);
      IF cost(S') < cost(S) THEN S = S';}
    IF cost(S) < cost(minS) then minS = S;}
  return(minS);}

```

FIGURE 10. Set-Based Iterative Improvement

are generated. A generalization of this idea is what we call the *Set-based Iterative Improvement* ( $SII_n$ ) algorithm. This hybrid algorithm is an II algorithm that uses the best plan of a randomly generated set as starting state for a local optimization. The  $n$  represents the size of the randomly generated start set. Figure 10 shows the pseudo-code of the algorithm.

Figure 11 shows the performance of  $SII_{100}$ , as well as TF and II for the space of join ordering, when using the enlarged catalog 3. The graph of the  $SII_{100}$  algorithm reflects the behaviour of both the TF and II algorithm. It converges as fast as the TF graph in the first part of the graph and then picks up the behaviour of the II algorithm, resulting in very good quality plans. Figure 11 is typical for the behaviour of the SII algorithm.

Figure 12 shows the performance of  $SII_{100}$  on the space of join ordering plus join-algorithm selection, also in combination with enlarged catalog 3. Although the TF algorithm has a weak performance for this search space, the  $SII_{100}$  algorithm maintains its good behaviour.

## 7 CONCLUSIONS

In this paper we examined the impact of several factors on the performance of probabilistic query optimization algorithms, in particular the relative behavior of random picking of solutions with respect to transformation-based search. The results of random picking give a direct indication of the proportion of good solution in the search space, while the transformation-based search also depends on the topology imposed by the specific set of transformations used.

Our experiments show that the results obtained in [GLPK94] for a main-memory database remain valid, for the most part, when the I/O-based cost model of [IK90, Kan91] is used instead. A transformation-free algorithm finds good plans faster than a transformation-based approach, but the

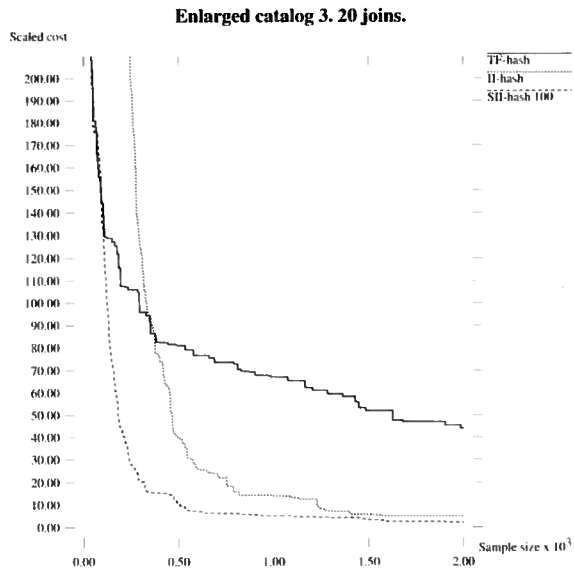


FIGURE 11. Hybrid search on the restricted space of hash-joins

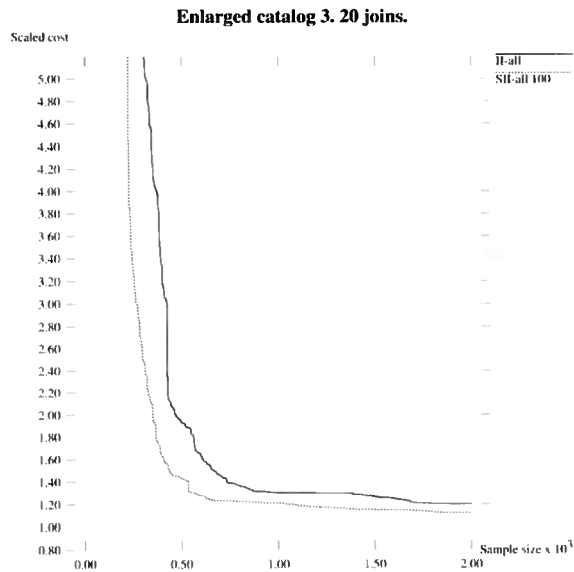


FIGURE 12. Hybrid search using all available join algorithms

transformation-based search finds the best plans in the end. This happens because the ratio of good plans is substantial and the topology imposed by associativity/commutativity/exchange transformations does not seem to aid the search significantly, especially at the beginning of the process. We observed that the presence of indices does not reshape the search space, and affects only marginally the performance of all the search methods.

We then studied the effect of selecting a join algorithm, in addition to a join evaluation order. In this case the search space becomes the product of two exponentially large spaces, and its properties turn out to be qualitatively different from those of selection of a join order evaluation alone. The proportion of good plans decreases in this combined space, and at the same time the topology induced by the change-algorithm rule seems to favor the transformation-based search.

Finally, we described and tested a two-phase optimization approach that starts with random picking to generate good plans quickly, and then applies transformations for further refinement. The result is a combination of the best of both search strategies: fast convergence to solutions of very high quality. We believe this hybrid approach is basically the best alternative in a *purely stochastic search*—i. e. one that does not consider heuristics—probably with an additional Simulated Annealing phase at the end as suggested in [IK90].

There are related issues that remain to be addressed. The first is how to incorporate heuristics in a robust manner. In our view, the use of heuristics in randomized search must be that of “rigging the odds” in favor of the better plans. We are in the process of formulating the necessary framework. Also, the two specific spaces identified in this paper on which the transformation-based and transformation-free schemes behave significantly differently provide a test case for the study of when and how are transformations advantageous for optimization.

*Acknowledgements.* To conduct the experiments reported on this paper, we coded the uniformly-distributed generation of join trees, and the TF and hybrid algorithms on top of the code for randomized query optimization developed at the University of Wisconsin [IK90, Kan91]. We are grateful to Yannis Ioannidis for kindly providing us with a copy of their software, and for allowing us to modify it for our experiments.

#### REFERENCES

- [ACV91] F. Andrès, M. Couprie, and Y. Viémont. A multi-environment cost evaluator for parallel database systems. *Proceedings of the 2nd Int. DASFAA Japan*, 1991.
- [CP85] S. Ceri and G. Pelagatti. *Distributed Databases: Principles and Systems*. McGraw-Hill, New York, 1985.

- [GLPK94] C. A. Galindo-Legaria, A. Pellenkoft, and M. L. Kersten. Fast, randomized join-order selection —Why use transformations? In *Proceedings of the Twentieth International Conference on Very Large Databases, Santiago*, 1994. Also CWI Technical Report CS-R9416.
- [GLPK95] C. A. Galindo-Legaria, A. Pellenkoft, and M. L. Kersten. Uniformly-distributed random generation of join orders. In *Proceedings of the International Conference on Database Theory, Prague*, 1995. Also CWI Technical Report CS-R9431.
- [IK90] Y. E. Ioannidis and Y. C. Kang. Randomized algorithms for optimizing large join queries. *Proc. of the ACM-SIGMOD Conference on Management of Data*, pages 312–321, 1990.
- [IK91] Y. E. Ioannidis and Y. C. Kang. Left-deep vs. bushy trees: An analysis of strategy spaces and its implications for query optimization. *Proc. of the ACM-SIGMOD Conference on Management of Data*, pages 168–177, 1991.
- [IW87] Y. E. Ioannidis and E. Wong. Query optimization by simulated annealing. *Proc. of the ACM-SIGMOD Conference on Management of Data*, pages 9–22, 1987.
- [Kan91] Y. C. Kang. *Randomized Algorithms for Query Optimization*. PhD thesis, University of Wisconsin-Madison, 1991. Technical report #1053.
- [KCV82] S. Kirkpatrick, C.D. Gelatt, Jr., and M.P. Vecchi. Optimization by simulated annealing. Technical Report RC 9355, IBM Thomas J. Watson Research Center, Yorktown, 1982.
- [KRB85] W. Kim, D. S. Reiner, and D. S. Batory, editors. *Query processing in database systems*. Springer, Berlin, 1985.
- [LVZ93] R. S. G. Lanzelotte, P. Valduriez, and M. Zaït. On the effectiveness of optimization search strategies for parallel execution spaces. *Proc. of the 19th VLDB Conference, Dublin, Ireland*, pages 493–504, 1993.
- [NSS86] S. Nahar, S. Sahni, and E. Shragowitz. Simulated annealing and combinatorial optimization. *23rd Design Automation Conference*, pages 293–299, 1986.
- [SG88] A. N. Swami and A. Gupta. Optimization of large join queries. *Proc. of the ACM-SIGMOD Conference on Management of Data*, pages 8–17, 1988.
- [Swa89a] A. N. Swami. *Optimization of Large Join Queries*. PhD thesis, Stanford University, 1989. Technical report STAN-CS-89-1262.
- [Swa89b] A. N. Swami. Optimization of large join queries: Combining heuristics and combinatorial techniques. *Proc. of the ACM-SIGMOD Conference on Management of Data*, pages 367–376, 1989.
- [Ull82] J. D. Ullman. *Principles of Database Systems*. Computer Science Press, Rockville, MD, 2nd edition, 1982.

# Job scheduling on a parallel shared memory bus computer

*Dedicated to Cor Baayen, with esteem and admiration*

H.J.J. te Riele

*Centrum voor Wiskunde en Informatica, Kruislaan 413,  
1098 SJ Amsterdam, The Netherlands.*

The advent of vector computers in the beginning of the eighties, and of parallel computers a few years later has triggered the development of new algorithms, especially tailored to these new architectures. One of the many initiatives of Cor Baayen was the stimulation of research activities in this new field and the provision of the necessary equipment, both at CWI and at SARA.

In this paper we study a problem which is typical for these new developments, namely the scheduling of jobs on *bus*-type parallel computers. The processing elements of such computers communicate with a common memory through channels called buses. Usually, there are less buses than processing elements, so that several processing elements have to share the same bus. It is a consequence of this restriction, as we show in this paper, that the *total processing time* of a parallel job may depend on the *order of execution* of the communication parts of the different subjobs. Unfortunately, this order of execution can *not*, in general, be influenced by the programmer. Therefore, this phenomenon must be accepted as an inherent uncertainty in the timing and reproducibility of jobs on parallel bus-type computers.

*AMS Subject Classification (1991):* Primary 69C12; Secondary 69D51

*CR Subject Classification (1991):* B.4.3, C.1.2

*Keywords & Phrases:* Bus-type parallel computers

## 1 INTRODUCTION

Consider a parallel bus - type computer with a shared memory having  $p * b$  processing elements (PEs), where  $b$  is the number of buses and  $p$  is the number of PEs per bus; see Figure 1.

Apart from the main shared memory, each PE has its own small local memory, called *cache*. It is important to re-use cache data as much as possible, in order to minimize transport of data between the cache and the main memory. Processing elements which share the same bus cannot send/receive data

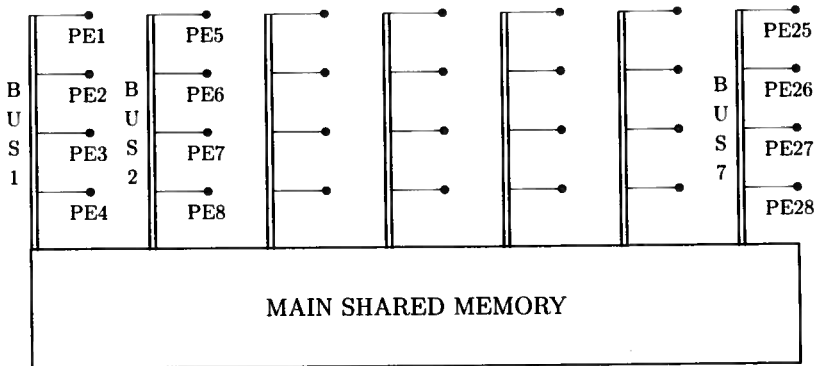
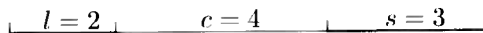


FIGURE 1. Parallel bus – type shared memory computer,  $b = 7$ ,  $p = 4$

to/from the main memory simultaneously. CWI has at least two computers of this kind, viz., the Cray S-MP ( $b = 7$ ,  $p = 4$ ; each PE has a data cache of 8 Kbytes; the size of the shared memory is 256 Mbytes) and the SGI Challenge ( $b = 1$ ,  $p = 4$ ; each PE has two data caches: a primary cache of 16 Kbytes, and a secondary (slightly slower) cache of 1 Mbytes; the shared memory has a size of 256 Mbytes).

We make the simplifying assumption that a job for our parallel shared memory bus computer can be split up in  $S$  equal subjobs. Not many real-life application jobs satisfy this condition, but basic building blocks like matrix-vector multiplication do. Each subjob consists of one part where data are *loaded* from the shared memory into the cache, a second part where *computations* are done with these data, and a final part where the results are *stored* from the cache into the shared memory. The times (in seconds) for these three parts are denoted by  $l$ ,  $c$ , and  $s$ , respectively. Schematically, we will represent a subjob as



where the lengths of the line segments have the ratio  $l : c : s$ . Communication parts ( $l$  and  $s$ ) are marked by thick lines.

The total time  $T$  of a job depends on  $b$ ,  $p$ ,  $S$ ,  $l$ ,  $c$ , and  $s$ , so

$$T = T(b, p, S, l, c, s).$$

By  $\underline{T}(b, p, S, l, c, s)$  we will denote the *minimal* time needed to complete the job. In general, it is easy to give upper and lower bounds for  $T$ . For example,

$$p(l + s) \leq T(1, p, p, l, c, s) \leq p(l + s) + c.$$

The lower bound just counts all the communication times (neglecting the computing times), and for the upper bound we assume that after the last PE has



loaded its data, the computing parts of all the subjobs are carried out simultaneously.

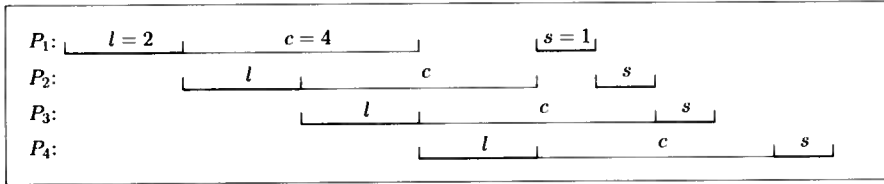


FIGURE 2. Job schedule with  $T(1, 4, 4, 2, 4, 1) = 13$

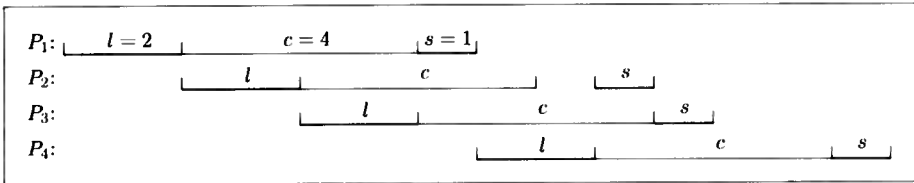


FIGURE 3. Job schedule with  $T(1, 4, 4, 2, 4, 1) = 14$

Figures 2 and 3 illustrate for the case  $b = 1$ ,  $p = 4$ ,  $S = 4$ ,  $l = 2$ ,  $c = 4$ ,  $s = 1$ , that  $T$  may depend on the *order* by which the different PEs execute their communicating parts. In the schedule of Figure 2, processing element  $P_1$  only starts with storing the data into the shared memory after all the PEs have loaded their data. In the schedule of Figure 3 processing element  $P_1$  starts with storing the data as soon as it has completed its computing part (and the bus channel is free). Consequently, we find  $T = 13$  and  $T = 14$ , respectively.

In this paper, we shall analyze the case  $b = 1$  in Section 2, and partly generalize this in Section 3. We present theorems which give the minimum times needed to execute a job on a bus-type parallel computer, under the assumption that the total job can be split up into a number of equal subjobs. Proofs will appear elsewhere, but no doubt the reader will be able to construct some of them without too much effort.

## 2 THE CASE $b = 1$

We start by assuming  $S = p$ , and give three examples with  $p = S = 4$ , viz.,  $l = 1$ ,  $c = 2$ ,  $s = 2$  (Figure 4),  $l = 2$ ,  $c = 2$ ,  $s = 1$  (Figure 5), and  $l = 1$ ,  $c = 4$ ,  $s = 2$  (Figure 6).

These examples suggest that in some cases  $\underline{T}(b, p, S, l, c, s) = \underline{T}(b, p, S, s, c, l)$ , i.e., that the time  $T$  remains fixed, if we interchange  $l$  and  $s$ . Define

$$\underline{m} = \min(l, s), \quad \text{and} \quad \overline{m} = \max(l, s),$$

then we have the following

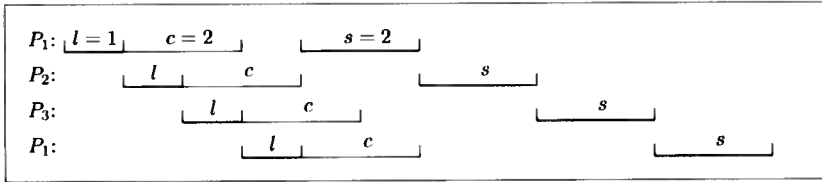


FIGURE 4. Job schedule with  $\underline{T}(1, 4, 4, 1, 2, 2) = 4(l + s) = 12$

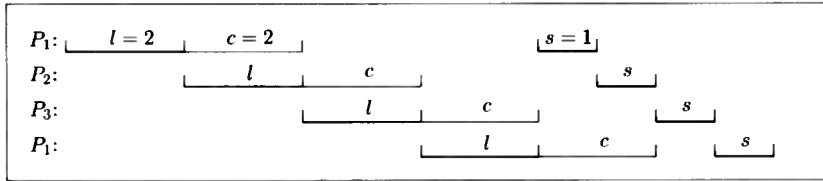


FIGURE 5. Job schedule with  $\underline{T}(1, 4, 4, 2, 2, 1) = 4(l + s) = 12$

**THEOREM 1** Let  $b = 1$  and  $S = p$ ;

- i. if  $c \leq (p - 1)\underline{m}$ , then  $\underline{T} = p(\underline{m} + \overline{m}) = p(l + s)$ ;
- ii. if  $c > (p - 1)\underline{m}$ , then  $\underline{T} = \underline{m} + c + p\overline{m}$ .

Figures 4 and 5 correspond to Theorem 1.i and Figures 2 and 6 correspond to Theorem 1.ii.

The next case we consider is  $S = k * p$  for some integer  $k \geq 2$ . In that case, each processing element will execute  $k$  subjobs. One possibility is that the schedule of Theorem 1 is just repeated  $k$  times, so that the total time is:  $kp(\underline{m} + \overline{m})$  if  $c \leq (p - 1)\underline{m}$ , and  $k(\underline{m} + c + p\overline{m})$  if  $c > (p - 1)\underline{m}$ . However, it turns out to be more efficient in general, if a PE continues with the loading part of the next subjob, as soon as the storage part of its previous subjob has been finished. This concentrates the communication parts of the work done by one PE, and therefore gives more freedom to carry out the computing parts in between them. An example with  $b = 1, p = 4, k = 2(S = 8), l = 1, c = 4, s = 2$  is given in Figure 7. Counting from the end of the job back to the beginning we find that

$$T(1, 4, 8, 1, 4, 2) = 4s + 4(l + s) + c + l = 25$$

(vs.  $T = 26$  if we repeat the schedule of Theorem 1.ii two times). Notice that if we fix the schedule of the load and storage parts, the first computing task

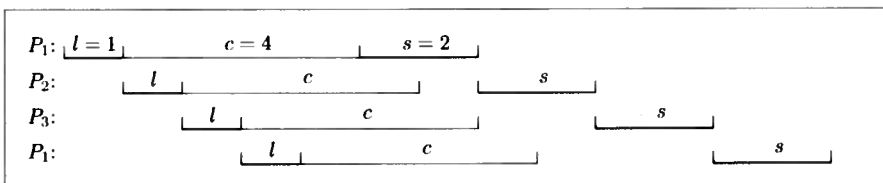


FIGURE 6. Job schedule with  $\underline{T}(1, 4, 4, 1, 4, 2) = l + c + 4s = 13$

of processing elements  $P_2$ ,  $P_3$ , and  $P_4$  could have been scheduled somewhat later, and the second computing task of *all* the four PEs could also have been scheduled somewhat later, without effect on the total computing time  $T$ .

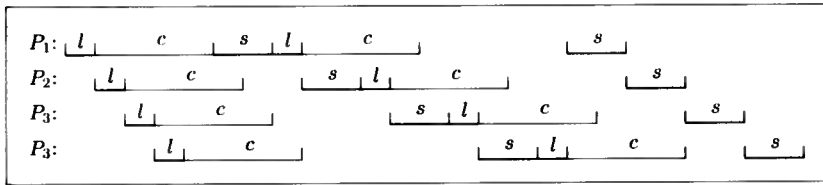


FIGURE 7. Job schedule with  $T(1, 4, 2 * 4, 1, 4, 2) = 25$

We have the following

**THEOREM 2** Let  $b = 1$  and  $S = k * p$  for some integer  $k \geq 2$ ;

- i. if  $c \leq (p - 1)\underline{m}$ , then  $\underline{T} = kp(\underline{m} + \overline{m})$  ;
- ii. if  $(p - 1)\underline{m} < c \leq (p - 1)\overline{m}$ , then  $\underline{T} = c + \underline{m} + p\overline{m} + (k - 1)p(\underline{m} + \overline{m})$ ;
- iii. if  $(p - 1)\overline{m} < c \leq (p - 1)(\underline{m} + \overline{m})$ , then  $\underline{T} = 2c + \underline{m} + \overline{m} + (k - 1)p(\underline{m} + \overline{m})$ ;
- iv. if  $(p - 1)(\underline{m} + \overline{m}) < c$ , then  $\underline{T} = k(c + \underline{m} + \overline{m}) + (p - 1)(\underline{m} + \overline{m})$ .

Figure 7 corresponds to Theorem 2.ii: we find

$$\underline{T}(1, 4, 8, 1, 4, 2) = 4 + 1 + 4 \cdot 2 + (2 - 1)4(1 + 2) = 25,$$

so the schedule of Figure 7 yields the minimal time. To further illustrate this theorem, we consider case iv., and compare its time with that obtained by just repeating Theorem 1.ii  $k$  times. We find, assuming that  $(p - 1)(\underline{m} + \overline{m}) < c$ ,

$$\frac{\underline{T}_{\text{Thm2.iv}}}{\underline{T}_{\text{Thm1.ii}}} = \frac{k(c + \underline{m} + \overline{m}) + (p - 1)(\underline{m} + \overline{m})}{k(c + \underline{m} + p\overline{m})} \rightarrow \frac{c + \underline{m} + \overline{m}}{c + \underline{m} + p\overline{m}}, \text{ as } k \rightarrow \infty.$$

For example, for  $p = 4$ ,  $c = 20$ ,  $l = 1$ ,  $s = 2$  this gives

$$\frac{\underline{T}_{\text{Thm2.iv}}}{\underline{T}_{\text{Thm1.ii}}} = \frac{23k + 9}{29k} \rightarrow \frac{23}{29} = 0.793, \text{ as } k \rightarrow \infty.$$

Now we study, for another example, how the total time  $T$  depends on  $c$ , if the other parameters are kept fixed. Assume  $b = 1$ ,  $p = 4$ ,  $k = 10$  ( $S = 40$ ),  $\underline{m} = 1$ ,  $\overline{m} = 2$ . If we simply repeat Theorem 1 ten times, we find that  $T = 120$  if  $c \leq 3$  and  $T = 10c + 90$  if  $c > 3$ . Theorem 2 gives the *minimum* times, with  $\underline{T} = 120$  for  $c \leq 3$ ,  $\underline{T} = c + 117$  for  $3 < c \leq 6$ ,  $\underline{T} = 2c + 111$  for  $6 < c \leq 9$ , and  $\underline{T} = 10c + 39$  for  $c > 9$ . This is represented graphically in Figure 8. It follows that an efficiency-loss of nearly 40% is possible (for  $c = 9$  we have a worst/best times ratio of  $180/129 \approx 1.40$ ).

### 3 THE GENERAL CASE

For the general case for  $b$  we present two theorems. In the next theorem, we assume that subjobs on different buses update different parts of the main

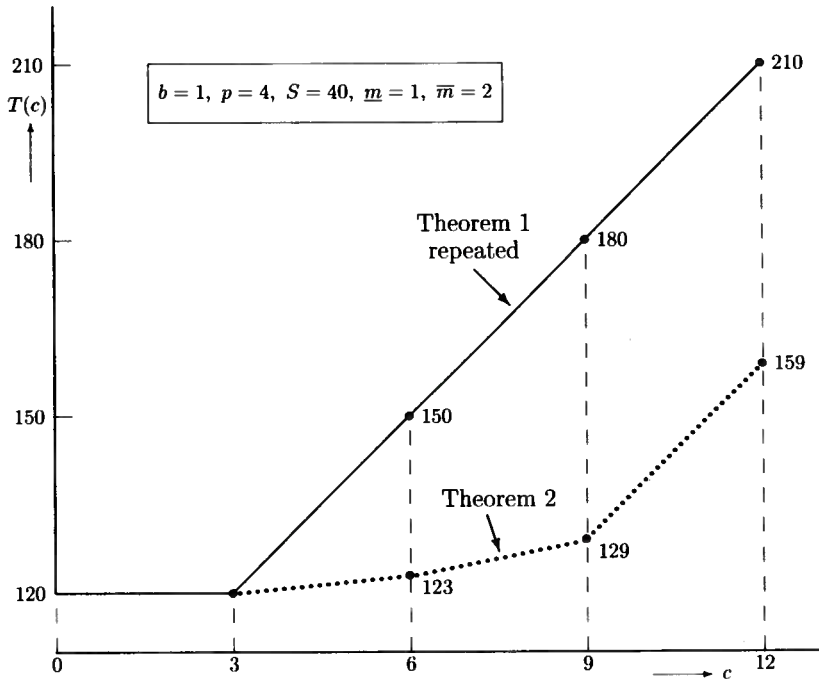


FIGURE 8. Total time  $T(c)$  obtained with Theorem 1 (drawn line) and Theorem 2 (dotted line)

memory, so they can update the main memory at the same time. Moreover, we restrict ourselves to the condition of case iv. in Theorem 2.

**THEOREM 3** Let  $(p - 1)(\underline{m} + \overline{m}) < c$ ,  $S = k_1pb + k_2$  with  $0 \leq k_2 < pb$ , and  $k_2 = k_3b + k_4$  with  $0 \leq k_4 < b$ ; assume that subjobs on different buses update different parts of the main memory.

- i. If  $k_2 = 0$ , then  $\underline{T} = k_1(c + \underline{m} + \overline{m}) + (p - 1)(\underline{m} + \overline{m})$ ;
- ii. if  $k_2 \neq 0$  and  $k_4 = 0$ , then  $\underline{T} = (k_1 + 1)(c + \underline{m} + \overline{m}) + (k_3 - 1)(\underline{m} + \overline{m})$ ;
- iii. if  $k_2 \neq 0$  and  $k_4 \neq 0$ , then  $\underline{T} = (k_1 + 1)(c + \underline{m} + \overline{m}) + k_3(\underline{m} + \overline{m})$ .

In our final theorem we assume that subjobs on different buses *not necessarily* update different parts of the main memory; this means that if one PE is updating the main memory, all the others can not (neither those connected to the same bus, nor those connected to other buses).

**THEOREM 4** Let  $S = p * b$ , so we have precisely one subjob for each PE; assume that if one PE updates the main memory, the others can not.

- i. If  $c \leq (p - 1)\underline{m}$ , then  $\underline{T} = pl + pbs$ ;
- ii. if  $c > (p - 1)\underline{m}$  and  $l \leq s$ , then  $\underline{T} = l + c + pbs$ ;
- iii. if  $c > (p - 1)\underline{m}$  and  $l > s$ , then  $\underline{T} = l + c + pbs + (p - 1)(l - s)$ .

#### 4 CONCLUSION

We have shown that the order of execution of communication parts of subjobs on a parallel shared memory bus-type computer can influence the total processing time of a parallel job unfavourably. Since, in general, the programmer can *not* influence this order of execution, this phenomenon must be accepted as an inherent uncertainty in parallel processing. Examples illustrate that an efficiency-loss of 40% is not uncommon.



# Rambling along paths, trees, flows, curves, knots, and rails

Alexander Schrijver

*CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands and Department of Mathematics,  
University of Amsterdam, Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands.*

As Professor of Mathematics at the Free University, Cor Baayen was an inspiring teacher. His lectures were lucid and skilful, and his broad knowledge enabled him to exhibit the students unexpected vistas and panoramas through several areas in mathematics and theoretical computer science, with topology, set theory, discrete mathematics, logic, and computability as landmarks. As a student you learned that everything is related to everything.

Another characteristic of Cor Baayen's lectures was that he always was eager to present courses on 'modern' topics in mathematics — modern in the sense of not belonging to the standard student curriculum in mathematics (many still don't belong to it). Thus we learned about boolean algebras, graphs, modal logic, proof theory, recursion theory, computability, etc. At the same time there was a strong interest in the historical side of the results discussed.

The courses of Cor Baayen (and his oral examinations, which generally outgrew to private lessons of at least three hours) being stimulating, he added a personal touch by inviting students from their first year at his home, for further metamathematical background. He has stimulated the enthusiasm of several students for mathematics and for doing research.

I think it appropriate not to restrict myself in this paper to one area, but rather to try to link some of the areas of Cor Baayen's interest, by a ramble through topology, discrete mathematics, and algorithmics, with due attention to the historical roots and to some connections with a few of the other interests of Cor Baayen.

**1. Roots of topology.** It seems that Leibniz was one of the first interested in topology, or what he called *geometria situs*. In 1679 he wrote in a letter to Christiaan Huygens:

... mais apres tous les progres que j'ay faits en ces matieres, je ne suis pas encor content de l'Algebre, en ce qu'elle ne donne ny les plus courtes voyes, ny les plus belles constructions de Geometrie. C'est pourquoy lorsqu'il s'agit de cela, je croy qu'il nous faut encor une autre analyse proprement geometrique ou lineaire qui nous exprime directement *situm*, comme l'Algebre exprime *magnitudinem*. Et je croy d'en voir le moyen et qu'on pourrait représenter des figures et mesme des machines et mouve-

mens en caracteres, comme l'Algebre represente les nombres ou grandeurs:  
et je vous envoie un *essay* qui me paroist considerable.

According to Listing, in his *Vorstudien zur Topologie* of 1847 [37], this was the first idea of a scientific and 'calculatory' elaboration of the modal side of the geometry,

... in welchen von einer Art Algorithmus die Rede ist, womit man die Lage räumlicher Gebilde eben so der Analyse unterwerfen müsste, wie es hinsichtlich der Grösse mittelst der Algebra geschieht.

(The essay referred to by Leibniz is following Listing not of 'eigentlich modalen Inhalts'.)

Listing also mentions work by Euler and others on 'die bekannte Aufgabe des sogenannten Rösselsprungs', by Vandermonde on the route by which a thread should go in order to represent for instance a braid or a garter of the weave of a stocking, and by Clausen on the smallest number of penstrokes with which a given figure can be drawn.

Listing, a student of Gauss, says that except for this, the modal side of geometry has 'to expect its elaboration and development almost completely from the future'. As reasons for the fact that since Leibniz not much has been done on the topic, Listing mentions the complexity of discovering effective methods to reduce spatial intuition to concepts, and the inadequacy of language for describing scientifically these, often highly entangled, concepts.

Listing does not claim that he had performed this hard job, and therefore he calls his treatise *Vorstudien zur Topologie*, thereby coining the name *topology*:

Es mag erlaubt sein, für diese Art Untersuchungen räumlicher Gebilde den Namen "Topologie" zu gebrauchen statt der von Leibniz vorgeschlagenen Benennung "geometria situs", welche an den Begriff des Masses, der hier ganz untergeordnet ist, erinnert, und mit dem bereits für eine andere Art geometrischer Betrachtungen gebräuchlich gewordenen Namen "géométrie de position" collidirt. Unter der *Topologie* soll also die Lehre von den modalen Verhältnissen räumlicher Gebilde verstanden werden, oder von den Gesetzen des Zusammenhangs, der gegenseitigen Lage und der Aufeinanderfolge von Punkten, Linien, Flächen, Körpern und ihren Theilen oder ihren Aggregaten im Raume, abgesehen von den Mass- und Grössenverhältnissen.

Listing discusses how several spatial configurations could be represented by a calculus. In particular he focuses on the orientation of objects, and on how one can use his observations when looking through the micro- or telescope, especially when also mirrors are involved. Moreover, he considers dextro- and laevorotation of screws, springs, ropes, spiral staircases, snail's shells, and stalks.

Listing finds that it is difficult to describe the orientation of objects by words, claiming the inadequacy of the description of dextro- and laevorotatory in Linnaeus' *Philosophia Botanica* (1751):



Den Ausdruck *caulis volubilis* nämlich erklärt Linné so: *spiraliter ascendens per ramum alienum* und zwar *sinistrorsum* (⊖) *secundum solem vulgo*, e.g. *Humulus*, *Lonicera* cet.; *dextrorsum* (⊕) *contra motum solis vulgi* e.g. *Convolvulus*, *Phaseolus*, cet. Bei der *Intorsio* wiederholt er diese Bestimmung und stellt sie mit den Windungstypen am *Cirrhus*, an der *Corolla* und anderen Organen zusammen. In einer Anmerkung hierzu gibt nun Linné seine Definition von *sinistrorsum* und *dextrorsum*, welche später — zum Theil aus Anlass des dabei vorgefallenen Druckfehlers — die verschiedensten Exegesen erfahren hat. Linné setzt fest: *sinistrorsum hoc est, quod respicit dextram, si ponas Te ipsum, in centro constitutum, meridiem adspicere; dextrorsum itaque contrarium*, und erklärt damit, dass er die nach der *rechten* Seite eines im Centrum stehende Beobachters hervorragenden Blumenblätter als Kennzeichen einer *links* gewundenen *Corolla* angesehen wissen wolle, und *vice versa*. Das *meridiem adspicere* ist in der concreten Sprache Linné's nicht sowohl ein überflüssiger, als vielmehr ein prägnanter Ausdruck für die aufrechte Stellung des mitten in der Blume gedachten Beobachters, der das Gesicht nach einem bestimmten Punkte des Horizonts kehren soll — versteht sich, den Scheitel nach oben gerichtet. Freilich bleibt bei diesen Erklärungen in topologischer Hinsicht manches zu ergänzen, manches zu fragen übrig.



Studying orientation brings Listing to knots. (A *knot* is a simple closed curve in  $\mathbb{R}^3$ .) They were considered before by Gauss in computing inductance in a system of linked circular wires. Listing introduced a (now standard) planar representation of crossings, as in Figure 1.

Eine Kreuzung dieser Art, wobei sich nach angegebener Weise in der Projection oder Zeichnung der überliegende von dem untenliegenden Faden durch den blossen Anblick leicht unterscheiden lässt, nennen wir eine *Ueberebkreuzung* im Gegensatz zur *Durchkreuzung*, wo ein wirklicher Durchschrittpunkt im Raume stattfindet, und die eben gedachte Entfernung beider Fäden bei  $K$  entweder Null ist, oder wenigstens als verschwindend betrachtet wird. Zwei Wege können demnach, wie beim gewöhnlichen Kreuzwege, einander durchkreuzen, oder aber, wie diess in manchen Städten und bei vielen Kreuzungen zwischen Eisenbahnen und anderen Fahrstrassen der Fall ist, einander überkreuzen.

He also introduces a calculus with  $\lambda$  (for laetotrop) and  $\delta$  (for dexiotrop) indicating the corners at the crossing as in Figure 2, claiming that this signing will facilitate an algorithmic discussion ('wie sie ihres Ortes geführt werden muss') of the equivalence of knots.



Without proof Listing states that the number of crossings in the trefoil knots (Figure 3) cannot be decreased, and that the two knots in the figure are not equivalent.

In particular, Listing was interested in knots in which each face of the projection is 'monotypic' — that is, contains either only  $\lambda$  or only  $\delta$ . Such knots are



Figure 3

with 3 edges each, and 2  $\lambda$ -faces with 2 edges each.

Clearly, the  $\lambda\delta$  type-symbol is an invariant under the *trivial operations* on the diagram: rerouting an edge through the unbounded face, and mirroring the diagram, while interchanging ‘up’ and ‘down’ at each crossing.

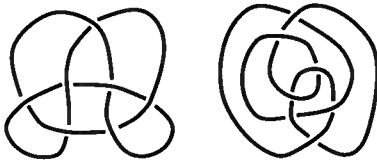


Figure 4

Listing realizes that the  $\lambda\delta$  type-symbol does not give an invariant for alternating knots — he gives an example of two equivalent alternating knots (Figure 4) that have different  $\lambda\delta$  type-symbols.

Interesting is that Listing mentions as one of the further applications of topology, beside natural sciences and art, also the area of industrial mechanics, for which Listing refers to the work of the computer pioneer Charles Babbage [4] on representing machine movements by symbols.

**2. Tait and knots.** Independently of Listing, P.G. Tait studied knots. He was interested in knots because of the ‘vortex atom’ model invented by his friend, the physicist W. Thomson (later Lord Kelvin), like Tait of Scottish origin.

Tait had a broad scientific interest in mathematics, physics and other disciplines, and published papers and notes on electrodynamics, magnetism, the molecular arrangement in crystals, determinants, quaternions, thermodynamics, the value of the Edinburgh Degree of M.A., the fecundity and fertility of women, earth rotation, comets, fluid dynamics, partial differential equations, spectral analysis, thermoelectricity, the retina, the pendulum motion, combinatorics, viscosity, integral calculus, sound and music, the double rainbow, thunderstorms, and the pace of a golf ball.

Studies of curves in the plane led him to investigating the four-colour problem, and he also applied them to knots. In a paper presented to the British Association in 1876, Tait [66] observed that the cells of a plane closed curve can be coloured black and white so that adjacent cells have different colours. He finishes by remarking:

The development of this subject promises absolutely endless work — but work of a very interesting and useful kind — because it is intimately connected with the theory of knots, which (especially applied in Sir W. Thomson’s Theory of *Vortex Atoms*) is likely soon to become an important branch of mathematics.

In the theory of ‘vortex atoms’ of Thomson [72], the internal coherence of atoms was assumed to be determined by a knot, or rather a link (a disjoint union of

now known as *alternating knots* — indeed, when following the knot one goes alternately over and under. The *type-symbol* assigned to such knots is for instance  $\delta^5 + 3\delta^3, \lambda^4 + 2\lambda^3 + 2\lambda^2$ , indicating that there is 1  $\delta$ -face with 5 edges, 3  $\delta$ -faces with 3 edges each, 1  $\lambda$ -face with 4 crossings, 2  $\lambda$ -faces

Listing realizes that the  $\lambda\delta$  type-symbol does not give an invariant for alternating knots — he gives an example of two equivalent alternating knots (Figure 4) that have different  $\lambda\delta$  type-symbols.

Interesting is that Listing mentions as one of the further applications of topology, beside natural sciences and art, also the area of industrial mechanics, for which Listing refers to the work of the computer pioneer Charles Babbage [4] on representing machine movements by symbols.

**2. Tait and knots.** Independently of Listing, P.G. Tait studied knots. He was interested in knots because of the ‘vortex atom’ model invented by his friend, the physicist W. Thomson (later Lord Kelvin), like Tait of Scottish origin.

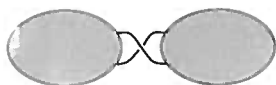
Tait had a broad scientific interest in mathematics, physics and other disciplines, and published papers and notes on electrodynamics, magnetism, the molecular arrangement in crystals, determinants, quaternions, thermodynamics, the value of the Edinburgh Degree of M.A., the fecundity and fertility of women, earth rotation, comets, fluid dynamics, partial differential equations, spectral analysis, thermoelectricity, the retina, the pendulum motion, combinatorics, viscosity, integral calculus, sound and music, the double rainbow, thunderstorms, and the pace of a golf ball.

Studies of curves in the plane led him to investigating the four-colour problem, and he also applied them to knots. In a paper presented to the British Association in 1876, Tait [66] observed that the cells of a plane closed curve can be coloured black and white so that adjacent cells have different colours. He finishes by remarking:

The development of this subject promises absolutely endless work — but work of a very interesting and useful kind — because it is intimately connected with the theory of knots, which (especially applied in Sir W. Thomson’s Theory of *Vortex Atoms*) is likely soon to become an important branch of mathematics.

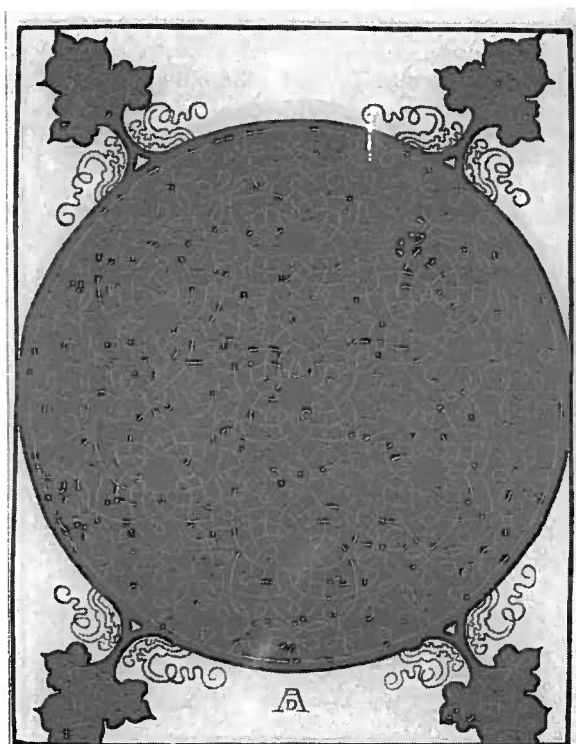
In the theory of ‘vortex atoms’ of Thomson [72], the internal coherence of atoms was assumed to be determined by a knot, or rather a link (a disjoint union of

knots), connecting the different indivisible parts of the atom, the ‘vortex tubes’ (a theory soon abandoned by Thomson). By classifying knots, Tait hoped to shed light on the periodic table of elements.



**Figure 5**

In a note communicated to the Royal Society of Edinburgh on 18 December 1876, Tait [61] observed that any closed curve in the plane gives an alternating knot, just by going alternately over and under. He conjectures that if such an alternating knot is *reduced*, that is, cannot be decomposed as in Figure 5, then it has a minimum number of crossings among all knots equivalent to it; that is, ‘cannot have the number of crossings reduced by any possible deformation.’ As a motivation for considering alternating knots, Tait [65] mentioned that they occur on various sculptured stones and in woodcuts of Dürer.



“I am indebted to Mr Dallas for a photograph of a remarkable engraving by Dürer, exhibiting a very complex but symmetrical linkage, in which this alternation is maintained throughout.” (Tait [65])

After having presented his subsequent ‘Note on the Measure of Beknottedness’ (Tait [62]), Tait’s attention was drawn by the physicist J.C. Maxwell (also Scottish) to Listing’s *Vorstudien zur Topologie*, which Tait next studied with great enthusiasm, calling it an ‘extremely valuable, but too brief, Essay’.

It made Tait aware of the fact that there exist alternating knots that are equivalent but cannot be obtained from each other by trivial operations, as they have different  $\lambda\delta$  type-symbols. In fact, in [63] he states that the sole point of Listing’s paper which (as far as knots are concerned) was thoroughly new to Tait — ‘though not unexpected’ — was an operation that Tait extracted from Listing’s assertion that the knots in Figure 4 are equivalent.

The operation transforms one alternating knot into another. To apply it, one needs to decompose the knot into two blocks as in the first picture in Figure

6. Then one of the blocks is rotated  $180^\circ$ , as indicated in the second picture of Figure 6. Later, Tait called this operation *flying*. Note that also the trivial operations can be obtained as the result of a series of flyings.

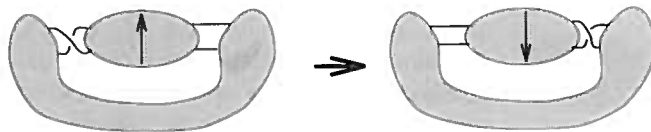


Figure 6

The new operation made Tait conclude that the classification of knots is much more difficult than Tait initially thought,

and it is so because the number of really distinct species of each order is very much *less* than I was prepared to find it.

It made him plan to give up the whole area of knots, as the note ends with:

And here I am glad to leave it, for at this stage it is entirely out of my usual sphere of work, and it has already occupied too much of my time.

But saying farewell to knots is not that easy, and Tait's abstinence was of very short duration. In the same 'Session 1876-77' of the Royal Society of Edinburgh he published five more notes on knots and links, including one on 'Sevenfold Knottiness' [64]. In this paper, the reduced alternating knots with seven crossings are classified. This may be considered as the root of 'Tait's flying conjecture' (although in [64] the term 'flying' is not used yet).

In his classification, the equivalence of knots is derived by applying only flying (including the trivial operations). On the other hand, Tait seemed to have only intuitive means of showing that certain knots are *nonequivalent* — at least, he does not describe in his paper why certain knots are nonequivalent. So Tait assumed without proof that equivalence of alternating knots is completely determined by flying. Therefore one may say that Tait conjectured:

**Tait's flying conjecture.** *Two reduced alternating knots are equivalent if and only if they can be obtained from each other by a series of flyings.*

Tait was aware of the fact that he did not yet have a way of proving nonequivalence of knots, as in [68] he wrote:

... and thus, though I have grouped together many widely different but equivalent forms, I cannot be *absolutely* certain that all those groups are essentially different one from another.

Tait's big article 'On knots' [65] seems the first in which he uses the term *flying*:

The deformation process is, in fact, one of *flying*, an excellent word, very inadequately represented by the nearest equivalent English phrase “turning outside in”.

Although it seems that he restricted the term for turning a knot completely upside down, earlier in the paper the operation of Figure 6 was mentioned:

... this process ... gets rid of a crossing at one place only by introducing it at another. It will be seen later that this process may in certain cases be employed to *change the scheme* of a knot, ...

Moreover, in a later paper, Tait [67] speaks of ‘flying of individual parts’ of a knot, thereby indicating that the general operation described above indeed should be called *flying*.

The word ‘flype’ is old Scottish and means according to *The Concise Scots Dictionary*: ‘fold back; turn wholly or partially inside out; tear off (the skin) in strips, peel’. *A Dictionary of the Older Scottish Tongue, from the Twelfth Century to the End of the Seventeenth* has as lemma:

**Flyp(e)**, *v.* [e.m.E. and ME. *flype* (c. 1400), of obscure origin; current in later Sc. and northern Eng. dialects.] *tr.* To fold back; to turn outwards. Thare laithlie lyning furthwart flypit; LYND. *Syde Tailis* 97. Ane pair of wyd slevis of arming flypand bakward; 1561 *Inv. Wardrobe* 128. Sum flyrand, thair phisnomeis thair flyp [*v.r.* flipe]; MONTG. *Flyt.* 510 (T). I used often to flype up the lids of my eyes; Row 452.

*The Scottish National Dictionary, designed partly on regional lines and partly on historical principles, and containing all the Scottish words known to be in use or to have been in use since c. 1700* gives among other the following usage:

Sc. 1896 Stevenson *W. of Hermiston* vi.:

“Miss Christina, if you please, Mr. Weir!” says I, and just flyped up my skirt tails.

...

Sc. 1721 J. Kelly *Proverbs* 218:

I will sooner see you fleip-ey’d, like a French Cat. A disdainful rejecting of an unworthy Proposal; spoken by bold Maids to the vile offers of young Fellows.

In a discussion of Listing’s *Vorstudien*, Tait [67] describes flying as follows:

When we *flype* a glove (as in taking it very wet, or as we skin a hare), we perform an operation which (not describable in English by any shorter phrase than “*turning outside in*”) changes its character from a right-hand glove to a left. A pair of trousers or a so-called *reversible* waterproof coat is, after this operation has been transformed, still a pair of trousers or a coat, but the legs or arms are interchanged; unless the garments, like those of “Paddius à Corko”, are buttoned behind.

The processes described by (Peter) Tait and the vocabulary introduced by him inspired the physicist (Jack) Maxwell to the following poem:

(CATS) CRADLE SONG

By a Babe in Knots.

Peter the Repeater  
Platted round a platter  
Slips of silvered paper  
Basting them with batter.

Flype 'em, slit 'em, twist 'em,  
Lop-looped laps of paper;  
Setting out the system  
By the bones of Neper.

Clear your coil of kinkings  
Into perfect plaiting,  
Locking loops and linkings  
Interpenetrating.

Why should a man benighted,  
Beduped, befooled, besotted,  
Call knotful knittings plighted,  
Not knotty but beknotted?

It's monstrous, horrid, shocking,  
Beyond the power of thinking,  
Not to know, interlocking  
Is no mere form of linking.

But little Jacky Horner,  
Will teach you what is proper,  
So pitch him, in his corner,  
Your silver and your copper.

Tait [65] also introduced a convenient auxiliary graphical representation of knot and link diagrams (more generally, sets of closed curves) in the plane. Colour the faces of a link diagram  $K$  black and white, so that adjacent faces have different colours, and so that the unbounded face has colour white. Now put a point in each of the black faces.

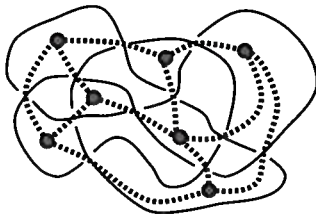


Figure 7

If any two black faces  $f, f'$  are adjacent to a common crossing, draw a line connecting the points in  $f$  and  $f'$  — cf. Figure 7. In this way we obtain a plane graph  $H_K$ , that uniquely determines the projection of the link diagram  $K$ , at least combinatorially. If the link diagram is alternating, we can reconstruct it from  $H_K$  (after adopting a convention on whether each black face corresponds to a dextrotrop or a laetotrop face of the link). We

thus obtain an equivalence of combinatorial questions on alternating knots and on plane graphs.

**3. Work on Tait's conjectures.** Since the work of Listing and Tait, the study of knots has come to great flourishing. Work on distinguishing knots by polynomial invariants (including the well-known Jones polynomial), the connections to mathematical physics, and the applications for instance to DNA have contributed to that. Especially, the work on polynomials has made it possible to prove the nonequivalence of several pairs of knots.

In this ramble I just want to restrict myself to some of the work done on Tait's conjectures. Using the Jones polynomial, Kauffman [27], Murasugi [43], and Thistlethwaite [69] were able to show Tait's conjecture that a reduced alternating link diagram attains a minimum number of crossings, taken over all (not necessarily alternating) links equivalent to it. In particular, any two equivalent

reduced alternating links have the same number of crossings.

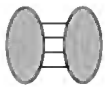


Figure 8

A special case of Tait's flying conjecture was considered in [57]. Call a link diagram  $K$  *well-connected* if it does not have a nontrivial cut that crosses the diagram in at most four curves only. That is, for any decomposition of the diagram as in Figure 8, one of the blocks should contain at most one crossing.

For a well-connected alternating link diagram, flying clearly loses most of its lustre. For well-connected links Tait's flying conjecture reduces to:

**Theorem 1.** *Let  $K$  and  $K'$  be links with well-connected alternating diagrams. Then  $K$  and  $K'$  are equivalent if and only if the diagrams arise from each other by trivial operations.*

Meantime, Menasco and Thistlethwaite [39] have announced a proof of Tait's flying conjecture in full generality.

We sketch some elements of the proofs. Let  $K$  and  $K'$  be two links, with reduced alternating diagrams. We must show that if  $K$  and  $K'$  are equivalent, then their diagrams arise from each other by a series of flyings. In both proofs, surfaces are introduced to trace the movements when transforming  $K'$  to  $K$ .

Let  $K$  be an alternating link, with link diagram having a dextrotrop unbounded face. Then the compact bordered surface  $\Sigma_K$  is 'the' surface with boundary  $K$  and with projection equal to the closure of the union of the laetrop faces. A pictorial impression is given in Figure 9.

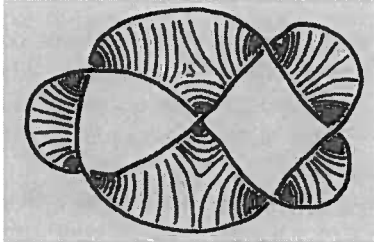


Figure 9

Now note that if we move link  $K'$  to link  $K$ , there will be two surfaces with boundary  $K$ : first the surface  $\Sigma_K$  associated with  $K$ ; second the transformed surface  $\tau(\Sigma_{K'})$ , where  $\tau : S^3 \rightarrow S^3$  describes the isotopy bringing  $K'$  to  $K$ . Thus the surface  $\tau(\Sigma_{K'})$  in a way bears the 'history' of moving  $K'$  to  $K$ .

There are some parameters of compact bordered surfaces that remain in-

variant under isotopy. First, the Euler characteristic is an invariant. A second parameter invariant under isotopy is the *twisting number*, which is about the number of twists one makes when driving on the surface, close to the boundary, like on a roller coaster (added up over all boundaries).

Now one can show that if  $K$  is a link with well-connected alternating diagram and if  $\Sigma$  is any compact bordered surface with boundary  $K$  and with the same Euler characteristic and twisting number as  $\Sigma_K$ , then there is an isotopy bringing  $\Sigma$  to  $\Sigma_K$ .

This directly gives, for any two equivalent links  $K$  and  $K'$  with well-connected alternating diagrams, that there is an isotopy bringing  $\Sigma_{K'}$  to  $\Sigma_K$ . Indeed, for this it suffices to show that  $\Sigma_K$  and  $\Sigma_{K'}$  have the same Euler characteristic

and the same twisting number. This follows directly from earlier results on the invariance of the number of black faces and of the ‘writhe’ of a link (Murasugi [44], Thistlethwaite [70], [71]).

Finally, to finish the proof of Theorem 1, one has for links  $K$  and  $K'$  with well-connected alternating diagrams: if there is an isotopy bringing  $\Sigma_{K'}$  to  $\Sigma_K$ , then the diagrams arise from each other by trivial operations. This fact is proved by showing that if  $\Sigma_K$  and  $\Sigma_{K'}$  are isotopic, then the cycle spaces of  $H_K$  and  $H_{K'}$  form isomorphic matroids. This is shown by comparing the twisting numbers of circuits in  $\Sigma_K$  and  $\Sigma_{K'}$ .







Hence, by a theorem of Whitney [76],  $H_K$  and  $H_{K'}$  are the same up to trivial operations (note that these plane graphs are 3-connected by the well-connectedness of the diagrams). This gives that the diagrams are the same up to trivial operations, and thus we have Tait’s flying conjecture for well-connected links.

The proof of the full Tait flying conjecture as announced by Menasco and Thistlethwaite [39] makes a more extensive use of invariants, including polynomial invariants, and applies them simultaneously to the surface  $\Sigma_K$  and to the surface  $\Sigma'_K$  obtained similarly as  $\Sigma_K$  but with respect to the dextrotrop faces (assuming the link diagram being on the 2-sphere).

**4. Reidemeister moves.** A basis of representing a knot by its diagram is that never more than two points of a knot project to the same point in the plane, and if two points have the same projection, it is a crossing. By this one does not lose generality.

Reidemeister [48] observed that this principle can be extended. If one considers the isotopic move of a knot, one has a fourth dimension, the time. Then one may assume that the move is so that at any fixed moment not more than three points of the knot project to the same point in the plane, and if three points have the same projection, they pairwise cross.

Further analysis led Reidemeister to showing that if two links are equivalent, then their diagrams can be moved to each other by a series of simple operations, called *Reidemeister moves*:

- (1)      *type I*: replacing  by , and conversely;
- type II*: replacing  by , and conversely;
- type III*: replacing  by .

(In Reidemeister’s book *Knotentheorie* [49], these operations are called  $\Omega.1$ ,  $\Omega.2$ , and  $\Omega.3$ .)

It enables to study knot equivalence just by diagrams, and it reduces knot equivalence to a combinatorial question. Most of the knot polynomials have been shown to be invariant by showing that they are invariant under the Reidemeister moves.



On the other hand, Reidemeister moves do not imply a finite algorithm to test if two given knots are equivalent. There is no upper bound known (expressed in the number of crossings of the knots) for the number of Reidemeister moves to be made to transform one knot to another, equivalent, knot. Equivalently, there is no upper bound known for the maximum number of crossings at intermediate diagrams when transforming two equivalent knots to each other by Reidemeister moves.

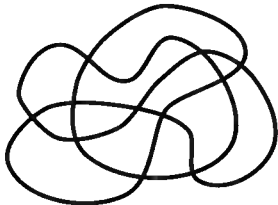








Figure 10

Consider next a closed curve in the plane, like in Figure 10, assuming that there are only a finite number of double points, each being a crossing of two curve parts. It is quite trivial to show that it can be unwrapped to a *simple* closed curve by a series of the following operations — which are also called Reidemeister moves:

- (2) *type I*: replacing  by , and conversely;  
*type II*: replacing  by , and conversely;  
*type III*: replacing  by .

Next it is an easy exercise to show something stronger: in transforming a plane closed curve to a simple curve we can restrict the Reidemeister moves to those *not increasing the number of crossings*. That is, the Reidemeister moves of types I and II are only applied from left to right in (2). A similar statement holds when transforming a system of plane closed curves to a system of pairwise disjoint simple closed curves, except that we should add a Reidemeister move of type 0:

- (3) *type 0*: replacing  by .

(Using the analogy between a system  $K$  of plane closed curves and the plane graph  $H_K$  as introduced by Tait (see Figure 7), one can derive from this the result of Grünbaum [23] that each plane graph can be obtained from the empty graph by a series of the following operations: (i) adding a new vertex, possibly connected by a new edge to an existing vertex; (ii) adding a new edge parallel to an existing edge; (iii) adding a new vertex in the ‘midst’ of an existing edge; (iv) ‘ $Y\Delta$ ’, that is, replacing a vertex  $v$  of degree 3, and the three edges incident with  $v$ , by a triangle connecting the three vertices adjacent to  $v$ ; (v) ‘ $\Delta Y$ ’, that is, the operation reverse to (iv).)

If we have a closed curve  $C$  on a compact surface  $S$  it is clear that in general one cannot make it simple by Reidemeister moves. The best one may hope for is to reduce the number of crossings to the minimum number of crossings taken over all closed curves freely homotopic to  $C$ .

That is, define

- (4)  $\text{mincr}(C) := \min\{\text{cr}(C') \mid C' \text{ freely homotopic to } C\}$ .

Here  $\text{cr}(C')$  denotes the number of selfcrossings of  $C'$ , counting multiplicities. Two closed curves  $C, C' : S^1 \rightarrow S$  are *freely homotopic*, in notation  $C \sim C'$ , if there exists a continuous function  $\Phi : S^1 \times [0, 1] \rightarrow S$  such that  $\Phi(x, 0) = C(x)$  and  $\Phi(x, 1) = C'(x)$  for each  $x \in S^1$ .

Call  $C$  *minimally crossing* if  $\text{cr}(C) = \text{mincr}(C)$ . Then it is shown in [22] that each closed curve  $C$  can be transformed to a minimally crossing closed curve by Reidemeister moves, *without increasing the number of crossings* throughout the moves.

This holds more generally for systems of closed curves. To this end define for closed curves  $C$  and  $D$  on  $S$ :

$$(5) \quad \text{mincr}(C, D) := \min\{\text{cr}(C', D') \mid C' \sim C, D' \sim D\}.$$

Here  $\text{cr}(C', D')$  is the number of crossings of  $C'$  and  $D'$ , counting multiplicities. A system  $C_1, \dots, C_k$  of closed curves on  $S$  is called *minimally crossing* if each  $C_i$  is minimally crossing and if  $\text{cr}(C_i, C_j) = \text{mincr}(C_i, C_j)$  for all  $i \neq j$ .

Then the following is proved in [22]:

**Theorem 2.** *Any system of closed curves on a surface can be transformed to a minimally crossing system by a series of Reidemeister moves, without increasing the number of crossings during the moves.*

(To be precise, one should add some tameness assumptions: the surface should be triangulizable, and the system of closed curves should have only a finite number of double points, each being a crossing.)

It is important to note that the main content of Theorem 2 is that one does not need to apply any of the operations (2) in the reverse direction — otherwise the result would follow quite straightforwardly with the techniques of simplicial approximation.

The idea of the proof is as follows (for one nontrivial closed curve  $C$ ). First it is shown that one may assume that  $S$  is ‘hyperbolic’, that is, has a hyperbolic distance on it. Then  $C$  is freely homotopic to a unique shortest closed curve  $C'$  on  $S$ . Consider the following operation. Choose a closed disk  $\Delta$  on  $S$ , convex with respect to the hyperbolic distance. Straighten out the intersections of  $C$  with  $\Delta$ ; that is, replace each intersection  $I$  by the shortest curve that has the same end points as  $I$ . Due to an extension of a theorem of Ringel [50], this can be done by applying Reidemeister moves to  $\Delta$ .

Now one may show that by choosing a finite number of closed disks  $\Delta$ , one can move  $C$  arbitrarily close to  $C'$ . Then making  $C$  minimally crossing essentially is reduced to making a closed curve on the annulus or the Möbius strip minimally crossing (depending on whether  $C$  is orientation preserving or not). This last turns out to boil down to the following auxiliary results on permutations.

Let  $\pi$  be a permutation of  $\{1, \dots, n\}$ . A *crossing pair* of  $\pi$  is a pair  $\{i, j\}$  with  $(i - j)(\pi(i) - \pi(j)) < 0$ . The *crossing number* (or *length* (cf. Bourbaki [7]))

$\text{cr}(\pi)$  of  $\pi$  is the number of crossing pairs of  $\pi$ .

Let  $\text{mincr}(\pi)$  denote the minimum of  $\text{cr}(\pi')$  taken over all conjugates  $\pi'$  of  $\pi$ . So  $\text{mincr}(\pi)$  only depends on the sizes of the orbits of  $\pi$ . A permutation is *minimally crossing* if  $\text{cr}(\pi) = \text{mincr}(\pi)$ . Similarly, *maximally crossing* is defined.

A *transposition* is any permutation  $(k, k + 1)$  for some  $k \in \{1, \dots, n - 1\}$ . Since each permutation  $\sigma$  is a product of transpositions, it is trivial to say that each permutation  $\pi$  can be transformed to a minimally crossing permutation by a series of operations

$$(6) \quad \pi \rightarrow \tau\pi\tau,$$

where  $\tau$  is a transposition. Similarly for maximally crossing.

What however can be proved more strongly is:

**Lemma.** *Each permutation  $\pi$  of  $\{1, \dots, n\}$  can be transformed to a minimally crossing permutation by a series of operations (6), while never increasing the number of crossing pairs. A similar statement holds for maximally crossing.*

Geck and Pfeiffer [21] proved the first part of the Lemma more generally for any Weyl group (instead of just a permutation group). It is not known if also the 'maximally crossing' part also holds for Weyl groups.

**5. Curves and circulations on surfaces.** One motivation for studying Reidemeister moves on surfaces was to derive a homotopic circulation theorem for graphs embedded on a surface. Once one has Theorem 2, such a circulation theorem can be derived by a number of straightforward arguments based on two kinds of duality: duality of graphs on surfaces and linear programming duality (Farkas' lemma).

Again, let  $S$  be a surface, and let  $G = (V, E)$  be an undirected graph embedded on  $S$ . For any closed curve  $D$  on  $S$ , let  $\text{cr}(G, D)$  denote the number of intersections of  $G$  and  $D$  (counting multiplicities). Moreover,  $\text{mincr}(G, D)$  denotes the minimum of  $\text{cr}(G, D')$  where  $D'$  ranges over all closed curves freely homotopic to  $D$  and *not intersecting*  $V$ .

We first derive the following theorem from Theorem 2, which was proved for the projective plane by Lins [36]:

**Theorem 3.** *Let  $G = (V, E)$  be an Eulerian graph embedded on a surface  $S$ . Then the edges of  $G$  can be decomposed into closed curves  $C_1, \dots, C_k$  such that for each closed curve  $D$  on  $S$ :*

$$(7) \quad \text{mincr}(G, D) = \sum_{i=1}^k \text{mincr}(C_i, D).$$

Here a graph is *Eulerian* if each vertex has even degree. (Connectedness of the graph is not assumed.) Moreover, *decomposing* the edges into  $C_1, \dots, C_k$  means that each edge of  $G$  is traversed by exactly one of the  $C_i$ .

Note that the inequality  $\geq$  in (7) trivially holds, for *any* decomposition of the edges into closed curves  $C_1, \dots, C_k$ . The content of the theorem is that there exists a decomposition attaining equality for each  $D$ .

The idea of the proof is as follows. First, by an easy construction we may assume that each vertex  $v$  of  $G$  has degree at most four. Next, we define the *straight decomposition* of  $G$  as the system of closed curves that decomposes the edges of  $G$  in such a way that in each vertex of  $G$ , opposite edges are traversed consecutively. So each vertex of  $G$  of degree four represents a (self-)crossing of  $C_1, \dots, C_k$ .

Up to some trivial operations, such a decomposition is unique, and conversely, it uniquely describes  $G$ . So any Reidemeister move applied to  $C_1, \dots, C_k$  carries over a modification of  $G$ . Hence we can speak of Reidemeister moves applied to  $G$ .

The following is easy to see:

- (8) if  $G'$  arises from  $G$  by one Reidemeister move of type III, then  $\text{mincr}(G', D) = \text{mincr}(G, D)$  for each closed curve  $D$ .

Let us call any graph  $G = (V, E)$  that is a counterexample to the theorem with each vertex having degree at most four and with a minimal number of faces, a *minimal counterexample*.

From (8) it directly follows that:

- (9) if  $G'$  arises from a minimal counterexample  $G$  by one Reidemeister move of type III, then  $G'$  is a minimal counterexample again.

Moreover one has:

- (10) if  $G$  is a minimal counterexample, then no Reidemeister move of type 0, I or II can be applied to  $G$  without increasing the number of vertices of  $G$ .

For suppose that a Reidemeister move of type II can be applied to  $G$ . Then  $G$  contains  $\bowtie \times$  as subconfiguration. Replacing this by  $\asymp \times$  would give a smaller counterexample (since the function  $\text{mincr}(G, D)$  does not change by this operation), contradicting the minimality of  $G$ .

One similarly sees that no Reidemeister move of type 0 or I can be applied.

The proof is finished by showing the contradictory statement that the straight decomposition  $C_1, \dots, C_k$  of any minimal counterexample  $G$  satisfies (7).

Choose a closed curve  $D$ . By Theorem 2 we can apply Reidemeister moves to the system  $D, C_1, \dots, C_k$  so as to obtain a minimally crossing system  $D', C'_1, \dots, C'_k$ .

By (10) we did not apply Reidemeister moves of type 0, I or II to  $C_1, \dots, C_k$ . Hence by (8) for the graph  $G'$  obtained from the final  $C'_1, \dots, C'_k$  we have  $\text{mincr}(G', D) = \text{mincr}(G, D)$ . So

$$(11) \quad \begin{aligned} \operatorname{mincr}(G, D) &= \operatorname{mincr}(G', D) \leq \operatorname{cr}(G', D') = \sum_{i=1}^k \operatorname{cr}(C'_i, D') \\ &= \sum_{i=1}^k \operatorname{mincr}(C'_i, D') = \sum_{i=1}^k \operatorname{mincr}(C_i, D). \end{aligned}$$

This proves Theorem 3.

Using surface duality one directly obtains from Theorem 3 the next theorem. If  $G$  is a graph embedded on a surface  $S$  and  $C$  is a closed curve in  $G$ , then  $\operatorname{minlength}_G(C)$  denotes the minimum length of any closed curve  $C' \sim C$  in  $G$ . (The *length* of  $C'$  is the number of edges traversed by  $C'$ , counting multiplicities.)

**Theorem 4.** *Let  $G = (V, E)$  be a bipartite graph cellularly embedded on a compact surface  $S$ . Then there exist closed curves  $D_1, \dots, D_t$  on  $S \setminus V$  such that each edge of  $G$  is crossed by exactly one  $D_j$  and by this  $D_j$  only once and such that for each closed curve  $C$ :*

$$(12) \quad \operatorname{minlength}_G(C) = \sum_{j=1}^t \operatorname{mincr}(C, D_j).$$

Now with linear programming duality (Farkas' lemma) one derives from Theorem 4 the following 'homotopic circulation theorem' — a fractional packing theorem for cycles of given homotopies in a graph on a compact surface.

Let  $G = (V, E)$  be a graph embedded on a compact surface  $S$ . For any closed curve  $C$  on  $G$  and any edge  $e$  of  $G$  let  $\operatorname{tr}_C(e)$  denote the number of times  $C$  traverses  $e$ . So  $\operatorname{tr}_C \in \mathbb{R}^E$ .

Call a function  $f : E \rightarrow \mathbb{R}$  a *circulation* (of value 1) if  $f$  is a convex combination of functions  $\operatorname{tr}_C$ . We say that  $f$  is *freely homotopic* to a closed curve  $C_0$  if we can take each  $C$  freely homotopic to  $C_0$ .

**Theorem 5** (homotopic circulation theorem). *Let  $G = (V, E)$  be an undirected graph embedded on a compact surface  $S$  and let  $C_1, \dots, C_k$  be closed curves on  $S$ . Then there exist circulations  $f_1, \dots, f_k$  such that  $f_i$  is freely homotopic to  $C_i$  ( $i = 1, \dots, k$ ) and such that  $\sum_{i=1}^k f_i(e) \leq 1$  for each edge  $e$ , if and only if for each closed curve  $D$  on  $S \setminus V$  one has*

$$(13) \quad \operatorname{cr}(G, D) \geq \sum_{i=1}^k \operatorname{mincr}(C_i, D).$$

We sketch the proof if  $G$  is cellularly embedded. Necessity of the condition is direct. To show sufficiency, by Farkas' lemma (cf. [54]) it suffices to show that if  $d \in \mathbb{Q}^k$  and  $l \in \mathbb{Q}_+^E$  such that  $\sum_{e \in E} \operatorname{tr}_C(e) \geq d_i$  for each  $i$  and each closed curve  $C \sim C_i$  in  $G$ , then  $\sum_{e \in E} l(e) \geq \sum_{i=1}^k d_i$ .

Then one can show that it may be assumed that each  $d_i$  and each  $l(e)$  is an even integer, and that  $l(e) > 0$  for each  $e$ . Replacing each edge  $e$  by a path of length  $l(e)$  makes  $G$  into a bipartite graph  $G'$ . Applying (13) to each of the  $D_j$  of Theorem 4 gives the required inequality.

**6. Disjoint curves in graphs on surfaces.** In the homotopic circulation theorem one may wonder when there exists an integer-valued circulation. This would correspond to a system of pairwise edge-disjoint cycles  $C'_1, \dots, C'_k$  in  $G$  with  $C'_i$  freely homotopic to  $C_i$ . However, the conditions given in the theorem are not sufficient to get an integer-valued circulation; and no additional conditions are known to ensure the existence of an integer-valued circulation.

If we want to have *vertex*-disjoint circuits, such conditions have been given in [55], proving a conjecture of L. Lovász and P.D. Seymour:

**Theorem 6.** *Let  $G$  be an undirected graph embedded on a compact surface  $S$  and let  $C_1, \dots, C_k$  be pairwise disjoint simple closed curves on  $S$ . Then there exist pairwise disjoint simple circuits  $C'_1, \dots, C'_k$  in  $G$  where  $C'_i$  is freely homotopic to  $C_i$  for  $i = 1, \dots, k$ , if and only if*

$$(14) \quad \text{cr}(G, D) \geq \sum_{i=1}^k \text{mincr}(C_i, D)$$

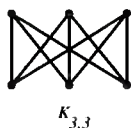
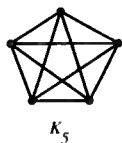
for each closed curve  $D$  on  $S$ , with strict inequality if  $D$  is doubly odd.

Here a closed curve  $D$  is *doubly odd* if  $D$  is the concatenation of two closed curves  $D_1$  and  $D_2$ , with a common beginning (= end) point, which is not on  $G$ , in such a way that  $\text{cr}(G, D_j) + \sum_{i=1}^k \text{mincr}(C_i, D_j)$  is odd for  $j = 1, 2$ . It is not difficult to see that the condition given in the theorem is necessary.

The problem solved in Theorem 6 arose during the graph minors project of N. Robertson and P.D. Seymour. Principal result of this deep project is a proof ([53]) of Wagner's conjecture: in any infinite class of graphs there are graphs  $G$  and  $H$  such that  $H$  is a minor of  $G$ . ( $H$  is a *minor* of  $G$  if  $H$  arises from  $G$  by a series of deletions and contractions of edges.)

Equivalent to Robertson and Seymour's theorem is that if  $\mathcal{G}$  is a class of graphs closed under taking minors, then there is a *finite* collection  $\mathcal{H}$  of graphs with the property that a graph  $G$  belongs to  $\mathcal{G}$  if and only if  $G$  does not have a minor  $H$  with  $H \in \mathcal{H}$ .

We may assume that  $\mathcal{H}$  does not contain two graphs  $H, H'$  such that  $H'$  is a minor of  $H$ . Then  $\mathcal{H}$  is called the set of *forbidden minors* of  $\mathcal{G}$ .



The well-known theorem of Kuratowski [34] (or rather, its equivalent formulation by Wagner [74]) states that if  $\mathcal{G}$  is the class of planar graphs, then  $\{K_5, K_{3,3}\}$  is the set of forbidden minors.

A consequence of Robertson and Seymour's theorem is that for *any* surface  $S$  there is a finite class of forbidden minors for the class of graphs embeddable on  $S$ . This was shown before by Archdeacon [2] for the projective plane and by Archdeacon and Huneke [3] for compact nonorientable surfaces.

Very roughly speaking, the proof of Robertson and Seymour of Wagner's conjecture is as follows. It can be shown that for any graph  $G$  there is a finite collection of surfaces such that each graph not containing  $G$  as a minor can be expressed as a tree-structure of 'pieces' such that each piece can 'almost' be drawn on a surface in the collection. Part of the proof next is that any graph  $H$  embedded on a surface  $S$  is a minor of each graph that is embedded densely enough on  $S$  ('enough' depending on  $H$ ).

Related to this last statement is the question under which conditions for two given graphs  $G$  and  $H$  embedded on  $S$ ,  $H$  is a minor of  $G$  on  $S$ . That is, when can we delete and contract edges of  $G$ , while keeping the embedding, so as to obtain  $H$  (possibly after a homotopic shift of  $H$  over  $S$ ). The case where  $H$  consists of disjoint loops only is solved in Theorem 6.

The more general case of this question where  $H$  is an arbitrary graph is not solved completely, but can be approached slightly similarly as follows. Let  $G$  and  $H$  be graphs embedded on  $S$ . For each edge  $f$  of  $H$  choose an edge  $e_f$  of  $G$ . Now we wish to complete these edges to a minor of  $G$  isomorphic to  $H$ . By this it is meant that one should find for each vertex  $v$  of  $H$  a tree  $T_v$  in  $G$  such that the  $T_v$  are mutually disjoint and such that for each edge  $f$  of  $H$ ,  $e_f$  is incident with  $T_v$  if and only if  $f$  is incident with  $v$ . Thus contracting each tree  $T_v$  to one vertex, the edges  $e_f$  would give a minor isomorphic to  $H$ .

Now an extension of Theorem 6 (cf. [56]) characterizes under which conditions such trees exist, given the homotopy of the trees. It amounts to finding disjoint trees  $T_1, \dots, T_k$  such that each  $T_i$  connects a given set  $V_i$  of vertices. If each  $V_i$  just consists of two vertices, it reduces to a *disjoint paths problem*.

**7. Menger and König.** Disjoint paths problems belong to the heart of classical graph theory. They go back to 1927, when the topologist Karl Menger [40] published an article called *Zur allgemeinen Kurventheorie* in which he showed a result that now is one of the most fundamental results in graph theory:

*Satz  $\beta$ . Ist  $K$  ein kompakter regulär eindimensionaler Raum, welcher zwischen den beiden endlichen Mengen  $P$  und  $Q$   $n$ -punktig zusammenhängend ist, dann enthält  $K$   $n$  paarweise fremde Bögen, von denen jeder einen Punkt von  $P$  und einen Punkt von  $Q$  verbindet.*

The result can be formulated as a maximum-minimum theorem in terms of graphs, as follows:

**Menger's theorem.** *Let  $G = (V, E)$  be an undirected graph and let  $P, Q \subseteq V$ . Then the maximum number of pairwise disjoint  $P - Q$  paths is equal to the minimum cardinality  $n$  of any set of vertices that intersects each  $P - Q$  path.*

Here a  $P - Q$  path is a path starting in  $P$  and ending in  $Q$ . Two paths are *disjoint* if they do not have any vertex or edge in common. The result became also known as the  $n$ -chain theorem or the  $n$ -arc theorem. Knaster [28] observed that (by an easy construction) Menger's theorem is equivalent to:

**Menger's theorem (variant).** *Let  $G = (V, E)$  be an undirected graph and let  $s, t \in V$  with  $st \notin E$ . Then the maximum number of pairwise internally disjoint  $s - t$  paths is equal to the minimum cardinality of any subset of  $V \setminus \{s, t\}$  that intersects each  $s - t$  path.*

Here an  $s - t$  path is a path starting in  $s$  and ending in  $t$ . Two paths are *internally disjoint* if they do not have a vertex or edge in common, except for the end vertices.

Why was Menger interested in this question? In his article he investigates a certain class of topological spaces called 'Kurven': a *curve* is a connected compact topological space  $X$  with the property that for each  $x \in X$  and each neighbourhood  $N$  of  $x$  there exists a neighbourhood  $N' \subseteq N$  of  $x$  such that  $\text{bd}(N')$  is totally disconnected. Here  $\text{bd}$  stands for 'boundary'; a space is *totally disconnected* if each point forms an open set. Notice that each graph, considered as a topological space, is a curve in Menger's terminology.

In particular, Menger was motivated by characterizing a certain furcation number of curves. To this end, a curve  $X$  is called *regular* if for each  $x \in X$  and each neighbourhood  $N$  of  $x$  there exists a neighbourhood  $N' \subseteq N$  of  $x$  such that  $|\text{bd}(N')|$  is finite. The *order* of a point  $x \in X$  is equal to the minimum natural number  $n$  such that for each neighbourhood  $N$  of  $x$  there exists a neighbourhood  $N' \subseteq N$  of  $x$  satisfying  $|\text{bd}(N')| \leq n$ .

According to Menger:

Eines der wichtigsten Probleme der Kurventheorie ist die Frage nach die Beziehungen zwischen der Ordnungszahl eines Punktes der regulären Kurve  $K$  und der Anzahl der im betreffenden Punkt zusammenstossenden und sonst fremden Teilbögen von  $K$ .

In fact, Menger used 'Satz  $\beta$ ' to show that if a point in a regular curve  $K$  has order  $n$ , then there exists a topological  $n$ -leg with  $p$  as top; that is,  $K$  contains  $n$  arcs  $P_1, \dots, P_n$  such that  $P_i \cap P_j = \{p\}$  for all  $i, j$  with  $i \neq j$ .

The proof idea is as follows. There exists a series  $N_1 \supset N_2 \supset \dots$  of open neighbourhoods of  $p$  such that  $N_1 \cap N_2 \cap \dots = \{p\}$  and  $|\text{bd}(N_i)| = n$  for all  $i = 1, 2, \dots$ , and such that

$$(15) \quad |\text{bd}(N)| \geq n \text{ for each neighbourhood } N \subseteq N_1.$$

This follows quite directly from the definition of order.

Now Menger showed that we may assume that the space  $G_i := \overline{N_i} \setminus N_{i+1}$  is a (topological) graph. For each  $i$ , let  $Q_i := \text{bd}(N_i)$ . Then (15) gives with Menger's theorem that there exist  $n$  pairwise disjoint paths  $P_{i,1}, \dots, P_{i,n}$  in  $G$



such that each  $P_{i,j}$  runs from  $Q_i$  to  $Q_{i+1}$ . Properly connecting these paths for  $i = 1, 2, \dots$  we obtain  $n$  arcs forming the required  $n$ -leg.

It was however noticed by König [30] that Menger gave a lacunary proof of ‘Satz  $\beta$ ’. Menger applies induction on  $|E|$ , where  $E$  is the edge set of the graph  $G$ . Menger first claims that one easily shows that  $|E| \geq n$ , and that if  $|E| = n$  then  $G$  consists of  $n$  disjoint arcs connecting  $P$  and  $Q$ . He states that if  $|E| > n$  then there is a vertex  $s \notin P \cup Q$ , or in his words (where the ‘Grad’ denotes  $|E|$ ):

Wir nehmen also an, der irreduzibel  $n$ -punktig zusammenhängende Raum  $K'$  besitze den Grad  $g (> n)$ . Offenbar enthält dann  $K'$  ein punktförmiges Stück  $s$ , welches in der Menge  $P + Q$  nicht enthalten ist.

Indeed, as Menger shows, if such a vertex  $s$  exists one is done: If  $s$  is not contained in any set  $W$  intersecting each  $P - Q$  path such that  $|W| = n$ , then we can delete  $s$  and the edges incident with  $s$  without decreasing the minimum in the theorem. If  $s$  is contained in some set  $W$  intersecting each  $P - Q$  path such that  $|W| = n$ , then we can split  $G$  into two subgraphs  $G_1$  and  $G_2$  that intersect in  $W$  in such a way that  $P \subseteq G_1$  and  $Q \subseteq G_2$ . By the induction hypothesis, there exist  $n$  pairwise disjoint  $P - W$  paths in  $G_1$  and  $n$  pairwise disjoint  $W - Q$  paths in  $G_2$ . By pairwise sticking these paths together at  $W$  we obtain paths as required.

However, such a vertex  $s$  need not exist. It might be that  $V$  is the disjoint union of  $P$  and  $Q$  in such a way that each edge connects  $P$  and  $Q$ . In that case,  $G$  is a bipartite graph, and what should be shown is that  $G$  contains a matching (= set of disjoint edges) of size  $n$ . This is a nontrivial basis of the proof.

It is unclear when Menger became aware of the hole. In his reminiscences on the origin of the  $n$ -arc theorem, Menger [42] wrote in 1981:

In the spring of 1930, I came through Budapest and met there a galaxy of Hungarian mathematicians. In particular, I enjoyed making the acquaintance of Dénes König, for I greatly admired the work on set theory of his father, the late Julius König—to this day one of the most significant contributions to the continuum problem—and I had read with interest some of Dénes papers. König told me that he was about to finish a book that would include all that was known about graphs. I assured him that such a book would fill a great need; and I brought up my  $n$ -Arc Theorem which, having been published as a lemma in a curve-theoretical paper, had not yet come to his attention. König was greatly interested, but did not believe that the theorem was correct. “This evening,” he said to me in parting, “I won’t go to sleep before having constructed a counterexample.” When we met the next day he greeted me with the words, “A sleepless night!” and asked me to sketch my proof for him. He then said that he would add to his book a final section devoted to my theorem. This he did; and it is largely thanks to König’s valuable book that the  $n$ -Arc Theorem has become widely known among graph theorists.

Dénes König was a pioneer in graph theory and in applying graphs to other areas like set theory, matrix theory, and topology. He had published in the

1910s theorems on perfect matchings and on factorizations of regular bipartite graphs in relation to the study of determinants by Frobenius.

At the meeting of 26 March 1931 of the Eötvös Loránd Matematikai és Fizikai Társulat (Loránd Eötvös Mathematical and Physical Society) in Budapest, König [29] presented a result that formed in fact the induction basis for Menger's theorem:

Páros körüljárású graphban az éleket kimerítő szögpontok minimális száma megegyezik a páronként közös végpontot nem tartalmazó élek maximális számával.

In other words:

**König's theorem.** *In a bipartite graph  $G = (V, E)$ , the maximum size of a matching is equal to the minimum number of vertices needed to cover all edges.*

König did not mention in his paper that this result provided the missing induction basis in Menger's proof, although he finishes with:

Megemlítettük végül, hogy eredményeink szorosan összefüggnek FROBENIUSnak determinánsokra és MENGERnek graphokra vonatkozó némely vizsgálatával. E kapcsolatokra másutt fogunk kiterjeszkedni.

'Másutt' became König [30], where a full proof of Menger's theorem is given, with the following footnote:

Der Beweis von MENGER enthält eine Lücke, da es vorausgesetzt wird (S. 102, Zeile 3–4) daß " $K'$  ein punktförmiges Stück  $s$  enthält, welches in der Menge  $P + Q$  nicht enthalten ist", während es recht wohl möglich ist, daß — mit der hier gewählten Bezeichnungsweise ausgedrückt — jeder Knotenpunkt von  $G$  zu  $H_1 + H_2$  gehört. Dieser — keineswegs einfacher — Fall wurde in unserer Darstellung durch den Beweis des Satzes 13 erledigt. Die weiteren — hier folgenden — Überlegungen, die uns zum Mengerschen Satz führen werden, stimmen in Wesentlichen mit dem — sehr kurz gefaßten — Beweis von MENGER überein. In Anbetracht der Allgemeinheit und Wichtigkeit des Mengerschen Satzes wird im Folgenden auch dieser Teil ganz ausführlich und den Forderungen der *reinkombinatorischen* Graphentheorie entsprechend dargestellt.

[Zusatz bei der Korrektur, 10.V.1933] Herr MENGER hat die Freundlichkeit gehabt — nachdem ich ihm die Korrektur meiner vorliegenden Arbeit zugeschiedt habe — mir mitzuteilen, daß ihm die oben beanstandete Lücke seines Beweises schon bekannt war, daß jedoch sein vor Kurzem erschienenenes Buch *Kurventheorie* (Leipzig, 1932) einen vollkommen lückenlosen und rein kombinatorischen Beweis des Mengerschen Satzes (des " $n$ -Kettensatzes") enthält. Mir blieb dieser Beweis bis jetzt unbekannt.

This book of Menger [41] was published in 1932, and contains a complete proof of Menger's theorem. Menger did not refer to any hole in his proof, but remarked:

Über den  $n$ -Kettensatz für Graphen und die im vorangehenden zum Beweise verwendete Methode vgl. Menger (Fund. Math. 10, 1927, S. 101 f.). Die obige detaillierte Ausarbeitung und Darstellung stammt von Nöbeling.

In his book *Theorie der endlichen und unendlichen Graphen*, published in 1936, König [31] calls his theorem *ein wichtiger Satz*, and he emphasizes the chronological order of the proofs of Menger's theorem and of König's theorem (which is implied by Menger's theorem):

Ich habe diesen Satz 1931 ausgesprochen und bewiesen, s. König [9 und 11]. 1932 erschien dann der erste lückenlose Beweis des Mengerschen Graphensatzes, von dem in §4 die Rede sein wird und welcher als eine Verallgemeinerung dieses Satzes 13 (falls dieser *nur für endliche* Graphen formuliert wird) angesehen werden kann.

**8. Disjoint paths and trees.** Menger's theorem addresses the problem of finding a set of paths with one common beginning vertex and one common end vertex. A more general problem is the following *disjoint paths problem*:

- (16)      given: a graph  $G = (V, E)$  and  $k$  pairs of vertices  $s_1, t_1, \dots, s_k, t_k$ ;  
             find: pairwise disjoint paths  $P_1, \dots, P_k$  where  $P_i$  runs from  $s_i$  to  $t_i$  ( $i = 1, \dots, k$ ).

This covers four variants of the problem: the graph can be directed or undirected, and 'disjoint' can mean: vertex-disjoint or edge-disjoint.

In 1974, D.E. Knuth (see [26]) showed that the edge-disjoint undirected variant, and hence also each of the other variants, is NP-complete — and this is even so if we restrict ourselves to planar graphs (Lynch [38]). This destroys (for those believing  $NP \neq co-NP$  or  $NP \neq P$ ) the hope for nice theorems (like Menger's theorem) and for fast algorithms for solving this problem.

On the other hand, Robertson and Seymour [52], as another important result of their graph minors project, proved that for each *fixed*  $k$ , there exists a polynomial-time algorithm for the disjoint paths problem for undirected graphs. Their algorithm has running time bounded by  $c_k |V|^3$ , for some constant  $c_k$  heavily depending on  $k$ . (It implies that for each fixed graph  $H$  there exists a polynomial time algorithm to test if a given graph  $G$  contains  $H$  as a minor.)

For *directed* graphs, the situation seems different. In 1980, Fortune, Hopcroft, and Wyllie [20] showed the NP-completeness of the vertex-disjoint paths problem for directed graphs, even when restricted to the case  $k = 2$ .

For *planar* directed graphs however there is a positive result ([58]):

**Theorem 7.** *For each fixed  $k$  there is a polynomial-time algorithm for the  $k$  vertex-disjoint paths problem for directed planar graphs.*

This is a result only of interest from the point of view of theoretical complexity: the degree of the polynomial bounding the running time of the algorithm is quadratic in  $k$ .

The proof of Theorem 7 is based on representing disjoint paths as ‘flows’ over a free group. Indeed, let a directed planar graph  $D = (V, A)$  and  $s_1, t_1, \dots, s_k, t_k \in V$  be given. Let  $G_k$  be the free group with  $k$  generators  $g_1, \dots, g_k$ . If  $\Pi = (P_1, \dots, P_k)$  is a solution to the disjoint paths problem, let  $\phi_\Pi : A \rightarrow G_k$  be defined by, for  $a \in A$ :  $\phi_\Pi(a) := g_i$  if  $P_i$  traverses  $a$  ( $i = 1, \dots, k$ ), and  $:= 1$  if no  $P_i$  traverses  $a$ .

Let  $F$  be the set of faces of  $D$ . Call two functions  $\phi, \psi : A \rightarrow G_k$  *homologous* if there exists a function  $p : F \rightarrow G_k$  such that for each arc  $a$  of  $D$  one has:

$$(17) \quad \psi(a) = p(f)^{-1} \phi(a) p(f'),$$

where  $f$  and  $f'$  are the faces at the left hand side and the right hand side of  $a$  respectively (with respect to the orientation of the plane and of the arc  $a$ ).

This defines an equivalence relation on functions  $A \rightarrow G_k$ . We now enumerate representatives of homology classes of functions  $A \rightarrow G_k$ . Generally there are infinitely many homology classes, but one can find in polynomial time a collection of  $O(|V|^{2k^2+3})$  homology classes of which one can be sure that it covers all functions  $\phi_\Pi$  with  $\Pi$  a solution to the vertex-disjoint paths problem (without having these functions explicitly).

For the representative  $\psi$  of each of these classes one should test if there is a path packing function  $\phi_\Pi$  homologous to  $\psi$ . This can be done in polynomial time, by reducing it to the following dual problem.

Given any directed graph  $D = (V, A)$  (not necessarily planar) and any group  $G$ , call two functions  $\phi, \psi : A \rightarrow G$  *cohomologous* if there exists a function  $p : V \rightarrow G$  such that for each arc  $a = (u, w)$  of  $D$  one has:

$$(18) \quad \psi(a) = p(u)^{-1} \phi(a) p(w).$$

Again this is an equivalence relation.

Consider the following *cohomology feasibility problem*:

- (19) given: a directed graph  $D = (V, A)$  and functions  $\phi : A \rightarrow G$  and  $H : A \rightarrow \mathcal{P}(G)$ ;  
 find: a function  $\psi$  cohomologous to  $\phi$  with  $\psi(a) \in H(a)$  for each  $a \in A$ .

This is in its general form an NP-complete problem: when  $G = C_3$  (the group with three elements) and  $\phi(a) = 1$  and  $H(a) = C_3 \setminus \{1\}$  for each arc  $a$ , the problem amounts to the 3-colourability of the vertices of  $D$ . However:

**Theorem 8.** *If  $G$  is the free group and each  $H(a)$  is hereditary, then the cohomology feasibility problem is solvable in polynomial time.*

Here a subset  $H$  of the free group is *hereditary* if for each (reduced) word  $w'ww''$  in  $H$ , also the word  $w$  belongs to  $H$ .

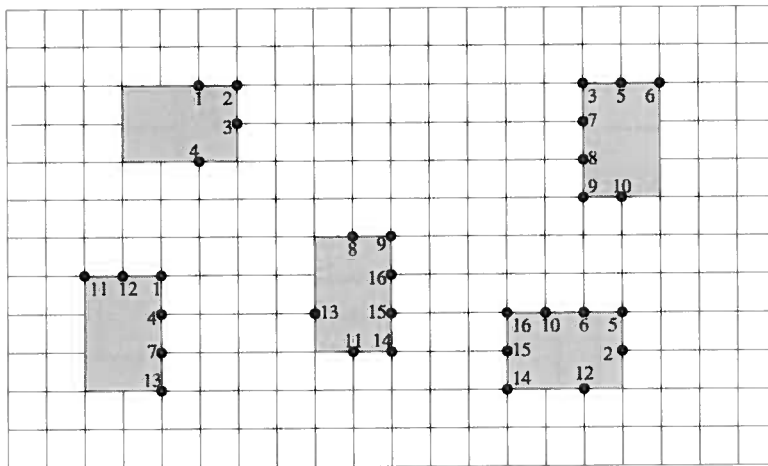
Now the problem of finding a path packing function  $\phi_\Pi$  homologous to a given function  $\psi$ , can be reduced to the cohomology feasibility problem on an

extension of the dual graph of  $D$ , where each  $H(a)$  is equal to  $\{1, g_1, \dots, g_k\}$  or to  $\{1, g_1, g_1^{-1}, \dots, g_k, g_k^{-1}\}$ . This finishes the outline of the proof of Theorem 7.

Theorem 7 can be generalized to disjoint *trees* connecting given sets of vertices, and Theorem 8 can be generalized to *free partially commutative groups* — see [59]. Moreover, necessary and sufficient conditions for the existence of a solution can be described in terms of cycles in the graaf  $D$ .

**9. VLSI-routing.** The approach described above for the vertex-disjoint paths problem in directed planar graphs is analogous to a method developed for the *VLSI-routing problem*. This problem asks for the routes that wires should make on a chip so as to connect certain pairs of pins and so that wires connecting different pairs of pins are disjoint.

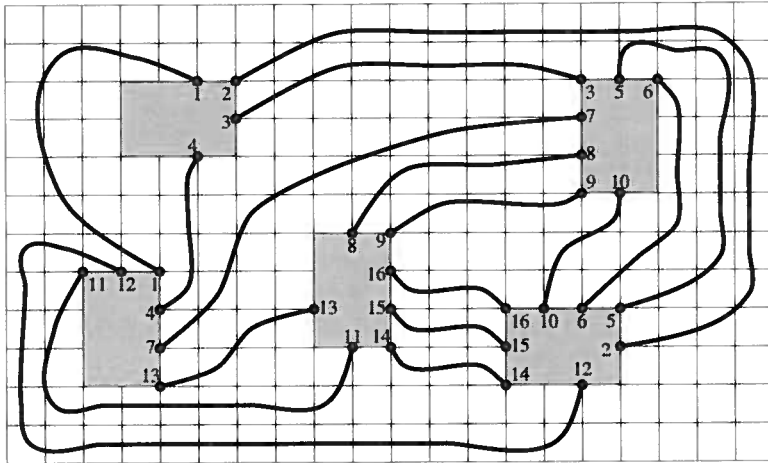
As the routes that the wires potentially can make form a graph, the problem to be solved can be modeled as a disjoint paths problem. Consider an example of such a problem as in Figure 11 — relatively simple, since generally the number of pins to be connected is of the order of several thousands. The grey areas are ‘modules’ on which the pins are located. Points with the same label should be connected.



**Figure 11**

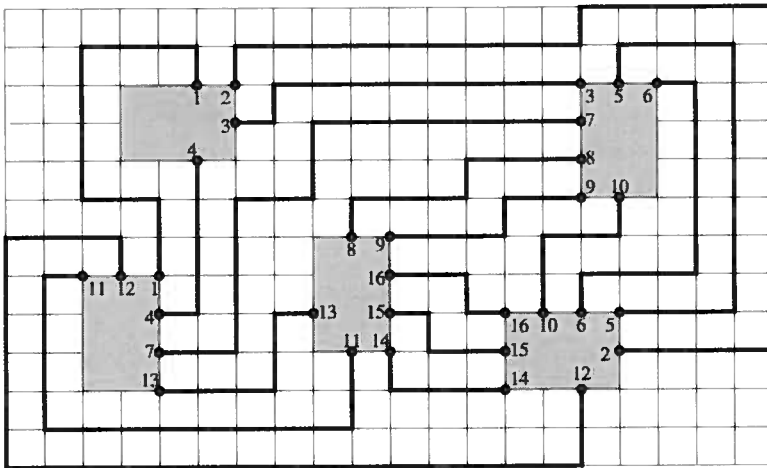
In the example, the graph is a ‘grid graph’, which is typical in VLSI-design since it facilitates the manufacturing of the chip and it ensures a certain minimum distance between disjoint wires. But even for such graphs the disjoint paths problem is NP-complete.

Now the following two-step approach was proposed by Pinter [46]. First choose the homotopies of the wires; for instance like in Figure 12. That is, for each  $i$  one chooses a curve  $C_i$  in the plane connecting the two vertices  $i$ , in such a way that they are pairwise disjoint, and such that the modules are not traversed.



**Figure 12**

Second, try to find disjoint paths  $P_1, \dots, P_k$  in the graph such that  $P_i$  is homotopic to  $C_i$ , in the space obtained from the plane by taking out the rectangles forming the modules. In Figure 13 such a solution is given.



**Figure 13**

It was shown by Leiserson and Maley [35] that this second step can be performed in polynomial time. So the hard part of the problem is the first step: finding the right topology of the layout.

Cole and Siegel [8] proved a Menger-type cut theorem characterizing the existence of a solution in the second step. That is, if there is no solution for the disjoint paths problem given the homotopies, there is an 'oversaturated' cut: a

curve  $D$  connecting two holes in the plane and intersecting the graph less than the number of times  $D$  necessarily crosses the curves  $C_i$ .

This can be used in a heuristic practical algorithm for the VLSI-routing problem: first guess the homotopies of the solution; second try to find disjoint paths of the guessed homotopies; if you find them you can stop; if you don't find them, the oversaturated cut will indicate a bottleneck in the chosen homotopies; amend the bottleneck and repeat.

Similar results hold if one wants to pack trees instead of paths (which is generally the case at VLSI-design), and the result can be extended to any planar graph [56]. As a theoretical consequence one has (by an enumeration argument similar to the one used for Theorem 7):

**Theorem 9.** *For each fixed number of modules, the planar VLSI-routing problem can be solved in polynomial time.*

**10. Railway timetabling.** The cohomology feasibility problem also shows up in the problem of making the timetable for Nederlandse Spoorwegen (Dutch Railways), a project currently performed for NS by CWI (Adri Steenbeek and me). The Dutch railway system belongs to the busiest in the world, with several short distance trajectories, while many connections are offered, with short transfer time.

Task is to provide algorithmic means to decide if a given set of conditions on the timetable can be satisfied. In particular, the hourly pattern of the timetable is considered. The basis of the NS-timetable is a periodic cycle of one hour, so that on each line there is a train at least once an hour.

How can this problem be modeled? First of all, each departure time to be determined is represented by a variable  $v_t$ . Here  $t$  is a train leg that should go every hour once. So  $v_t$  represents a variable in the cyclic group  $C_{60} = \mathbb{Z}/60\mathbb{Z}$ . Similarly, the arrival time is represented by a variable  $a_t$  in  $C_{60}$ .

In the problem considered by us, a fixed running time was assumed for each leg. This implies that if train leg  $t$  has a running time of 11 minutes, then  $a_t - v_t = 11$ . The waiting period of a train in a station is prescribed by an interval. E.g., if  $t$  and  $t'$  are two consecutive train legs of one hourly train, and if it is required that the train stops at the intermediate station for a period of at least 2 and at most 5 minutes, then one poses the condition  $v_{t'} - a_t \in [2, 5]$  (as interval of  $C_{60}$ ).

This gives relations between train legs of one hourly train. To make connections, one has to consider train legs in two different trains. So if one wants to make a connection from leg  $t$ , arriving in Utrecht say, of one train, to a leg  $t'$  departing from Utrecht of another train, so that the transfer time is at least 3 and at most 7 minutes, then one gets the condition  $v_{t'} - a_t \in [3, 7]$ .

Finally, there is the condition that for safety each two trains on the same trajectory should have a timetable distance of at least 3 minutes. That is, if train leg  $t$  of one train and train leg  $t'$  of another train run on the same railway

section, then one should pose the condition  $v_{t'} - v_t \in [3, 57]$ .

By representing each variable by a vertex, the problem can be modeled as follows. Let  $D = (V, A)$  be a directed graph, and for each  $a \in A$ , let  $H(a)$  be an interval on  $C_{60}$ . Find a function  $p : V \rightarrow C_{60}$  such that  $p(w) - p(u) \in H(a)$  for each arc  $a = (u, w)$  of  $D$ .

This is a special case of the cohomology feasibility problem. Note that (as  $C_{60}$  is abelian) one may equivalently find a ‘length’ function  $l : A \rightarrow C_{60}$  such that  $l(a) \in H(a)$  for each  $a \in A$  and such that each undirected circuit in  $D$  has length 0. (For arcs  $a$  in the circuit traversed backward one takes  $-l(a)$  for its length.)

It is not difficult to formulate this problem as an integer linear programming problem. Indeed, if for any arc  $a = (u, w)$ ,  $H(a)$  is equal to the interval  $[l_a, u_a]$ , we can put:

$$(20) \quad l_a \leq x_w - x_u + 60y_a \leq u_a,$$

where  $y_a$  is required to be an integer. Thus we get a system of  $|A|$  linear inequalities with  $|V|$  real variables  $x_v$  and  $|A|$  integer variables  $y_a$ . In fact, if there is a solution, there is also one with the  $x_v$  being integer as well (as the  $x$  variables make a network matrix).

Now in solving (20), one may choose a spanning tree  $T$  in  $D$ , and assume that  $y_a = 0$  for each arc  $a$  in  $T$  (cf. Serafini and Ukovich [60]). Alternatively, one may consider the problem as follows.

A *circulation* is a function  $f : A \rightarrow \mathbb{R}$  such that the ‘flow conservation law’:

$$(21) \quad \sum_{a \in \delta^-(v)} f(a) = \sum_{a \in \delta^+(v)} f(a)$$

holds for each vertex  $v$  of  $D$ . Here  $\delta^-(v)$  and  $\delta^+(v)$  denote the sets of arcs entering  $v$  and leaving  $v$ , respectively.

Let  $L$  be the lattice of all integer-valued circulations. Now one can describe the problem as one of finding a linear function  $\Phi : L \rightarrow \mathbb{Z}$  such that there exist  $z_a$  (for  $a \in A$ ) with the properties that  $l_a \leq z_a \leq u_a$  for each arc  $A$  and  $z^T f = 60\Phi(f)$  for each  $f \in L$ .

The existence of such  $z_a$  can be checked in polynomial time, given the values of  $\Phi$  on a basis of  $L$ . Hence, in a searching for a feasible timetable one can branch on values of  $\Phi$  on an appropriate basis of  $L$ . Given  $\Phi$ , if there exist  $z_a$ , one can optimize the  $z_a$  under any linear (or convex piecewise linear) objective function (for instance, passenger waiting time).

Typically, the problems coming from NS have about 3000 variables with about 10,000 constraints. In a straightforward way they can be reduced to about 200 variables with about 600 constraints. The above observations turn out to require a too heavy framework in order to solve the problem fast in practice (although they are of help in optimizing a given solution).

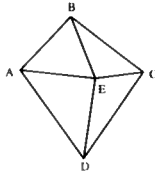
The package CADANS (Combinatorisch-Algebraïsch Dienstregeling-Algorithmen voor de Nederlandse Spoorwegen) that CWI is developing for NS for solving



the problem above, is based on a fast constraint propagation technique and fast branching heuristics designed by Adri Steenbeek. It gives, within time of the order of 1-10 minutes either a solution (i.e., a feasible timetable), or an inclusionwise minimal set of constraints that is infeasible. If CADANS gives the latter answer, the user should drop, or relax, at least one of the constraints in the minimal set in order to make the constraints feasible. Thus CADANS can be used interactively to support the planner. Alternatively, it can uncover bottlenecks in the infrastructure, and indicate where extra infrastructure (viaducts, flyovers, four-tracks) should be built in order to make a given set of conditions feasible.

**11. Transportation and flow problems.** Railway transportation forms a classical source of problems studied in operations research. In 1939, Kantorovich [25] published in Leningrad a monograph called *Mathematical Methods of Organizing and Planning Production*, in which he outlined a new method to maximize a linear function under given linear inequality constraints, thus laying the fundamentals for *linear programming*. He gave the following application:

Let there be several points  $A, B, C, D, E$  which are connected to one another by a railroad network. It is possible to make the shipments from  $B$  to  $D$  by the shortest route  $BED$ , but it is also possible to use other routes as well: namely  $BCD, BAD$ . Let there also be given a schedule of freight shipments; that is, it is necessary to ship from  $A$  to  $B$  a certain number of carloads, from  $D$  to  $C$  a certain number, and so on. The



problem consists of the following. There is given a maximum capacity for each route under the given conditions (it can of course change under new methods of operation in transportation). It is necessary to distribute the freight flows among the different routes in such a way as to complete the necessary shipments with a minimum expenditure of fuel, under the condition of minimizing the empty runs of freight cars and taking account of the maximum capacities of the routes. As was already shown, this problem can also be solved by our methods.

In 1941, Hitchcock [24] formulated another variant of a transportation problem. Independently, during the Second World War, Koopmans was on the staff of the Combined Shipping Adjustment Board (an agency formed by the Allied to coordinate the use of their merchant fleets). Influenced by his teacher Tinbergen (cf. [73]) he was interested in the topic of ship freights and capacities. His task at the Board was the planning of assigning ships to convoys so as to accomplish prescribed deliveries, while minimizing empty voyages (cf. [12]). Koopmans found in 1943 a method for the *transshipment problem*, but due to wartime restrictions he published it only after the war [32].

Koopmans and Reiter [33] investigated the economic implications of the method:

For the sake of definiteness we shall speak in terms of the transportation of cargoes on ocean-going ships. In considering only shipping we do not lose generality of application since ships may be “translated” into trucks, aircraft, or, in first approximation, trains, and ports into the various sorts of terminals. Such translation is possible because all the above examples involve particular types of movable transportation equipment.

The cultural lag of economic thought in the application of mathematical methods is strikingly illustrated by the fact that linear graphs are making their entrance into transportation theory just about a century after they were first studied in relation to electrical networks, although organized transportation systems are much older than the study of electricity.

The breakthrough in linear programming came around 1950 when Dantzig [10] published the *simplex method* for the linear programming problem. The success of the method was caused by a very simple tableau-form and pivoting rule and by the large efficiency in practice. Dantzig also described a direct implementation of the simplex method to the transportation problem ([9]).

In the beginning of the 1950s, T.E. Harris at the RAND Corporation (the think tank of the U.S. Air Force in Santa Monica, California) called attention for the following special case of the problem considered by Kantorovich:

Consider a rail network connecting two cities by way of a number of intermediate cities, where each link of the network has a number assigned to it representing its capacity. Assuming a steady state condition, find a maximal flow from one given city to the other.

This question raised a stream of research at RAND. The problem can be formalized as follows.

Let be given a directed graph  $D = (V, A)$ , with two special vertices, a ‘source’  $s$  and a ‘sink’ or ‘terminal’  $t$ . Then an  $s - t$  flow is a function  $f : A \rightarrow \mathbb{R}_+$  such that for each vertex  $v \neq s, t$  the flow conservation law (21) holds. The value of  $f$  is equal to the net flow leaving  $s$ ; that is:

$$(22) \quad \text{value}(f) := \sum_{a \in \delta^-(s)} f(a) - \sum_{a \in \delta^+(s)} f(a).$$

It is not difficult to prove that this value is equal to the net flow entering  $t$ .

If moreover a ‘capacity’ function  $c : A \rightarrow \mathbb{R}_+$  is given, one says that  $f$  is *subject to  $c$*  if  $f(a) \leq c(a)$  for each arc  $a$ .

Now the *maximum flow problem* can be formulated:

$$(23) \quad \begin{array}{l} \text{given: a directed graph } D = (V, A), \text{ vertices } s, t \in V, \text{ and a ‘capacity’} \\ \text{function } c : A \rightarrow \mathbb{R}_+; \\ \text{find: a flow } f \text{ subject to } c \text{ maximizing } \text{value}(f). \end{array}$$

In their basic paper “Maximal flow through a network” (published as a RAND Report of 19 November 1954), Ford and Fulkerson [17] observed that this

is just a linear programming problem, and hence can be solved with Dantzig's simplex method.

Main result of Ford and Fulkerson's paper is the famous *max-flow min-cut theorem*. To this end, the concept of a *cut* is defined. Let  $U$  is any set with  $s \in U$  and  $t \notin U$ . Then  $\delta^+(U)$  (the set of all arcs leaving  $U$ ) is an  $s-t$  cut. The *capacity* of the cut is the sum of all  $c(a)$  for  $a \in \delta^+(U)$ .

It is clear that the capacity of any cut is an upper bound on the maximal value of  $s-t$  flows. What Ford and Fulkerson [17] showed is:

**Max-flow min-cut theorem.** *The maximal value of the  $s-t$  flows is equal to the minimal capacity of the  $s-t$  cuts.*

Since (as follows from an observation of Dantzig [9]) there is an integer-valued maximum flow if all capacities are integer, an arc-disjoint version of Menger's theorem follows from the max-flow min-cut theorem.

Alternative proofs of the max-flow min-cut theorem were given by Robacker [51] and by Elias, Feinstein, and Shannon [14]. In this last paper it is claimed that the result was known by workers in communication theory:

This theorem may appear almost obvious on physical grounds and appears to have been accepted without proof for some time by workers in communication theory. However, while the fact that this flow cannot be exceeded is indeed almost trivial, the fact that it can actually be achieved is by no means obvious. We understand that proofs of the theorem have been given by Ford and Fulkerson and Fulkerson and Dantzig. The following proof is relatively simple, and we believe different in principle.

The max-flow min-cut theorem being also a combinatorial result, one was interested in obtaining combinatorial methods for finding maximum flows. First, Ford and Fulkerson [17] gave a simple algorithm for the maximal flow problem in case the graph, added with an extra edge connecting  $s$  and  $t$ , is planar.

Next, a heuristic method, the *flooding technique*, was presented by Boldyreff [6] on 3 June 1955 at the New York meeting of the Operations Research Society of America (RAND Report of 5 August 1955). The method was intuitive, and the author did not claim generality:

It has been previously assumed that a highly complex railway transportation system, too complicated to be amenable to analysis, can be represented by a much simpler model. This was accomplished by representing each complete railway operating division by a point, and by joining pairs of such points by arcs (lines) with traffic carrying capacities equal to the maximum possible volume of traffic (expressed in some convenient unit, such as trains per day) between the corresponding operating divisions.

In this fashion, a network is obtained consisting of three sets of points — points of origin, intermediate or junction points, and the terminal points (or points of destination) — and a set of arcs of specified traffic carrying capacities, joining these points to each other.

Boldyreff's arguments for designing a heuristic procedure are formulated as:

In the process of searching for the methods of solving this problem the following objectives were used as a guide:

1. That the solution could be obtained quickly, even for complex networks.
2. That the method could be explained easily to personnel without specialized technical training and used by them effectively.
3. That the validity of the solution be subject to easy, direct verification.
4. That the method would not depend on the use of high-speed computing or other specialized equipment.

Boldyreff's 'flooding technique' pushes a maximum amount of flow greedily through the network. If at some vertex a 'bottleneck' arises (i.e., there are more trains arriving than can be pushed further through the network), it is eliminated by returning the excess trains to the origin. It is empirical, not using backtracking, and not leading to an optimum solution in all cases:

Whenever arbitrary decisions have to be made, ordinary common sense is used as a guide. At each step the guiding principle is to move forward the maximum possible number of trains, and to maintain the greatest flexibility for the remaining network.

Boldyreff speculates that 'in dealing with the usual railway networks a single flooding, followed by removal of bottlenecks, should lead to a maximal flow.' He gives as an example of a complex network, a railway transportation system with 41 vertices and 85 arcs, for which 'the total time of solving the problem is less than thirty minutes.'

Soon after, Ford and Fulkerson presented in a RAND Report of 29 December 1955 [18] their 'very simple algorithm' for the maximum flow problem, based on finding 'augmenting paths'. The algorithm finds in a finite number of steps a maximum flow, if all capacities have rational values. After mentioning the maximum flow problem, they remark:

This is of course a linear programming problem, and hence may be solved by Dantzig's simplex algorithm. In fact, the simplex computation for a problem of this kind is particularly efficient, since it can be shown that the sets of equations one solves in the process are always triangular. However, for the flow problem, we shall describe what appears to be a considerably more efficient algorithm; it is, moreover, readily learned by a person with no special training, and may easily be mechanized for handling large networks. We believe that problems involving more than 500 nodes and 4,000 arcs are within reach of present computing machines.

Ford and Fulkerson's algorithm for the maximum-flow problem formed a breakthrough. It has implementations that require only polynomially bounded running time, as was shown by Dinits [11] and Edmonds and Karp [13]. In the

latter paper, also a polynomial-time algorithm is given for the *minimum-cost* flow problem. It implies a polynomial-time algorithm for the minimum-cost circulation problem.

**12. Routing of railway stock.** The work on the minimum-cost circulation problem can be applied to minimizing the railway stock needed to run a schedule. NS (Nederlandse Spoorwegen) runs an hourly train service on its route Amsterdam - Schiphol Airport - Leyden - The Hague - Rotterdam - Dordrecht - Roosendaal - Middelburg - Vlissingen *vice versa*, with timetable as in Table 1.

train number	2123	2127	2131	2135	2139	2143	2147	2151	2155
Amsterdam V		6.48	7.55	8.56	9.56	10.56	11.56	12.56	13.56
Rotterdam A		7.55	8.58	9.58	10.58	11.58	12.58	13.58	14.58
Rotterdam V	7.00	8.01	9.02	10.03	11.02	12.03	13.02	14.02	15.02
Roosendaal A	7.40	8.41	9.41	10.43	11.41	12.41	13.41	14.41	15.41
Roosendaal V	7.43	8.43	9.43	10.45	11.43	12.43	13.43	14.43	15.43
Vlissingen A	8.38	9.38	10.38	11.38	12.38	13.38	14.38	15.38	16.38
train number	2159	2163	2167	2171	2175	2179	2183	2187	2191
Amsterdam V	14.56	15.56	16.56	17.56	18.56	19.56	20.56	21.56	22.56
Rotterdam A	15.58	16.58	17.58	18.58	19.58	20.58	21.58	22.58	23.58
Rotterdam V	16.00	17.01	18.01	19.02	20.02	21.02	22.02	23.02	
Roosendaal A	16.43	17.43	18.42	19.41	20.41	21.41	22.41	23.54	
Roosendaal V	16.45	17.45	18.44	19.43	20.43	21.43			
Vlissingen A	17.40	18.40	19.39	20.38	21.38	22.38			
train number	2108	2112	2116	2120	2124	2128	2132	2136	2140
Vlissingen V			5.30	6.54	7.56	8.56	9.56	10.56	11.56
Roosendaal A			6.35	7.48	8.50	9.50	10.50	11.50	12.50
Roosendaal V		5.29	6.43	7.52	8.53	9.53	10.53	11.53	12.53
Rotterdam A		6.28	7.26	8.32	9.32	10.32	11.32	12.32	13.32
Rotterdam V	5.31	6.29	7.32	8.35	9.34	10.34	11.34	12.34	13.35
Amsterdam A	6.39	7.38	8.38	9.40	10.38	11.38	12.38	13.38	14.38
train number	2144	2148	2152	2156	2160	2164	2168	2172	2176
Vlissingen V	12.56	13.56	14.56	15.56	16.56	17.56	18.56	19.55	
Roosendaal A	13.50	14.50	15.50	16.50	17.50	18.50	19.50	20.49	
Roosendaal V	13.53	14.53	15.53	16.53	17.53	18.53	19.53	20.52	21.53
Rotterdam A	14.32	15.32	16.32	17.33	18.32	19.32	20.32	21.30	22.32
Rotterdam V	14.35	15.34	16.34	17.35	18.34	19.34	20.35	21.32	22.34
Amsterdam A	15.38	16.40	17.38	18.38	19.38	20.38	21.38	22.38	23.38

**Table 1: Timetable Amsterdam-Vlissingen vice versa**

The trains have more stops, but for our purposes only those given in the table are of interest.

For each of the legs of any scheduled train, Nederlandse Spoorwegen has determined an expected number of (second-class) passengers, given in Table 2. The problem to be solved is: What is the minimum amount of train stock necessary to perform the service in such a way that at each leg there are enough seats?

train number	2123	2127	2131	2135	2139	2143	2147	2151	2155
Amsterdam-Rotterdam		340	616	407	336	282	287	297	292
Rotterdam-Roosendaal	58	272	396	364	240	221	252	267	287
Roosendaal-Vlissingen	328	181	270	237	208	188	180	195	290
train number	2159	2163	2167	2171	2175	2179	2183	2187	2191
Amsterdam-Rotterdam	378	527	616	563	320	184	161	190	123
Rotterdam-Roosendaal	497	749	594	395	254	165	130	77	
Roosendaal-Vlissingen	388	504	381	276	187	136			
train number	2108	2112	2116	2120	2124	2128	2132	2136	2140
Vlissingen-Roosendaal			138	448	449	436	224	177	184
Roosendaal-Rotterdam		167	449	628	397	521	281	214	218
Rotterdam-Amsterdam	61	230	586	545	427	512	344	303	283
train number	2144	2148	2152	2156	2160	2164	2168	2172	2176
Vlissingen-Roosendaal	181	165	225	332	309	164	142	121	
Roosendaal-Rotterdam	174	206	298	422	313	156	155	130	64
Rotterdam-Amsterdam	330	338	518	606	327	169	157	154	143

**Table 2: Numbers of required seats**

In order to answer this question, one should know a number of further characteristics and constraints. In a first variant of the problem considered, the train stock consists of one type of two-way train-units, each consisting of three carriages, and each having 163 seats. Each unit has at both ends an engineer's cabin, and units can be coupled together, up to 15 carriages, that is, 5 train-units.

The train length can be changed, by coupling or decoupling units, at the terminal stations of the line, that is at Amsterdam and Vlissingen, and *en route* at two intermediate stations: Rotterdam and Roosendaal. Any train-unit decoupled from a train arriving at place  $X$  at time  $t$  can be linked up to any other train departing from  $X$  at any time later than  $t$ . (The Amsterdam-Vlissingen schedule is such that in practice this gives enough time to make the necessary switchings.)

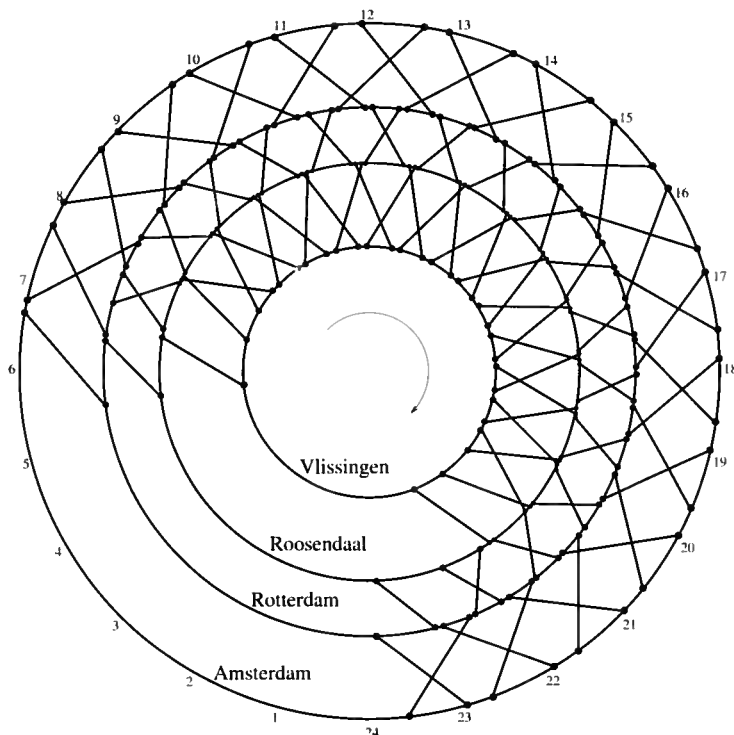
A last condition is that, for logistic reasons, for each place  $X \in \{\text{Amsterdam, Rotterdam, Roosendaal, Vlissingen}\}$ , the number of train-units staying overnight at  $X$  should be constant during the week (but may vary for different places).

Given these problem data and characteristics, one may ask for the minimum number of train-units that should be available to perform the daily cycle of train rides required.

If only one type of railway stock is used, the classical min-cost circulation method can be applied (Bartlett [5], cf. [15], [16], [45], [47], [75]). To this end, a directed graph  $D = (V, A)$  is constructed as follows. For each place  $X \in \{\text{Amsterdam, Rotterdam, Roosendaal, Vlissingen}\}$  and for each time  $t$  at which any train leaves or arrives at  $X$ , we make a vertex  $(X, t)$ . So the vertices of  $D$  correspond to all 198 time entries in the timetable (Table 1).

For any leg of any train, leaving place  $X$  at time  $t$  and arriving at place  $Y$  at time  $t'$ , we make a directed arc from  $(X, t)$  to  $(Y, t')$ . For instance, there is an arc from (Roosendaal, 7.43) to (Vlissingen, 8.38).

Moreover, for any place  $X$  and any two successive times  $t, t'$  at which any train leaves or arrives at  $X$ , we make an arc from  $(X, t)$  to  $(X, t')$ . Thus in our example there will be arcs, e.g., from (Rotterdam, 8.01) to (Rotterdam, 8.32), from (Rotterdam, 8.32) to (Rotterdam, 8.35), from (Vlissingen, 8.38) to (Vlissingen, 8.56), and from (Vlissingen, 8.56) to (Vlissingen, 9.38).



**Figure 14: The graph  $D$ . All arcs are oriented clockwise**

Finally, for each place  $X$  there will be an arc from  $(X, t)$  to  $(X, t')$ , where  $t$  is the last time of the day at which any train leaves or arrives at  $X$  and where  $t'$  is the first time of the day at which any train leaves or arrives at  $X$ . So there is an arc from (Roosendaal, 23.54) to (Roosendaal, 5.29).

We can now describe any possible routing of train stock as a function  $f : A \rightarrow \mathbb{Z}_+$ , where  $f(a)$  denotes the following. If  $a$  corresponds to a leg, then  $f(a)$  is the number of units deployed for that leg. If  $a$  corresponds to an arc from  $(X, t)$  to  $(X, t')$ , then  $f(a)$  is equal to the number of units present at place  $X$  in the time period  $t-t'$  (possibly overnight).

First of all, this function is a *circulation*, that is, the *flow conservation law* (21) holds. Moreover, in order to satisfy the demand and capacity constraints,  $f(a)$  should satisfy  $d(a) \leq f(a) \leq 5$ , where  $d(a)$  is the minimum number of train-units necessary for leg  $a$ , based on the lower bound on seats for leg  $a$ .

Now observe that the total number of units needed, is equal to the total flow on the ‘overnight’ arcs. So if we wish to minimize the total number of units deployed, we could restrict ourselves to minimizing  $\sum_{a \in A^\circ} f(a)$ , where  $A^\circ$  denotes the set of overnight arcs. (So  $|A^\circ| = 4$  in the Amsterdam - Vlissingen example.)

It is easy to see that this fully models the problem. Hence determining the minimum number of train-units amounts to solving a minimum-cost circulation problem, where the cost function is quite trivial: we have  $\text{cost}(a) = 1$  if  $a$  is an overnight arc, and  $\text{cost}(a) = 0$  for all other arcs.

Having this model, we can apply standard min-cost circulation algorithms, based on min-cost augmenting paths and cycles (cf. Ford and Fulkerson [19] and Ahuja, Magnanti, and Orlin [1]). Implementation gives solutions of the problem (for the above data) in about 0.05 CPUseconds on an SGI R4400.

Alternatively, the problem can be solved easily with any linear programming package, since by the integrality of the input data and by the total unimodularity of the underlying matrix the optimum basic solution will have integer values only. With the linear programming package CPLEX (version 2.1) the optimum solution given in Table 3 was obtained again in about 0.05 CPUseconds (on an SGI R4400):

train number	2123	2127	2131	2135	2139	2143	2147	2151	2155
Amsterdam-Rotterdam		3	4	3	3	2	2	2	2
Rotterdam-Roosendaal	1	2	3	3	2	2	2	2	2
Roosendaal-Vlissingen	3	2	2	2	2	2	2	2	2
train number	2159	2163	2167	2171	2175	2179	2183	2187	2191
Amsterdam-Rotterdam	5	5	4	4	2	2	1	2	1
Rotterdam-Roosendaal	4	5	4	3	2	2	1	1	
Roosendaal-Vlissingen	3	4	3	2	2	1			
train number	2108	2112	2116	2120	2124	2128	2132	2136	2140
Vlissingen-Roosendaal			1	3	3	3	2	2	2
Roosendaal-Rotterdam		2	4	4	3	4	2	2	2
Rotterdam-Amsterdam	1	2	4	4	3	4	3	2	2
train number	2144	2148	2152	2156	2160	2164	2168	2172	2176
Vlissingen-Roosendaal	2	2	2	3	2	2	1	4	
Roosendaal-Rotterdam	2	3	2	4	3	1	1	1	1
Rotterdam-Amsterdam	3	3	4	4	3	2	1	1	1

**Table 3: Minimum circulation with one type of stock**

Required are 22 units, divided during the night over Amsterdam: 4, Rotterdam: 2, Roosendaal: 8, and Vlissingen: 8.



It is quite direct to modify and extend the model. Instead of minimizing the number of train-units one can minimize the amount of carriage-kilometers that should be made every day, or any linear combination of both quantities. In addition, one can put an upper bound on the number of units that can be stored at any of the stations.

Instead of considering one line only, one can more generally consider *networks* of lines that share the same railway stock, including trains that are scheduled to be split or combined. (Nederlandse Spoorwegen has trains from The Hague and Rotterdam to Leeuwarden and Groningen that are combined to one train on the common trajectory between Utrecht and Zwolle.)

If only one type of unit is employed for that part of the network, each unit having the same capacity, the problem can be solved fast even for large networks.

**13. Two types of stock.** The problem becomes harder if there are several types of trains that can be deployed for the train service. Clearly, if for each scheduled train we would prescribe which type of unit should be deployed, the problem could be decomposed into separate problems of the type above. But if we do not make such a prescription, and if some of the types can be coupled together to form a train of mixed composition, we should extend the model to a ‘multi-commodity circulation’ model.

Let us restrict ourselves to the case Amsterdam-Vlissingen again, where now we can deploy two types of two-way train-units, that can be coupled together. The two types are type IC3, each unit of which consists of 3 carriages and has 163 seats, and type IC4, each unit of which consists of 4 carriages and has 218 seats.

Again, the demands of the train legs are given in Table 2. The maximum number of carriages that can be in any train again is 15. This means that if a train consists of  $x$  units of type IC3 and  $y$  units of type IC4 then  $3x + 4y \leq 15$  should hold.

It is quite easy to extend the model above to the present case. Again we consider the directed graph  $D = (V, A)$  as above. At each arc  $a$  let  $f(a)$  be the number of units of type IC3 on the leg corresponding to  $a$  and let  $g(a)$  similarly represent type IC4. So both  $f : A \rightarrow \mathbb{Z}_+$  and  $g : A \rightarrow \mathbb{Z}_+$  are circulations, that is, satisfy the flow circulation law:

$$(24) \quad \begin{aligned} \sum_{a \in \delta^-(v)} f(a) &= \sum_{a \in \delta^+(v)} f(a), \\ \sum_{a \in \delta^-(v)} g(a) &= \sum_{a \in \delta^+(v)} g(a), \end{aligned}$$

for each vertex  $v$ . The capacity constraint now is:

$$(25) \quad 3f(a) + 4g(a) \leq 15$$

for each arc  $a$  representing a leg. The demand constraint can be formulated as follows:

$$(26) \quad 163f(a) + 218g(a) \geq p(a),$$

for each arc  $a$  representing a leg, where  $p(a)$  denotes the number of seats required (Table 2). Note that in contrary to the case of one type of unit, now we cannot speak of a minimum number of units required: now there are two dimensions, so that minimum train compositions need not be unique.

If  $\text{cost}_{\text{IC3}}$  and  $\text{cost}_{\text{IC4}}$  represent the cost of purchasing one unit of type IC3 and of type IC4, respectively, then the problem is to find  $f$  and  $g$  so as to

$$(27) \quad \text{minimize} \sum_{a \in A^p} (\text{cost}_{\text{IC3}}f(a) + \text{cost}_{\text{IC4}}g(a)).$$

The classical min-cost circulation algorithms do not apply now. Moreover, when solving the problem as a linear programming problem, we lose the pleasant phenomenon observed above that we automatically would obtain an optimum solution  $f, g : A \rightarrow \mathbb{R}$  with *integer* values only.

So the problem is an integer linear programming problem, with 198 integer variables. Solving the problem in this form with the integer programming package CPLEX (version 2.1) would give (for the Amsterdam-Vlissingen example) a running time of several hours, which is too long, for instance when one wishes to compare several problem data.

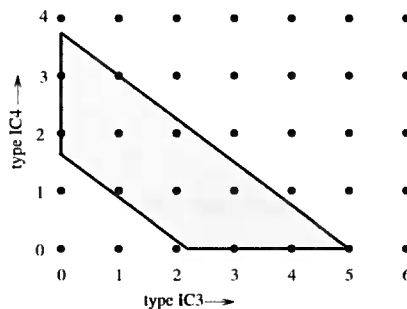
However, there are ways of speeding up the process, by sharpening the constraints and by exploiting more facilities offered by CPLEX. The conditions (25) and (26) can be sharpened in the following way. For each arc  $a$  representing a leg, the two-dimensional vector  $(f(a), g(a))$  should be an integer vector in the polygon

$$(28) \quad P_a := \{(x, y) | x \geq 0, y \geq 0, 163x + 218y \geq p(a), 3x + 4y \leq 15\}.$$

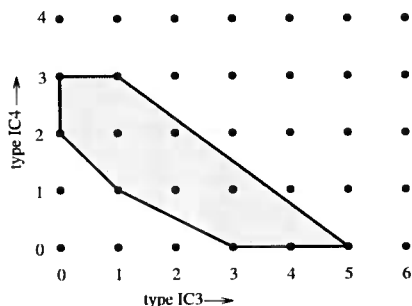
For instance, the trajectory Rotterdam-Amsterdam of train 2132 gives the polygon

$$(29) \quad P_a = \{(x, y) | x \geq 0, y \geq 0, 163x + 218y \geq 344, 3x + 4y \leq 15\}.$$

In a picture:



In a sense, the inequalities are too wide. The constraints given in (29) could be tightened so as to describe exactly the convex hull of the integer vectors in the polygon  $P_a$  (the ‘integer hull’), as in:



Thus for segment Rotterdam-Amsterdam of train 2132 the constraints in (29) can be sharpened to:

$$(30) \quad x \geq 0, y \geq 0, x + y \geq 2, x + 2y \geq 3, y \leq 3, 3x + 4y \leq 15.$$

Doing this for each of the 99 polygons representing a leg gives a sharper set of inequalities, which helps to obtain more easily an integer optimum solution from a fractional solution. (This is a weak form of application of the technique of *polyhedral combinatorics*.) Finding all these sharpened inequalities can be done in a pre-processing phase, and takes about 0.04 CPUseconds.

Implementation of these techniques makes that CPLEX gives a solution to the Amsterdam-Vlissingen problem in 1.58 CPUseconds — see Table 4.

train number	2123	2127	2131	2135	2139	2143	2147	2151	2155
Amsterdam-Rotterdam		0+2	0+3	4+0	0+2	0+2	1+2	0+2	1+1
Rotterdam-Roosendaal	0+1	0+2	0+2	4+0	0+2	0+2	1+3	0+3	1+1
Roosendaal-Vlissingen	0+2	0+2	0+2	2+0	0+1	0+1	0+2	0+2	2+0
train number	2159	2163	2167	2171	2175	2179	2183	2187	2191
Amsterdam-Rotterdam	0+3	2+1	0+3	1+2	0+2	0+1	1+2	0+1	0+1
Rotterdam-Roosendaal	0+3	2+2	0+3	0+2	1+1	2+0	1+3	1+0	
Roosendaal-Vlissingen	0+2	2+1	0+2	0+2	2+0	0+1			
train number	2108	2112	2116	2120	2124	2128	2132	2136	2140
Vlissingen-Roosendaal			1+0	0+3	1+2	0+2	0+2	0+1	1+1
Roosendaal-Rotterdam		1+2	3+0	0+3	0+2	1+2	0+2	2+1	1+3
Rotterdam-Amsterdam	0+1	0+2	4+0	0+3	0+3	1+2	0+2	2+0	0+2
train number	2144	2148	2152	2156	2160	2164	2168	2172	2176
Vlissingen-Roosendaal	1+1	0+1	0+2	0+2	2+0	0+2	2+0	0+1	
Roosendaal-Rotterdam	0+1	0+3	1+3	0+3	1+1	0+1	2+2	0+1	1+0
Rotterdam-Amsterdam	1+1	0+3	1+2	0+3	1+1	0+1	0+2	0+1	0+1

**Table 4: Minimum circulation with two types of stock**

In this table  $x + y$  means:  $x$  units of type IC3 and  $y$  units of type IC4. In total, one needs 7 units of type IC3 and 12 units of type IC4, divided during the night as in Table 5.

	number of units of type IC3	number of units of type IC4	total number of units	total number of carriages
Amsterdam	0	2	2	8
Rotterdam	0	2	2	8
Roosendaal	3	3	6	21
Vlissingen	2	5	7	26
Total	5	12	17	63

**Table 5: Required stock (two types)**

So compared with the solution for one type only, the possibility of having two types gives both a decrease in the number of train-units (17 instead of 22) and in the number of carriages (63 instead of 66).

Our research for NS in fact has focused on more extended problems that require more complicated models and techniques. One requirement is that in any train ride Amsterdam-Vlissingen there should be at least one unit that makes the whole trip. Moreover, it is required that, at any of the four stations given (Amsterdam, Rotterdam, Roosendaal, Vlissingen) one may either couple units to or decouple units from a train, but not both simultaneously. Moreover, one may couple fresh units only to the front of the train, and decouple laid off units only from the rear. (One may check that these conditions are not met by all trains in the solution given in Table 4.)

This all causes that the order of the different units in a train does matter, and that conditions have a more global impact: the order of the units in a certain morning train can still influence the order in some evening train. This does not fit directly in the circulation model described above, and requires a combinatorial extension.

## References

- [1] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, *Network Flows — Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.
- [2] D. Archdeacon, *A Kuratowski Theorem for Projective Planes*, Ph.D. Thesis, Ohio State University, Columbus, Ohio, 1980.
- [3] D. Archdeacon, P. Huneke, A Kuratowski theorem for non-orientable surfaces, *Journal of Combinatorial Theory, Series B* 46 (1989) 173–231.
- [4] Ch. Babbage, On a method of expressing by signs the actions of machinery, *Philosophical Transactions of the Royal Society of London* (1826) 250–265.
- [5] T.E. Bartlett, An algorithm for the minimum number of transport units to maintain a fixed schedule, *Naval Research Logistics Quarterly* 4 (1957) 139–149.

- [6] A.W. Boldyreff, Determination of the maximal steady state flow of traffic through a railroad network, *Journal of the Operations Research Society of America* 3 (1955) 443–465.
- [7] N. Bourbaki, *Groupes et algèbres de Lie*, Hermann, Paris, 1968.
- [8] R. Cole, A. Siegel, River routing every which way, but loose, in: *Proceedings of the 25th Annual Symposium on Foundations of Computer Science*, IEEE, 1984, pp. 65–73.
- [9] G.B. Dantzig, Application of the simplex method to a transportation problem, in: *Activity Analysis of Production and Allocation — Proceedings of a Conference* (Tj.C. Koopmans, ed.), John Wiley & Sons, New York, 1951, pp. 359–373.
- [10] G.B. Dantzig, Maximization of a linear function of variables subject to linear inequalities, in: *Activity Analysis of Production and Allocation — Proceedings of a Conference* (Tj.C. Koopmans, ed.), John Wiley & Sons, New York, 1951, pp. 339–347.
- [11] E.A. Dinits, Algorithm for solution of a problem of maximum flow in a network with power estimation (in Russian), *Doklady Akademii Nauk SSSR* 194 (1970) 754–757 [English translation: *Soviet Mathematics Doklady* 11 (1970) 1277–1280].
- [12] R. Dorflman, The discovery of linear programming, *Annals of the History of Computing* 6 (1984) 283–295.
- [13] J. Edmonds, R.M. Karp, Theoretical improvements in algorithmic efficiency for network flow problems, *Journal of the Association for Computing Machinery* 19 (1972) 248–264.
- [14] P. Elias, A. Feinstein, C.E. Shannon, A note on the maximum flow through a network, *IRE Transactions on Information Theory* IT-2 (1956) 117–119.
- [15] G.J. Feeney, The empty boxcar distribution problem, *Proceedings of the First International Conference on Operational Research (Oxford 1957)*, M. Davies, R.T. Eddison, T. Page, eds., Operations Research Society of America, Baltimore, Maryland, 1957, pp. 250–265.
- [16] A.R. Ferguson, G.B. Dantzig, The problem of routing aircraft, *Aeronautical Engineering Review* 14 (1955) 51–55.
- [17] L.R. Ford, Jr, D.R. Fulkerson, Maximal flow through a network, *Canadian Journal of Mathematics* 8 (1956) 399–404.
- [18] L.R. Ford, Jr, D.R. Fulkerson, A simple algorithm for finding maximal network flows and an application to the Hitchcock problem, *Canadian Journal of Mathematics* 9 (1957) 210–218.
- [19] L.R. Ford, Jr, D.R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, New Jersey, 1962.
- [20] S. Fortune, J. Hopcroft, J. Wyllie, The directed subgraph homeomorphism problem, *Theoretical Computer Science* 10 (1980) 111–121.
- [21] M. Geck, G. Pfeiffer, On the irreducible characters of Hecke algebras, *Advances in Mathematics* 102 (1993) 79–94.
- [22] M. de Graaf, A. Schrijver, *Making curve systems minimally crossing by Reidemeister moves*, preprint, 1994.
- [23] B. Grünbaum, *Convex Polytopes*, Wiley-Interscience, London, 1967.

- [24] F.L. Hitchcock, The distribution of a product from several sources to numerous localities, *Journal of Mathematics and Physics* 20 (1941) 224–230.
- [25] L.V. Kantorovich, *Mathematical Methods of Organizing and Planning Production* (in Russian), Publication House of the Leningrad State University, Leningrad, 1939 [English translation: *Management Science* 6 (1959-60) 366–422].
- [26] R.M. Karp, On the computational complexity of combinatorial problems, *Networks* 5 (1975) 45–68.
- [27] L.H. Kauffman, State models and the Jones polynomial, *Topology* 26 (1987) 395–407.
- [28] B. Knaster, Sui punti regolari nelle curvi di Jordan, in: *Atti del Congresso Internazionale dei Matematici* [Bologna 3–10 Settembre 1928] Tomo II, Nicola Zanichelli, Bologna, [1930], pp. 225–227.
- [29] D. König, Graphok és matrixok (Hungarian) [Graphs and matrices], *Matematikai és Fizikai Lapok* 38 (1931) 116–119.
- [30] D. König, Über trennende Knotenpunkte in Graphen (nebst Anwendungen auf Determinanten und Matrizen), *Acta Litterarum ac Scientiarum Regiae Universitatis Hungaricae Franciscus-Josephinae, Sectio Scientiarum Mathematicarum [Szeged]* 6 (1932-4) 155–179.
- [31] D. König, *Theorie der endlichen und unendlichen Graphen*, Akademische Verlagsgesellschaft, Leipzig, 1936 [reprinted: Chelsea, New York, 1950].
- [32] Tj.C. Koopmans, Optimum utilization of the transportation system, in: *The Econometric Society Meeting* (Washington, D.C., 1947; D.H. Leavens, ed.), 1948, pp. 136–146 [reprinted in: *Econometrica* 17 (Supplement) (1949) 136–146] [reprinted in: *Scientific Papers of Tjalling C. Koopmans*, Springer, Berlin, 1970, pp. 184–193].
- [33] Tj.C. Koopmans, S. Reiter, A model of transportation, in: *Activity Analysis of Production and Allocation — Proceedings of a Conference* (Tj.C. Koopmans, ed.), John Wiley & Sons, New York, 1951, pp. 222–259.
- [34] K. Kuratowski, Sur le problème des courbes gauches en topologie, *Fundamenta Mathematicae* 15 (1930) 271–283.
- [35] C.E. Leiserson, F.M. Maley, Algorithms for routing and testing routability of planar VLSI-layouts, in: *Proceedings of the 17th Annual ACM Symposium on the Theory of Computing*, ACM, 1985, pp. 69–78.
- [36] S. Lins, A minimax theorem on circuits in projective graphs, *Journal of Combinatorial Theory, Series B* 30 (1981) 253–262.
- [37] J.B. Listing, Vorstudien zur Topologie, *Göttinger Studien* (1847) 811–875.
- [38] J.F. Lynch, The equivalence of theorem proving and the interconnection problem, (*ACM*) *SIGDA Newsletter* 5 (1975) 3:31–36.
- [39] W.W. Menasco, M.B. Thistlethwaite, The Tait flyping conjecture, *Bulletin of the American Mathematical Society* 25 (1991) 403–412.
- [40] K. Menger, Zur allgemeinen Kurventheorie, *Fundamenta Mathematicae* 10 (1927) 96–115.
- [41] K. Menger, *Kurventheorie*, Teubner, Leipzig, 1932 [reprinted: Chelsea, New York, 1967].

- [42] K. Menger, On the origin of the  $n$ -arc theorem, *Journal of Graph Theory* 5 (1981) 341–350.
- [43] K. Murasugi, Jones polynomials and classical conjectures in knot theory, *Topology* 26 (1987) 187–194.
- [44] K. Murasugi, Jones polynomials and classical conjectures in knot theory. II, *Mathematical Proceedings of the Cambridge Philosophical Society* 102 (1987) 317–318.
- [45] A.R.D. Norman, M.J. Dowling, *Railroad Car Inventory: Empty Woodrack Cars on the Louisville and Nashville Railroad*, Technical Report 320-2926, IBM New York Scientific Center, New York, 1968.
- [46] R.Y. Pinter, River routing: methodology and analysis, in: *Third CalTech Conference on Very Large Scale Integration*, Springer, Berlin, 1983, pp. 141–163.
- [47] J.W.H.M.T.S.J. van Rees, Een studie omtrent de circulatie van materieel, *Spoor-en Tramwegen* 38 (1965) 363–367.
- [48] K. Reidemeister, Elementare Begründung der Knotentheorie, *Abhandlungen aus dem mathematischen Seminar der Hamburgischen Universität* 5 (1926/1927) 24–32.
- [49] K. Reidemeister, *Knotentheorie*, Springer, Berlin, 1932.
- [50] G. Ringel, Teilungen der Ebene durch Geraden oder topologische Geraden, *Mathematische Zeitschrift* 64 (1955), 79–102.
- [51] J.T. Robacker, *On Network Theory*, Report RM-1498, The RAND Corporation, Santa Monica, California, [May 26] 1955.
- [52] N. Robertson, P.D. Seymour, *Graph minors. XIII. The disjoint paths problem*, preprint, 1986 (revised 1994).
- [53] N. Robertson, P.D. Seymour, *Graph minors. XV. Wagner's conjecture*, preprint, 1988.
- [54] A. Schrijver, *Theory of Linear and Integer Programming*, Wiley, Chichester, 1986.
- [55] A. Schrijver, Disjoint circuits of prescribed homotopies in a graph on a compact surface, *Journal of Combinatorial Theory, Series B* 51 (1991) 127–159.
- [56] A. Schrijver, Disjoint homotopic paths and trees in a planar graph, *Discrete & Computational Geometry* 6 (1991) 527–574.
- [57] A. Schrijver, Tait's flyping conjecture for well-connected links, *Journal of Combinatorial Theory, Series B* 58 (1993) 65–146.
- [58] A. Schrijver, Finding  $k$  disjoint paths in a directed planar graph, *SIAM Journal on Computing* 23 (1994) 780–788.
- [59] A. Schrijver, *Free partially commutative groups, cohomology, and paths and circuits in directed graphs on surfaces*, preprint, 1994.
- [60] P. Serafini, W. Ukovich, A mathematical model for periodic scheduling problems, *SIAM Journal on Discrete Mathematics* 2 (1989) 550–581.
- [61] P.G. Tait, Applications of the theorem that two closed plane curves intersect an even number of times, *Proceedings of the Royal Society of Edinburgh* 9 (1875–1878) 237–246.
- [62] P.G. Tait, Note on the measure of beknottedness, *Proceedings of the Royal Society of Edinburgh* 9 (1875–1878) 289–298.
- [63] P.G. Tait, On knots, *Proceedings of the Royal Society of Edinburgh* 9 (1875–1878) 306–317.

- [64] P.G. Tait, Sevenfold knottiness, *Proceedings of the Royal Society of Edinburgh* 9 (1875–1878) 363–366.
- [65] P.G. Tait, On knots, *Transactions of the Royal Society of Edinburgh* 28 (1877) 145–190.
- [66] P.G. Tait, Some elementary properties of closed plane curves, *The Messenger of Mathematics* 6 (1877) 132–133.
- [67] P.G. Tait, Listing's Topologie, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 17 (1884) 30–46.
- [68] P.G. Tait, On knots. Part II, *Transactions of the Royal Society of Edinburgh* 32 (1884–1885) 327–342.
- [69] M.B. Thistlethwaite, A spanning tree expansion of the Jones polynomial, *Topology* 26 (1987) 297–309.
- [70] M.B. Thistlethwaite, Kauffman's polynomial and alternating links, *Topology* 27 (1988) 311–318.
- [71] M.B. Thistlethwaite, On the Kauffman polynomial of an adequate link, *Inventiones Mathematicae* 93 (1988) 285–296.
- [72] W. Thomson, Vortex statics, *Proceedings of the Royal Society of Edinburgh* 9 (1875–1878) 59–73.
- [73] J. Tinbergen, Scheepsruimte en vrachten, *De Nederlandsche Conjunctuur* (1934) maart 23–35.
- [74] K. Wagner, Über eine Eigenschaft der ebene Komplexe, *Mathematische Annalen* 114 (1937) 570–590.
- [75] W.W. White, A.M. Bomberault, A network algorithm for empty freight car allocation, *IBM Systems Journal* 8 (1969) 147–169.
- [76] H. Whitney, Congruent graphs and the connectivity of graphs, *American Journal of Mathematics* 54 (1932) 150–168.



# Data Mining: Exploratory Data Analysis on Very Large Databases

*To professor Baayen at the occasion of his retirement*

Arno Siebes (arno@cwi.nl)

CWI

Artificial Intelligence and Database research are recognised parents of Data Mining research. Statistics is only considered related in as far as it allows the assessment of the quality of the results of mining. In this expository paper it is shown that Statistics can lay legitimate claims of parenthood. More in particular, it is shown how Data Mining can be seen naturally as a generalisation of both Projection Pursuit and Cluster Analysis. Subsequently it is discussed how this link can help to give Data Mining firm mathematical foundations.

## 1 INTRODUCTION

One of the younger branches of Computer Science, called *Data Mining* or *Knowledge Discovery*, was born out of a, partial, merger of Database and Artificial Intelligence research.

### 1.1 What is Data Mining?

The goal of Data Mining is to discover information in large databases. Both large and small organisations have set up and maintained databases for years, often for pure accounting reasons. The mountains of data accumulated this way, form potential treasure-troves of strategic information.

For example, consider an insurance company. From your own car insurance policy you can deduce that such a company does not associate the same risk with all of its clients. Rather, this risk depends on where you live, your type of car, your age, and many other factors. If the insurance company has registered all the relevant information of its insureds in databases, it should be able to derive precise rules that tell which risk to assign to which client. The derivation of such *risk-profiles* is an example of data mining.

Many production processes are partly or completely automated. A side effect of this automation is that many aspects of the production process, such as the

quality of the end product and the parameter settings of the machinery along the way, are recorded electronically. The optimal parameter settings, those that one can be confident of the quality of the end product, are hidden in these databases. Data mining intends to facilitate unearthing this knowledge.

## 1.2 The Roots

Data Mining is based on techniques inherited from both AI and database research. Statistics is used to assess the validity of the results. The roots of Data Mining in these three areas is discussed briefly in this subsection. In the last part, on Statistics, the goal of this paper is set out.

### 1.2.1 Artificial Intelligence

The AI parent of Data Mining is without a shadow of doubt *Machine Learning*. One of the aspects of intelligent beings is that they adapt their behaviour to their environment. So, it is only natural that early AI researchers developed systems that mimicked this behaviour.

One of the oldest examples of such systems is the *Perceptron* by Rosenblatt [36], a system for *pattern recognition*. The object of pattern recognition is to sort patterns into different classes so that patterns which belong to a class share features. If we call the set of all possible feature combinations the feature space, Perceptron performs well for those patterns that are linearly separable in feature space. Minsky and Pappert showed the limitations of the Perceptron if the patterns are not linearly separable in [29]. *Neural networks* are a way to overcome these limitations, see, e.g., [10] for an introduction in this area.

Neural networks are by far not the only attempt at building learning automata. In fact, an overview of machine learning research is far beyond the scope of this article, if not beyond the scope of a single book. The interested reader is referred to the collection of papers bundled in [38] and the books edited by Michalsky, [27, 28, 23] to get a feeling for the area. The more theoretically inclined reader might enjoy [1].

The most important development in machine learning for current Data Mining research is the introduction of rule induction systems, [14]. Rule induction is similar to neural networks in that it seeks to separate patterns. The major difference is that it separates using *descriptions* rather than weights in a network. The descriptions are expressions in the attributes or features of the objects. Hence, the results of rule induction are directly interpretable by human beings.

### 1.2.2 Databases

The problems in database research that gave rise to Data Mining are more diverse and less well-documented than those in AI. Rather than attempting to describe briefly all these seemingly unrelated problem areas, we describe one in somewhat more detail.

One of the main problems in maintaining large data sets, electronically or otherwise, is to keep them error-free. One of the contributions of database research towards the resolution of this problem is the notion of *integrity constraints*. The constraints on a database describe which entries in a database and which database states are to be considered legal. The more accurate the constraints are, the more errors at, say, data entry can be obviated.

The traditional way to discover constraints is to elicit them from domain experts. Thus, inherently, there is the risk that some constraints are missed. One way to alleviate this risk is by searching for additional constraints when the database is in existence. By confronting the domain experts with constraints that are satisfied by the current database state, these missing constraints can be identified. Pioneering papers in this area are [30, 2, 3].

In theory, a constraint is simply a logical expression. In practice, however, database management systems support only the enforcement of a restricted set of constraints, such as *functional dependencies*. In table  $R$  attribute  $A$  functionally determines attribute  $B$ , denoted by  $A \rightarrow B$ , if whenever two entries share the same  $A$ -value they also share the same  $B$ -value.

For this restricted class of constraints, the problem is solved. Efficient algorithms can be found in, e.g., [24]. While Manilla gives precise bounds on the sample sizes needed to conclude the constraints with sufficient confidence in [21].

### 1.2.3 Statistics

As should be clear from the examples given before, Data Mining is based on *inductive inference*. In other words generalities, such as *rules* or *laws*, are induced from a finite number of examples. Such a conclusion is, of course, never *logical*, the logical conclusions can be inferred using *deduction*. The epistemological problems of induction and its conclusions have been discussed by philosophers since at least the time of Hume. Some interesting points of view pertaining these problems can be found in [12].

Since a long time, Statistics is the most successful approach to assess the validity of inductive conclusions. It is therefore to be expected that Statistics is used in Data Mining precisely for this reason. In other words, Statistics is related to Data Mining.

However, Statistics offers more than is currently used. An introductory course in Statistics and Probability is sufficient to read almost all the literature on Data Mining. Curiously, all Statistics that comes under the name of *Exploratory Data Analysis* is absent in these requirements.

It is the intention of this paper to show that Statistics is more than related to data mining. It could have been, and perhaps should be considered as, one of its parents. More in particular it is shown that Data Mining can be seen as a natural generalisation of a statistical techniques known as *Projection Pursuit* and *Cluster Analysis*

### 1.3 A Roadmap

The object of this paper is expository, the reader is neither expected to be a statistician nor a data miner. The only new fact in this paper is the surprisingly strong link between Exploratory Data Analysis techniques and Data Mining.

In Section 2, we give a brief review of the classical techniques Regression Analysis, Principle Component Analysis and Cluster Analysis. In the next section Projection Pursuit is introduced and, following Huber [15] it is shown how this subsumes the first two techniques of Section 2.

In the fourth section Data Mining is defined and it is shown how it generalises both Projection Pursuit and Cluster Analysis. In Section 5, the contribution of AI and databases is discussed in the light of this new viewpoint.

In the final section of this paper it is discussed how this link might help to give data mining firm mathematical underpinnings. Since the discovery of such underpinnings needs guidance from experimentation, the architecture of a data mine tool is also briefly discussed.

## 2 CLASSICAL EXPLORATORY DATA ANALYSIS

In many laboratory experiments parameters can be individually set. Consequently, hypotheses underlying these experiments can be tested with straightforward statistical techniques. Not all sciences are so lucky, however. In the life sciences and in the social sciences the parameters cannot even be set by the scientist. To analyse this kind of data Statistics developed *Multivariate Analysis*.

Tukey coined the name *Exploratory Data Analysis* (EDA) [40] for, a subset of, these techniques to indicate that the analysis is only part of the work. The interpretation of the results, the formulation of hypotheses and their subsequent testing are equally important. Since I agree with this observation, I have adopted this catchy name.

In this section three “classical” techniques, Regression Analysis, Principle Component Analysis, and Cluster Analysis, are briefly reviewed. The motivation for this section is twofold. In the first place it may serve as a reminder for the average data miner. In the second place, the power of Projection Pursuit is argued in the next section by discussing how it subsumes the first two techniques. Subsequently it is argued that Data Mining subsumes both Projection Pursuit and Cluster Analysis. Far more information on EDA can be found in standard textbooks such as [40, 25].

### 2.1 Regression Analysis

Suppose that the insurance company from the introduction has  $d$  real valued attributes in its clients database. If it assumes that, say, the expected claim amount is a function of these variables, it can use Regression to determine this function.

In formal terminology, let  $(X, Y)$  be a pair of random variables such that  $X$  is  $\mathcal{R}^d$  valued while  $Y$  is  $\mathcal{R}$  valued. The problem is to estimate the *response*

surface

$$f(x) = E(Y|X = x)$$

from  $n$  observations  $(X_1, Y_1), \dots, (X_n, Y_n)$  of  $(X, Y)$ .

A simple way to fit a function to these  $n$  observations is through *least squares estimation*. First a parametric form for  $f$  is chosen, e.g., if  $f$  is assumed to be a linear surface, we have  $f(\vec{x}) = \sum_{i=1}^n a_i x_i + a_0$ . Following, the parameters  $a_i$  are estimated by minimising

$$\sum_{i=1}^n (Y_i - f(X_i))^2.$$

This can be generalised by assuming  $Y$  to be  $\mathcal{R}^k$  valued rather than  $\mathcal{R}$  valued. If  $f$  is then assumed to be linear we get what is known as *multivariate regression*. A generalised least squares estimation exists for this case.

Regression analysis is an example of EDA, if only because one can try different parametric forms for  $f$  and choose the one that fits best. Of course, the number of parameters should be small compared to the number of observations. In the terminology of Machine Learning, one should beware of *overfitting*.

## 2.2 Principle Component Analysis

With Principle Component Analysis (PCA), one hopes to explain most of the variability in the data using only the *principle components* with the highest variability. In other words, PCA is a technique to reduce the dimensionality of the data.

Let  $X$  be an  $\mathcal{R}^d$  valued random variable and let  $X_1, \dots, X_n$  be a set of  $n$  observations of  $X$ . In statistical terminology,  $(X_1, \dots, X_n)^T$  is a data matrix. For example, we have a group of  $n$  students who all participated in  $d$  examinations and  $X_{ij}$  denotes the score of student  $i$  for examination  $j$ . The sample mean vector  $\bar{X}$  is simple defined by

$$\left( \frac{1}{n} \sum_{i=1}^n X_{i1}, \dots, \frac{1}{n} \sum_{i=1}^n X_{id} \right)^T,$$

In other words,  $\bar{X}_i$  denotes the mean score for examination  $i$ . The sample covariance matrix is the  $d \times d$  matrix  $S$  with entries

$$s_{ij} = \frac{1}{n} \sum_{r=1}^n (X_{ri} - \bar{X}_i)(X_{rj} - \bar{X}_j).$$

The covariance matrix  $S$  can be written in the form  $S = GLG^T$  in which  $G$  is an orthogonal matrix and  $L$  a diagonal matrix of the eigenvalues of  $S$ , with  $l_1 \geq l_2 \geq \dots \geq l_p \geq 0$ .

The principle component transformation is defined by rotation

$$W = (G^T(X_1 - \bar{X}), \dots, G^T(X_n - \bar{X}))$$

the columns of  $W$  represent *uncorrelated* linear combinations of the variables; they are called the *principle components*.

The importance of PCA lies in the observation that  $(l_1 + \dots + l_k)/(l_1 + \dots + l_d)$  represents the “proportion of the total variation” explained by the first  $k$  principle components. So, if in our examination example  $l_1/(l_1 + \dots + l_d) = 0.75$  and its eigenvector is  $(1, 0, \dots, 0)$ , then we can conclude that 75% of the variation of the scores of the students is due to the first examination.

### 2.3 Cluster Analysis

Cluster Analysis (CA) is similar to pattern recognition discussed before. Again we try to classify based on similarity. Different from the previous two techniques, CA does not require the data to be real valued. To simplify our brief discussion we, however, make this assumption.

Again, let  $X$  be an  $\mathcal{R}^d$  valued random variable and let  $\mathcal{X} = X_1, \dots, X_n$  be a set of  $n$  observations of  $X$ . A clustering of  $\mathcal{X}$  is a cover of  $\mathcal{X}$  by disjoint subsets  $C_1, \dots, C_k$ . The goal is that the observations in the same class are similar while observations in different classes are different.

For example, if the  $X_i$  are observations on flowers, recording the length of the stem, the number of petals, et cetera, a clustering should put observations of flowers of the same kind in the same class.

Inherent in this statement of the cluster problem is the concept of an optimality criterion which dictates when a desirable partitioning has been found. This criterion can be phrased using a *quality function*. The higher the quality of a partitioning, the better it is.

More in particular, we need a measure of the *homogeneity* within a cluster and the *disparity* between clusters. Both measures can very well be based on a *distance function* or *metric* on  $\mathcal{R}^d$ .

For example, in *complete linkage* one of the restrictions on a class is that the distance between two observations may not exceed some threshold value  $r$ . In the *centroid method*, the distance between classes is defined as the distance between their centroids. One of the objectives of this method is to maximise the distance between classes.

There are way to many clustering algorithms to attempt even the shallowest of surveys here. An old, but very readable survey, can be found in [5]. This brief description ends with the observation that *clustering by complete enumeration* is completely out of the question.

Briefly, this technique would simply enumerate all possible clusterings, evaluate the quality of all of them and report the one(s) with the highest quality. This approach is infeasible simply by the sheer number of possible clusterings. The number of partitions of  $n$  objects in  $m$  non-empty subsets is given by *Stirling's numbers of the second kind*:

$$S(n, m) = \frac{1}{m!} \sum_{j=0}^m \binom{m}{j} (-1)^j (m-j)^n.$$

So, since the number of classes is in general not specified, the total number of clustering alternatives is given by:

$$\sum_{m=1}^n S(n, m).$$

### 3 PROJECTION PURSUIT

Mapping multivariate data into low dimensions for visual inspection is a commonly used technique in data analysis; if only because of the uncanny ability of humans to discover structure in two-dimensional plots. The discovery of such mappings that reveal the salient features of the multidimensional data set is in general far from trivial. *Projection Pursuit* (PP) introduced by Friedman and Tukey in [9] is a technique to discover such mappings.

In a nutshell, PP works as follows. We have a  $p$ -dimensional dataset  $X$  and we examine “all”, say, two-dimensional projections of  $X$ . We are given some quality function, called the projection index, with which we calculate the quality of all the projections. PP then reports the projection with the highest quality.

Stated as such, PP sounds like just another EDA technique which might as well have been discussed in the previous section. After a brief discussion of PP, however, it is shown, following Huber [15], that PP subsumes many EDA techniques.

#### 3.1 What is Projection Pursuit?

The simplest mappings from higher to lower dimensions are, linear, projections. That is, linear maps  $A$  of, say, rank 1 or 2. By definition, PP searches for a projection  $A$  that maximises a quality function, in this context it is called the *projection index*.

To get more concrete, let  $X$  be a  $\mathcal{R}^d$  valued random variable and let  $\mathcal{X} = \{X_1, \dots, X_n\}$  be a set of  $n$  observations of  $X$ . A 1-dimensional projection  $A$  is then a  $1 \times d$  matrix of rank 1. The quality of  $A$  should be determined from the data set  $A(\mathcal{X}) = \{AX_1, \dots, AX_n\}$ .

Many projection indices are possible, an important observation by Huber is that the index should measure how far the projection is away from a set of data points sampled under a normal distribution. The heuristic arguments underlying this claim are:

- A multivariate distribution is normal iff all its one-dimensional projections are normal. Thus, if the least normal one dimensional projection is normal, we need not look at any other projection.
- For most high-dimensional data sets most low-dimensional projections are approximately normal.



A simple projection index in this case is, thus, a  $\chi^2$ -test. Another example is the sample entropy, i.e.,

$$\frac{1}{n} \sum_{i=1}^n \log(\hat{f}(AX_i))$$

in which  $\hat{f}$  is the density estimate of the projected points. Friedman and Tukey's original index  $I$  is the product of two functions  $s$  and  $k$ , where  $s$  measures the spread of the data and  $k$  describes the "local density" of the data after projection.

Defining the index is only part of the work. The question is, of course, how to find the projection  $A$  that maximizes the index. Friedman and Tukey mention that their projection index is sufficiently continuous to allow the use of hill-climbing algorithms for the maximization.

A simple form of hill-climbing is as follows. First we choose a random projection matrix  $A = (a_1, \dots, a_d)$  and compute its quality. Subsequently, we construct a set  $\{A_1, \dots, A_N\}$  by adding small vectors to  $A$  in "all possible directions". Then we compute the quality of all these projections. The new projection  $A'$  is that projection from the set  $\{A, A_1, \dots, A_N\}$  that has maximal quality. If  $A = A'$  we stop, else we iterate.

This form of hill-climbing will always end in a local maximum. To find a global maximum the algorithm should be repeated with different initial projections. Moreover, in fact the search is not so much for the global maximum as well as for a projection that gives the analyst insight in the distribution of the data. In other words, we can stop as soon as we find a local maximum that satisfies this criterium.

Besides hill-climbing many more search algorithms exist, we return to this topic later in this paper.

### 3.2 *Projection Pursuit subsumes classical techniques*

It is straightforward that PP is a generalisation of PCA. For, in PCA we simply calculate the eigenvalues and eigenvectors of the covariance matrix and project the data orthogonally into the space spanned by the eigenvectors belonging to the largest eigenvalues. This projection clearly fits into our description of PP above.

#### 3.2.1 *Regression*

The subsumption of Regression by PP is less straightforward than that of PCA above. A central role is played by the "curse of dimensionality" caused by the fact that a high-dimensional space is mostly empty. To give an example let  $d = 20$ , which is actually low in most data mining examples. Assume that we have a large number of points uniformly distributed in a 20-dimensional unit ball. Then the radius of a ball containing 5% of the data is  $(0.05)^{(0.05)} = 0.86$ . So, if we want to pick out small features the sample size has to be gigantic. In



other words, for high-dimensional data sets standard Regression is not likely to produce good approximations.

In this case [15], it is often attractive to approximate the response surface by a sum of *ridge functions*:

$$f(x) \approx \sum_{i=1}^m g_i(a_i^T x)$$

In other words, we assume that  $f$  can be approximated by the sum of a set of  $\mathcal{R}$ -valued functions, each of which is defined on a 1-dimensional projection of  $\mathcal{X}$ . The idea is now to use PP to find the “optimal projections” for this approximation. More in particular, Friedman and Stuetzle’s Projection Pursuit Regression process [8] works as follows. Assume we have already determined the first  $m - 1$  vectors  $a_i$  and functions  $g_i$ . Let

$$r_i = Y_i - \sum_{i=1}^{m-1} g_i(a_i^T x)$$

be the residuals of this approximation. Choose a unit vector  $a \in \mathcal{R}^d$  and fit a smooth function  $g$  through the data set formed by the pairs  $(a^T X_i, r_i)$ . Calculate the sum of squared residuals relative to this  $g$ ,

$$\sum_{i=1}^n (r_i - g(a^T X_i))^2$$

and then minimise this sum over all possible choices for  $a$ . The resulting  $a$  and  $g$  are then inserted as the next term in the approximating sum. This iterative procedure stops if the improvement becomes small.

In a similar sense, PP can be said to subsume density estimation. That is, in cases of high-dimensionality Projection Pursuit Density Estimation yields an acceptable approximation.

### 3.2.2 Clustering

If stating that PP subsumes Regression was already stretching the limits, stating that it subsumes Clustering certainly oversteps these limits. However, PP can certainly help to detect clusters; one might say that this was the motivation for developing PP. In fact, Huber presents the following list of as possible actions after one has found some interesting projections:

1. Identify clusters, isolate them and investigate them separately.
2. Identify clusters and locate them (i.e., replace them by, say, their center and classify points according to membership to a cluster).
3. Find a parsimonious description (separate structure from random noise in a nonparametric fashion).

Data Mining not only generalises PP, it *does* generalise Clustering. How it achieves this, is discussed in the next section.

## 4 DATA MINING

For some researchers, Data Mining is simply the application of Machine Learning techniques to large databases. This point of view, however, is far too broad; if only because some techniques simply do not scale up to the massive amounts of data available in databases.

Klösgen and Zytkow define KDD, one of the many aliases of Data Mining, in [22] as

**Knowledge Discovery in Databases (KDD)** is a major direction in machine discovery dealing with knowledge discovery processes in databases. KDD applies to the ready data available in all application domains of science and in applied domains of marketing, planning, controlling, etc. Typically, KDD has to deal with inconclusive data, noisy data, and sparse data.

where *machine discovery* and *knowledge discovery process* are defined by respectively:

**Machine Discovery** is a subfield of Artificial Intelligence which develops discovery methods and discovery systems to support knowledge discovery processes.

**Knowledge Discovery Process** aims at finding out new knowledge about an application domain. Typically, a discovery process consists of many discovery steps, each attempting at the completion of a particular discovery task, and accomplished by the application of a discovery method. A discovery process emerges iteratively and depends on the dynamic, result dependent discovery goals. The process iterates many times through the same domain, typically based on search in various hypotheses spaces. New knowledge is inferred from data often with the use of old knowledge. Domain exploration and discovery focussing are discovery processes applied in new domains, where old knowledge is not available.

For the definition of the unfamiliar terms in these definitions, the reader is referred to [22]. In this paper we use a, slightly, formalised restricted version of this general definition. It is not meant as a general introduction to Data Mining. Again, this is far beyond the scope of this paper. The interested reader is referred to [14, 31, 32].

### 4.1 Descriptions and Quality

Central in Data Mining is the notion of a *description*. Recall that a database table consists of a *schema* and a *state*. A schema is a set of *attribute names*  $\mathcal{A} = \{A_1, \dots, A_p\}$  together with a set of *attribute domains*  $\{D_1, \dots, D_p\}$ , such that  $D_i$  is the *domain* of  $A_i$ . A state of the table can be seen as a finite subset of  $D_1 \times \dots \times D_p$ .

Usually, databases have more than one table and the tables are subject to constraints etcetera, but these nuances are unimportant for our present purposes. In other words, we will equate databases with tables as defined above, i.e., a database state  $db \subseteq_{fin} D_1 \times \cdots \times D_p$ . By  $DB$  we will denote the set of all possible database states.

A *tuple*  $t$  is simply an element of a database state, i.e.,  $t \in db$ . Rather than using a projection-notation like  $\pi_{D_i}(t)$ ,  $t.A_i$ , or even  $t_i$  if  $\mathcal{A}$  is understood, is used to denote the value of  $t$  for attribute  $A_i$ .

With these conventions, we can define a *description language*  $\Phi$  as a first order language such that  $\forall \phi \in \Phi \forall db \forall t \in db$  it can be decided whether  $\phi$  holds for  $t$  in  $db$ . Note that usually the attribute names in  $\mathcal{A}$  will be among the non-logical symbols of  $\Phi$ .

A popular description language is that of *set-descriptions*, these are descriptions of the form:

$$A_i \in V_i \wedge \cdots \wedge A_k \in V_k, \text{ where } A_j \in \mathcal{A} \wedge V_j \subseteq D_j \wedge V_j \text{ is finite.}$$

The description  $age \in [19, 24] \wedge gender = male$  is an example. Since the  $V_i$  are assumed to be finite, this is a first order language in disguise.

The cover of a description  $\phi$ , denoted by  $\langle \phi \rangle_{db}$ , in a database state  $db$  is the set of all tuples in  $db$  that satisfy  $\phi$ ; if  $db$  is clear from the context, this subscript is often omitted. For example,  $\langle age \in [19, 24] \wedge gender = male \rangle$  denotes all tuples in the database that describe young men.

Besides descriptions, a central role is played by *quality functions*, similar to those encountered in EDA. In fact, there are three classes of quality functions that are used in Data Mining:

**Class 1** these are quality functions that assign a quality to a single description for a given database state. That is, they are functions of type  $\Phi \times DB \rightarrow \mathcal{R}$ .

**Class 2** these are quality functions that assign a quality to a finite set of descriptions for a given database. That is, they are of type  $\mathcal{P}_{fin}(\Phi) \times DB \rightarrow \mathcal{R}$ .

**Class 3** these are quality functions that assign a quality to a set of descriptions for a given database state based on a combination of a Class 1 and a Class 2 quality function. In other words, a quality function of this class is specified by three functions:

1.  $Q_1 : \Phi \times DB \rightarrow \mathcal{R}$ ;
2.  $Q_2 : \mathcal{P}_{fin}(\Phi) \times DB \rightarrow \mathcal{R}$ ;
3.  $f : \mathcal{P}_{fin}(\mathcal{R}) \times \mathcal{R} \rightarrow \mathcal{R}$ ;

and  $Q_3 : \mathcal{P}_{fin}(\Phi) \times DB \rightarrow \mathcal{R}$  is defined by  $Q_3 = f(\{Q_1\}, Q_2)$ .

Given the set of descriptions  $\Phi$  and the quality function(s), Data Mining is simply: “find the (set of) description(s) with the highest quality”. Simple variations are of the form: “give me the  $n$  best descriptions” etcetera.

It is now easy to see that both Cluster Analysis and Projection Pursuit are examples of Data Mining.

#### 4.1.1 Cluster Analysis

Define the description language  $\Phi$  such that all finite subsets of  $\mathcal{P}(D_1 \times \cdots \times D_p)$  can be described. Moreover, define  $Q_1$  as a function that measures the homogeneity of the clusters, i.e., of the  $\langle \phi_i \rangle$ ,  $Q_2$  as a function that measures the disparity between the clusters, and define  $f$  as a function that combines  $Q_1$  and  $Q_2$ . The resulting Class 3 quality function and the description language  $\Phi$  together form a specification of the clustering problem as defined before.

#### 4.1.2 Projection Pursuit

This one is even more simple, define  $\Phi$  such that all and only all projection planes can be described. Moreover, define the Class 1 quality function as your favourite PP index. The result is PP as a Data Mining problem.

#### 4.1.3 Subsumption

It is disputable whether Data Mining with Class 3 quality functions is a more general problem than Cluster Analysis. However, although the two problems may be close in theory, they are widely different in practice. Most often in Cluster Analysis, the quality is somehow related to a distance function. In Data Mining, however, the quality function is simply part of the specification of the kind of information one is interested in.

The fact that Data Mining with Class 1 quality functions is more general than Projection Pursuit is far less disputable. There is at least one paper that studies PP on discrete data rather than continuous data, [4], but in Data Mining one does not fix an a priori “projection dimension”, rather one lets the system find the most striking projection.

The generality of Data Mining does have its price, however. In the first place, one has to specify each search task. That is, one has to choose an appropriate description language and a reasonable quality function. This specification comes at the price of a thorough analysis of the problem. In other words, Data Mining is not “plug and play”.

The second down-side lies in the search algorithms. The generality of the Data Mining problem implies that it is difficult to use the structure in a problem to speed up the search. In other words, the search algorithms should be able to cope with a large collection of widely different quality functions which, e.g., do not have to depend on a metric as in Cluster Analysis.

In the next subsection we give an example on the definition of quality functions. In the next section we return to the problem of search algorithms.

### 4.2 Risk Profiles: an example of quality functions

One of the Dutch insurance companies has asked us to derive *risk-profiles* from their car-insurance databases. A set of risk-profiles is a classification of the

insurants such that the insurance company can expect all clients in the same class to cause the same claim-amount per year. The relevance of this knowledge for the insurance business is obvious.

As a first approximation, we derive risk-profiles for the probability that someone will cause a claim, rather than for the expected claim-amount. In this section we briefly explain how these risk-profiles were found; more information can be found in [39].

#### 4.2.1 The Problem

The assumptions underlying this task are as follows. First, we assume that there only a few groups of clients, such that clients in the same group share the same probability of causing a claim. Secondly, we assume that these groups can be distinguished using only a few, say 80, properties of the clients and their cars; moreover, these properties are present in the database as attributes. Finally, we assume that these groups can be distinguished by our description language  $\Phi$ .

A precise definition of  $\Phi$  is not important here. It is a sublanguage of the language of set-descriptions that satisfies the following properties:

1.  $\Phi$  should be *sparse*, this more or less means that  $\langle \phi \rangle$  should be large and with attributes such as *area* and *age* there should be no gerrymandering;
2. If  $\langle \phi \rangle \cap \langle \psi \rangle$  is large for  $\phi, \psi \in \Phi$ , then  $\phi \wedge \psi \in \Phi$ .

To state our problem in terms of descriptions, define a set  $\{\phi_1, \dots, \phi_k\}$  of descriptions to be a *disjunctive cover*, abbreviated to *discovery*, if:

1.  $\forall i, j \in \{1, \dots, k\} : i \neq j \rightarrow [\phi_i \wedge \phi_j \rightarrow \perp]$
2.  $[\bigvee_{i=1}^k \phi_i] \rightarrow \top$

The problem can then be restated as: find a discovery  $\{\phi_1, \dots, \phi_k\}$  such that  $\forall v_1, v_2 \in D_1 \times \dots \times D_n : [p(v_1) = p(v_2)] \leftrightarrow [\forall i \in \{1, \dots, k\} : [\phi_i(v_1) \leftrightarrow \phi_i(v_2)]]$ .

#### 4.2.2 Analysis of the problem

A set of clients is called *homogeneous* if all members have the same probability of causing a claim. A description  $\phi$  is homogeneous, if the set of all clients that satisfy  $\phi$  is homogeneous. Clearly, the discovery we want to find should be homogeneous, i.e., all its descriptions should be homogeneous.

For a homogeneous description  $\phi$ , the probability of causing a claim of the clients that satisfy  $\phi$  can easily be estimated from the database. Since, all tuples in  $\langle \phi \rangle$  can be seen as records of trials of the same Bernoulli experiment. The outcome of this experiment is 1 (a success (sic)) if there was an accident and 0 otherwise.

So, using standard probability theory, [7], we can compute the, say 95%, confidence interval  $CI_\phi$  for the probability of causing a claim of the clients that satisfy  $\phi$ .

In fact, we will compute  $CI_\phi$  in this way for all descriptions  $\phi$ , regardless of whether they are homogeneous or not. Since our end-result is a homogeneous discovery this does not introduce errors.

The question is now, how do we decide whether a description is homogeneous or not. Intuitively,  $\phi$  is homogeneous, if all subsets of  $\langle\phi\rangle$  have the same associated probability. But this cannot not work, in a vase with with  $n$  blue and  $m$  red marbles one can find subsets with fractions of blue marbles varying from 0 to 1.

However, we are not interested in random subsets, but only in subsets that can be described by  $\Phi$  and  $\Phi$  is assumed to be sparse. Therefore, we define a description  $\phi \in \Phi$  to be homogeneous<sup>1</sup> if:

$$\forall \psi \in \Phi : \phi \wedge \psi \in \Phi \rightarrow CI_\phi \cap CI_{\phi \wedge \psi} \neq \emptyset$$

In other words, if we call  $\phi \wedge \psi$  an *extension* of  $\phi$ , a description is homogeneous if its associated probability cannot be distinguished, with 95% certainty, from those of its extensions

Not all homogeneous discoveries are answers to our question, because not all homogeneous discoveries satisfy the condition that the associated probabilities are distinct. Those homogeneous discoveries that do satisfy this condition are said to *split* the database. In other words, a homogeneous discovery  $\{\phi_1, \dots, \phi_l\}$  splits the database if:

$$\forall i, j \in \{1, \dots, l\} : i \neq j \rightarrow CI_{\phi_i} \cap CI_{\phi_j} = \emptyset$$

All such discoveries are potential answers to our question.

#### 4.2.3 Existence and Quality

If  $\Phi$  is carefully defined, many homogeneous discoveries will exist. For example, from a list  $\Psi = [\phi_1, \dots, \phi_n]$ ,  $\phi_i \in \Phi$  of descriptions we can generate the list  $\Psi' = \{\phi_1, \neg\phi_1 \wedge \phi_2, \neg\phi_1 \wedge \neg\phi_2 \wedge \phi_3, \dots, (\neg\phi_1 \wedge \dots \wedge \neg\phi_n)\}$ .  $\Psi'$  is potentially a homogeneous discovery if it is,  $\Psi$  is called a *decision list*, [35].

Whether there exist homogeneous discoveries that split the database depends more on the actual database state than on the design of  $\Phi$ . In other words, there might be 0, 1 or many.

If there are 0, we are out of luck. The database simply does not contain enough information to partition the clients through risk-profiles. If there are many, we seem to be in similar straits because we can assign many different risks to the same client. However, the *quality* of the different discoveries may differ considerably. In other words, one might be naturally the best.

<sup>1</sup>A related notion of homogeneity has been introduced independently in [37]

A detailed discussion of quality measures on discoveries is outside the scope of this paper. One aspect, however, is interesting to note. One reason for having many homogeneous discoveries is that many descriptions are homogeneous by definition, i.e., all those descriptions which have no extensions in  $\Phi$ .

These trivially homogeneous descriptions are in a sense too small to count. In other words, a homogeneous description with a large cover is better than one with a small cover. Extending this to discoveries, a discovery that partitions the database in large subsets is better than one that partitions it into smaller subsets.

Similarly, the better a set of descriptions distinguishes between its components, the better it is. To formalise this, define that a homogeneous set of descriptions  $\{\phi_1, \dots, \phi_l\}$  *strongly splits* the database if its descriptions differ in all aspects:

$$\forall \psi \in \Phi \forall i, j \in \{1, \dots, l\} : \left[ \begin{array}{cc} i \neq j & \wedge \\ \phi_i \wedge \psi \in \Phi & \wedge \\ \phi_j \wedge \psi \in \Phi & \end{array} \right] \rightarrow CI_{\phi_i \wedge \psi} \cap CI_{\phi_j \wedge \psi} = \emptyset$$

Let  $\{\phi_1, \dots, \phi_k\}$  and  $\{\psi_1, \dots, \psi_l\}$  be two homogeneous discoveries that strongly split the database and such that all  $\langle \phi_i \rangle$  and  $\langle \psi_j \rangle$  are large. Then there is for each  $\phi_i$  at least one  $\psi_j$  such that  $\langle \phi_i \rangle \cap \langle \psi_j \rangle \gg \emptyset$  and thus  $\langle \phi_i \wedge \psi_j \rangle \in \Phi$ . But since both discoveries strongly split the database, there can be at most one. So,  $\langle \phi_i \rangle \approx \langle \psi_j \rangle$  and  $CI_{\phi_i} \approx CI_{\psi_j}$ . In other words, in this case there is essentially only one way to partition the database in a good way.

The fact that the discoveries are not unique is simply caused by the fact that a set of tuples can have more than one description. For example, it could happen that almost all young clients are male and vice versa. In that case the descriptions *age = young* and *gender = male* are equally good from a theoretical point of view. Not necessarily from a practical point of view. For, it is very well possible that the description *age = young* makes sense to a domain expert while *gender = male* does not. Hence, both options should be presented to the domain expert.

#### 4.2.4 The Search

If a homogeneous discovery exists that splits the database, it must contain a homogeneous description with the highest associated probability. This suggests a simple algorithm to find such a discovery:

Make a list of homogeneous descriptions as follows:

find a  $\phi$  that has the maximal associated probability.

remove  $\langle \phi \rangle$  from *db* and add  $\phi$  to the list.

continue with this process until  $\top$  is homogeneous on the remainder of *db*;

Check whether the decision list splits the database.

In other words, we can use the associated probability of a rule as a measure of its quality.

## 5 WHAT COMPUTER SCIENCE OFFERS DATA MINING

If Data Mining can be considered as a generalisation of more or less standard statistical techniques, what has Computer Science to offer? In other words, how can Computer Science help to solve the Data Mining problems? In this section we discuss how the two Computer Science parents, AI and database technology help to solve Data Mining tasks with a reasonable performance.

### 5.1 AI: Search Techniques

Much effort in Machine Learning and in AI in general has been invested in efficient and/or robust search techniques. The range of these often problem specific techniques is far too large to discuss in this paper. Rather, we will concentrate on one technique, viz., *genetic search* [11, 26, 18]. To simplify our discussion, we start with the assumption that we have a Class 1 quality function.

Genetic search, like all genetic algorithms, is defined in analogy with biological evolution, i.e., it is based on the survival of the fittest. It maintains a population of proposed solutions (*chromosomes*) for a given problem. Iteratively, the population undergoes a *simulated evolution*: relative “good” solutions produce offspring, which subsequently replace the “worse” ones.

Each iteration, called a *reproduction cycle*, is performed in three steps. During the selection step a new population is formed from stochastically best samples (with replacement). Then, during the *recombination* step some of the members of the newly selected population are altered. Finally, all such altered individuals are evaluated.

The recombination is based on two operators: *mutation* and *crossover*. Mutation introduces random variability into the population, and crossover exchanges random pieces of both chromosomes in the hope of propagating partial solutions. Schematically, we have the following algorithm:

```
t := 0
initialise P(t)
evaluate P(t)
while (not termination-condition) do
  t := t+1
  select P(t) from P(t-1)
  recombine P(t)
  evaluate P(t)
od
```

Hence, for the specification of a genetic algorithm for a particular problem we must have the following five components:

1. a “genetic” representation for potential solutions to the problem,
2. a way to create an initial population of potential solutions,



3. an evaluation function that plays the role of the environment, rating solutions in terms of their “fitness”,
4. genetic operators that alter the composition of children during reproduction,
5. values for various parameters that the genetic algorithm uses (population size, probabilities of applying genetic operators, etc.).

For our search problem, the items 1, 3, and 4 can be defined as follows. The chromosomes are simply the descriptions in our description language  $\Phi$ , say slightly modified set-descriptions. More in particular set-descriptions of the form:

$$A_1 \in V_1 \wedge \cdots \wedge A_p \in V_p, \text{ where } A_j \in \mathcal{A} \wedge V_j \subseteq D_j \wedge V_j \text{ is finite or } V_j = D_j.$$

In other words all set-descriptions cover all attributes, the cases where  $V_j = D_j$  simply cover the attributes on which one doesn't select.

The evaluation function is simply our quality function. The genetic operators can be defined as follows:

**Crossover** For two descriptions  $A_1 \in V_1 \wedge \cdots \wedge A_p \in V_p$  and  $A_1 \in W_1 \wedge \cdots \wedge A_p \in W_p$ , choose two elements  $i, j \in \{1, \dots, p\}$  and conclude the descriptions:

$$\begin{aligned} A_1 \in V_1 \wedge \cdots \wedge A_{i-1} \in V_{i-1} \wedge \\ A_i \in W_i \wedge \cdots \wedge A_j \in W_j \wedge A_{j+1} \in V_{j+1} \wedge \cdots \wedge A_p \in V_p \\ A_1 \in W_1 \wedge \cdots \wedge A_{i-1} \in W_{i-1} \wedge \\ A_i \in V_i \wedge \cdots \wedge A_j \in V_j \wedge A_{j+1} \in W_{j+1} \wedge \cdots \wedge A_p \in W_p \end{aligned}$$

**Mutation** For a description  $\phi$ , choose an  $i \in \{1, \dots, p\}$  and a random  $W_i \subseteq D_i$ , and replace  $A_i \in V_i$  in  $\phi$  by  $A_i \in W_i$ .

Alternatively, one might execute one step of the Hill-climber algorithm as a mutation step.

The good parameters for the algorithm can hardly be defined in advance, they have to be found by experimentation. It is well-known, however, that the population size should be relatively large, say a few hundred, and that quite some iteration steps, again say a few hundred, are needed before such a system will converge.

If we consider Class 2 or Class 3 quality functions we have to deal with *genomes*, i.e., sets of descriptions rather than with descriptions. In principle, this only changes the possible genetic operators. For example, genomes can switch complete chromosomes or they can pair their chromosomes and combine these pairs as above. In mutation, one might also consider simply dropping chromosomes or change one of the chromosomes with a completely new, arbitrary, chromosome. This freedom of choice implies that we simply should take

a larger collection of operations with a varying probability of being actually chosen. For more information and other possible choices, see [14].

As explained at the beginning of this subsection, there are many more search algorithms than genetic search. One of the distinct advantages of genetic search, however, is its inherent parallelism. All recombinations and all quality evaluations can be done in parallel. This promises a considerable speed-up of the process.

### *5.2 Databases: Handling massive volumes of data*

Large in Statistics is a different term from large in Databases. In statistics a large sample consists of a few thousand records. Large Databases have hundreds of thousands if not millions of records. Moreover, the number of possibly relevant attributes in Data Mining count easily up to 50 or 60. Using samples to cope with these large volumes of data means invariably a loss of resolution in our search. It is far easier for a group of 20,000 to stand out significantly in a crowd of a million than it is for a group of 20 to stand out significantly in a crowd of a thousand.

However, standard database technology is not the answer to the problem of massive amounts of data either. For, the discovery process queries the database severely, since:

1. dbms's are tuned to a variety of uses including transactions,
2. discovery is in principle a read-only process on the database; having access, during the search, to the newest data does not improve the quality of the information significantly,
3. during the search, old results can often be reused,

it is profitable to have a knowledge discovery tool with its own data-server, geared specially towards discovery.

Such a data server is a dbms-kernel tailored for data mining purposes. That is, it contains no transaction management functionality nor write protection. In fact, one can only store new, derived, data, one cannot update data.

What it does contain however, are various mechanisms to speed up query processing as much as possible. For data mining causes an avalanche of queries posed to the database as is witnessed by the description of genetic search.

First and foremost, the data server is a parallel system, since it has been proven that parallel systems can answer bursts of queries far more efficiently than mono-processor systems.

Secondly, it contains a query-optimisation module that optimizes queries both statically and dynamically. Static optimisation is rewriting a query into the most efficient form given database characteristics. Dynamic optimisation means that the query is processed as efficiently as possible given all the other queries that are processed concurrently, [41].

Finally, it contains a browsing optimization module. Many queries in a data mining search are related. That is, later queries can be executed much more efficiently if some previous results are stored temporarily than when they are executed against the complete database. The browsing optimization module tries to optimize query processing by storing such intermediate results [20].

Another aspect of efficient query processing is using the most suited data structure. Therefore, the data-server can dynamically adapt its data-layout (in main memory) to suit the current search process as much as possible [19].

All these techniques are either well-studied in database research or are currently under vigorous investigation

## 6 FOUNDATIONS OF DATA MINING: THEORY AND EXPERIMENT

Data Mining becomes a mature tool for the exploratory data analyst only if one can trust the results. In other words, Data Mining should be given sound mathematical foundations. These foundations comprise two aspects, viz., the quality functions and convergence of the search process.

The primary goal of Data Mining is the discovery of strategic information. In other words, the results will be used to predict the, near, future. The quality functions should be chosen in such a way that such extrapolations are, at least statistically, valid.

Given a description language, a quality function, and a database state, we get a so called *fitness landscape*; a multi dimensional graph of the quality function over the descriptions. The task is to list the descriptions of high quality.

Most often, the size of the set of descriptions makes an exhaustive search over the fitness landscape intractable, as discussed above. Heuristic searches are the only viable option. The immediate question is then, of course, how well do the results found by heuristic search compare with the, almost hypothetical, results of exhaustive search.

The only way in which a heuristic search can consistently outperform random search is by exploiting the shape of the fitness landscape. The shape of this landscape is governed by the, perhaps implicit, structure in the set of descriptions and the behaviour of the quality function on this structure.

In other words, to design good search algorithms one should study the structure of the set of descriptions, e.g., does it form a lattice, is it a topological space or even a metrical space? Moreover, one should study the behaviour of quality functions on this structure, e.g., are they continuous or monotonic.

### 6.1 What Statistics might offer

Since we have argued that Statistics should be considered as one of the parents of Data Mining, it is only natural to ask what Statistics might offer towards the resolution of these two foundational problems.

For the first problem, the quality function, this is rather obvious. If only because of the quality functions defined for Projection Pursuit. More in general,

if we want statistically valid results, we should use Statistics to test the validity of our results.

For the second problem, the convergence problem, the situation is less clear. There are some results on convergence properties for genetic algorithms in a framework of stochastic processes [33, 34]. However, these results apply to the case of an infinite population size in continuous space. To make this results usefull as foundations for Data Mining, they should be extended to finite population sizes in mixed continuous and discrete spaces.

For a different type of search algorithms, viz., Neural Networks, there are more results. In [10], for example, techniques from Statistical Mechanics are used to analyse the behaviour of Neural Networks. Moreover, in [17], the authors derive the *Information Geometry* of *Boltzmann Machines*, a special class of Neural Networks, along the lines of [16]. These results still depend on continuous space, but no longer do they depend on an infinite population size. The problem with these results, however, is that it far from clear how Neural Nets can be used in a search for descriptions.

## 6.2 *Data Surveyor: experimental guidance*

The development of the mathematical underpinnings of Data Mining cannot be the result of theoretical studies alone. Consider, e.g., the quality functions. Although different functions may be more or less mathematically equivalent, their usability in practice might differ considerably. Similar remarks are valid for the convergence problem. One can make synthetic databases in which known results are hidden. By testing the search strategies on such examples, some insight in the convergence process may be gained.

To get the experimental guidance for the theoretical development, a data mine tool called *Data Surveyor* is currently under development at CWI, [13]. Currently, it has a two-level architecture consisting of a data server on top of which the Surveyor kernel is executed. In the near future, it will be extended to a three level architecture.

The bottom layer will still consist of the data server. The middle layer will consist of a set of different search modules. The top layer will be the user-interface with which the user can formulate data mining tasks and guide the search task. The rationale for have several different search modules is twofold. First, it is very well possible that different algorithms perform better for different data mining problems. Second, by using different search algorithms on the same real world database states more insight in the convergence properties of the various algorithms can be gained.

**Acknowledgements:** The author wishes to thank Marcel Holsheimer, and Johan van den Akker for stimulating discussions and constructive criticism on an earlier version of this paper.

## REFERENCES

1. Martin Anthony and Norman Biggs. *Computational learning theory: an*

- introduction, volume 30 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1992.
2. A. Borgida, T.M. Mitchell, and K. Williamsson. Learning improved integrity constraints and schemas from exceptions in databases and knowledge bases. In J. Mylopoulos M.L. Brodie, editor, *On Knowledge Base Management Systems: Integrating Artificial Intelligence and Database technologies*. Springer-Verlag, 1986.
  3. James P. Delgrande. Formal limits on the automatic generation and maintenance of integrity constraints. In *Proc. 6th ACM Sigact-Sigmod-Sigart Symposium on Principles of Database Systems*, 1987.
  4. P. Diaconis. Projection pursuit for discrete data. Technical Report 198, Stanford University, 1983.
  5. Benjamin S. Duran and Patrick L. Odell. *Cluster Analysis, A Survey*. Lecture Notes in Economics and Mathematical Systems, vol 100. Springer-Verlag, 1974.
  6. Usama M. Fayyad and Ramasamy Uthurusamy, editors. *AAAI-94 Workshop Knowledge Discovery in Databases*, Seattle, Washington, 1994.
  7. William Feller. *An introduction to probability theory and its applications, Vol 1*. Wiley, 1950.
  8. J.H. Friedman and W. Stuetzle. Projection pursuit regression. *Journal of the American Statistical Association*, 76:817–823, 1981.
  9. J.H. Friedman and J.W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computing*, C-23:881–889, 1974.
  10. John Hertz, Anders Krogh, and Richard G. Palmer. *Introduction to the Theory of Neural Networks*. Santa Fe Institute Lecture Notes vol 1. Addison-Wesley, 1991.
  11. John H. Holland. *Adaptation in natural artificial systems*. University of Michigan Press, Ann Arbor, 1975.
  12. John H. Holland, Keith J. Holyoak, Richard E. Nisbett, and Paul R. Thagard. *Induction: processes of inference, learning and discovery*. Computational models of cognition and perception. MIT Press, Cambridge, 1986.
  13. Marcel Holsheimer, Martin Kersten, and Arno Siebes. Data surveyor: Searching the nuggets in parallel. In Piatetsky-Shapiro and Frawley [32], chapter 4.
  14. Marcel Holsheimer and Arno P.J.M. Siebes. Data mining: the search for knowledge in databases. Technical Report CS-R9406, CWI, January 1994.
  15. Peter J. Huber. Projection pursuit. *The Annals of Statistics*, 13(2):435–475, 1985.
  16. Shun ichi Amari. *Differential-Geometrical Methods in Statistics*. Lecture Notes in Statistics, vol. 28. Springer-Verlag, 1985.
  17. Shun ichi Amari, Koji Kurata, and Hiroshi Nagaoka. Information geometry of boltzmann machines. *IEEE transactions on Neural Networks*, 3(2):260–271, 1992.
  18. Cezary Z. Janikow. *Inductive Learning of Decision Rules from Attribute-*

- Based Examples: A Knowledge-Intensive Genetic Algorithm Approach*. PhD thesis, University of North Carolina at Chapel Hill, 1991.
19. Martin L. Kersten. Goblin: A DBPL designed for Advanced Database Applications. In *2nd Int. Conf. on Database and Expert Systems Applications, DEXA '91*, Berlin, Germany, August 1991.
  20. Martin L. Kersten and Michiel de Boer. Query optimization strategies for browsing sessions. In *Proc. IEEE Int. Conf. on Data Engineering*, Houston, 1994.
  21. Jyrki Kivinen and Heikki Mannila. Approximate dependency inference from relations. In *Proc. 4th Int. Conf. on Database Theory*, 1992.
  22. Willi Klösgen and Jan Zytkow. Machine discovery terminology. In Fayyad and Uthurusamy [6], pages 463–473.
  23. Yves Kodratoff and Ryszard S. Michalski, editors. *Machine Learning, an Artificial Intelligence approach*, volume 3. Morgan Kaufmann, San Mateo, California, 1990.
  24. Heikki Mannila and Kari-Jouko Rähkä. Algorithms for inferring functional dependencies from relations. *Data and Knowledge Engineering*, 12:83–99, 1994.
  25. K.V. Mardia, J.T. Kent, and J.M. Bibby. *Multivariate Analysis*. Probability and Mathematical Statistics. Academic Press, 1979.
  26. Zbigniew Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Artificial Intelligence. Springer-Verlag, 1992.
  27. Ryszard S. Michalski, Jaime G. Carbonell, and Tom M. Mitchell, editors. *Machine Learning, an Artificial Intelligence approach*, volume 1. Morgan Kaufmann, San Mateo, California, 1983.
  28. Ryszard S. Michalski, Jaime G. Carbonell, and Tom M. Mitchell, editors. *Machine Learning, an Artificial Intelligence approach*, volume 2. Morgan Kaufmann, San Mateo, California, 1986.
  29. M. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, 1969.
  30. J.C. Mitchell. Inference rules for functional and inclusion dependencies. In *Proc. 2nd ACM Sigact-Sigmod Symposium on Principles of Database Systems*, 1983.
  31. Gregory Piatetsky-Shapiro and William J. Frawley, editors. *Knowledge Discovery in Databases*. AAAI Press, Menlo Park, California, 1991.
  32. Gregory Piatetsky-Shapiro and William J. Frawley, editors. *Knowledge Discovery in Databases*, volume II. MIT Press, Menlo Park, California, forthcoming.
  33. Xiaofeng Qi and Francesco Palmieri. Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space part i: Basic properties of selection and mutation. *IEEE transactions on Neural Networks*, 5(1):102–119, 1994.
  34. Xiaofeng Qi and Francesco Palmieri. Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space part ii: Analysis of the diversification role of crossover. *IEEE transactions on Neural*

- Networks*, 5(1):120–129, 1994.
35. Ronald L. Rivest. Learning decision lists. *Machine Learning*, 2:229 – 246, 1987.
  36. F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 96(6):386–408, 1958.
  37. Richard Segal and Oren Etzioni. Learning decision lists using homogeneous rules. In *Proceedings of the 12th National Conference on Artificial Intelligence*, pages 619–625. AAAI/MIT Press, 1994.
  38. Jude W. Shavlik and Thomas G. Dietterich, editors. *Readings in Machine Learning*. Morgan Kaufmann, 1990.
  39. Arno Siebes. Homogeneous discoveries contain no surprises: Inferring risk-profiles from large databases. In Fayyad and Uthurusamy [6], pages 97 – 108.
  40. J.W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.
  41. Carel Arie van den Berg. *Dynamic Query Processing in a Parallel Object-Oriented Database System*. PhD thesis, Twente University, 1994.





# Bernoulli Polynomials Old and New: Problems in Complex Analysis and Asymptotics

N. M. Temme

*CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands  
e-mail: nicot@cwi.nl*

This paper is dedicated to Cor Baayen with thanks for his support and recognition of my work, and for his considerable contributions to the advancement of the position and scientific status of SMC and CWI.

We consider two problems on generalized Bernoulli polynomials  $B_n^\mu(z)$ . One is connected with defining functions instead of polynomials by making the degree  $n$  of the polynomial a complex variable. In the second problem we are concerned with the asymptotic behaviour of  $B_n^\mu(z)$  when the degree  $n$  tends to infinity.

## 1. INTRODUCTION

At present Bernoulli numbers are introduced through generating functions, as we shall do below, but historically they arose in connection with the sums of the  $p$ -th power of the first  $n - 1$  integers  $1 + 2^p + \dots + (n - 1)^p$ . The Greeks, Hindus and Arabs all had rules amounting to

$$\begin{aligned}\sum_{i=1}^{n-1} i &= \frac{1}{2}n(n-1) = \frac{1}{2}n^2 - \frac{1}{2}n, \\ \sum_{i=1}^{n-1} i^2 &= \frac{1}{3}n^3 - \frac{1}{2}n^2 + \frac{1}{6}n, \\ \sum_{i=1}^{n-1} i^3 &= \frac{1}{4}n^4 - \frac{1}{2}n^3 + \frac{1}{4}n^2, \\ \sum_{i=1}^{n-1} i^4 &= \frac{1}{5}n^5 - \frac{1}{2}n^4 + \frac{1}{3}n^3 - \frac{1}{30}n.\end{aligned}$$

Nowadays we write for  $p = 0, 1, 2, \dots$ ,  $n = 1, 2, 3, \dots$  (putting  $0^0 = 1$ )

$$\sum_{i=0}^{n-1} i^p = \frac{1}{p+1} \sum_{k=0}^p \binom{p+1}{k} B_k n^{p+1-k},$$

where the coefficient of the linear term  $n$  equals the  $p$ -th Bernoulli number.

In this way the numbers were mentioned (without using their present names and notation) by Jakob I. Bernoulli in his posthumous *Ars conjectandi* of 1713. In fact he gave the above general formula, observing that the numbers also occur in the coefficients of the other powers of  $n$ . See the Latin text and table of Bernoulli's first ten *summae potestatum* from his Collected Works. Bernoulli actually made a mistake in the coefficient of  $n^2$  in the ninth row, which he gave as  $-\frac{1}{2}$ , but which should read  $-\frac{3}{20}$ . Euler also tackled the problem of summing powers and in 1755 he published a proof of the Bernoulli forms based on the calculus of finite differences, christening the coefficients of  $n$  the *Bernoulli numbers* in honour of Jakob.

Next we give some general definitions through generating functions. The *generalized Bernoulli polynomials*  $B_n^\mu(z)$  are defined for all complex numbers  $z$  and  $\mu$  by the expansion

$$\sum_{n=0}^{\infty} B_n^\mu(z) \frac{t^n}{n!} = e^{zt} \left( \frac{t}{e^t - 1} \right)^\mu, \quad |t| < 2\pi. \quad (1.1)$$

An immediate consequence of this definition is the representation of the generalized Bernoulli polynomials as a Cauchy type integral:

$$B_n^\mu(z) = \frac{n!}{2\pi i} \int_C e^{zt} \left( \frac{t}{e^t - 1} \right)^\mu \frac{dt}{t^{n+1}}, \quad (1.2)$$

where the contour  $C$  is a circle with radius less than  $2\pi$  around the origin.

There are several reductions for this general definition.

- When  $\mu = 1$  we have the *Bernoulli polynomials*  $B_n(z)$ .
- When  $z = 0$  we have the *generalized Bernoulli numbers*  $B_n^\mu$ .
- When  $\mu = 1$  and  $z = 0$ , we have the classical *Bernoulli numbers*  $B_n$ .

The quantities  $B_n^\mu(z)$  are polynomials of degree  $n$  in both  $\mu$  and  $z$ ;  $\mu$  is called the order. The classical numbers  $B_n$  occur in practically every field of mathematics, in particular in combinatorial theory, finite difference calculus, numerical analysis, analytical number theory, and probability theory. For the polynomials the same remarks apply, although in several occurrences the polynomials give just a convenient method of notation instead of giving insight or possibilities to further manipulate analytical expressions.

In this paper we consider two problems on the generalized Bernoulli polynomials  $B_n^\mu(z)$ . One is connected with defining functions  $B_\nu(z)$  where  $\nu$  is a complex variable. We derive a functional equation that generalizes the well-known property  $B_n(1-z) = (-1)^n B_n(z)$ , and that gives information how to interpret  $B_\nu(x)$  when  $x < 0$ . In the second problem we are concerned with the asymptotic behaviour of  $B_n^\mu(z)$  when the degree  $n$  tends to infinity. We consider this problem in connection with our earlier results for Stirling numbers and discuss some other results from the literature. Finally we give new asymptotic representations.

Atque si porrò ad altiores gradatim potestates pergere, levique negotio sequentem adornare laterculum licet:

*Summæ Potestatum.*

$$\begin{aligned} \int n &= \frac{1}{2} n n + \frac{1}{2} n . \\ \int n n &= \frac{1}{3} n^3 + \frac{1}{2} n n + \frac{1}{6} n . \\ \int n^3 &= \frac{1}{4} n^4 + \frac{1}{2} n^3 + \frac{1}{4} n n . \\ \int n^4 &= \frac{1}{5} n^5 + \frac{1}{2} n^4 + \frac{1}{3} n^3 * - \frac{1}{30} n . \\ \int n^5 &= \frac{1}{6} n^6 + \frac{1}{2} n^5 + \frac{5}{12} n^4 * - \frac{1}{12} n n . \\ \int n^6 &= \frac{1}{7} n^7 + \frac{1}{2} n^6 + \frac{1}{2} n^5 * - \frac{1}{6} n^3 * + \frac{1}{42} n . \\ \int n^7 &= \frac{1}{8} n^8 + \frac{1}{2} n^7 + \frac{7}{12} n^6 * - \frac{7}{24} n^4 * + \frac{1}{12} n n . \\ \int n^8 &= \frac{1}{9} n^9 + \frac{1}{2} n^8 + \frac{2}{3} n^7 * - \frac{7}{15} n^5 * + \frac{2}{9} n^3 * - \frac{1}{30} n . \\ \int n^9 &= \frac{1}{10} n^{10} + \frac{1}{2} n^9 + \frac{3}{4} n^8 * - \frac{7}{10} n^6 * + \frac{1}{2} n^4 * - \frac{1}{12} n n . \\ \int n^{10} &= \frac{1}{11} n^{11} + \frac{1}{2} n^{10} + \frac{5}{6} n^9 * - 1 n^7 * + 1 n^5 * - \frac{1}{2} n^3 * + \frac{5}{66} n . \end{aligned}$$

Quin imò qui legem progressionis inibi attentius inspexerit, eundem etiam continuare poterit absque his ratiociniorum ambagibus: Sumtâ enim  $c$  pro potestatis cujuslibet exponente, fit summa omnium  $n^c$  seu

$$\begin{aligned} \int n^c &= \frac{1}{c+1} n^{c+1} + \frac{1}{2} n^c + \frac{c}{2} A n^{c-1} + \frac{c \cdot c - 1 \cdot c - 2}{2 \cdot 3 \cdot 4} B n^{c-3} \\ &+ \frac{c \cdot c - 1 \cdot c - 2 \cdot c - 3 \cdot c - 4}{2 \cdot 3 \cdot 4 \cdot 5 \cdot 6} C n^{c-5} \\ &+ \frac{c \cdot c - 1 \cdot c - 2 \cdot c - 3 \cdot c - 4 \cdot c - 5 \cdot c - 6}{2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8} D n^{c-7} \dots \text{ \& ita deinceps,} \end{aligned}$$

exponentem potestatis ipsius  $n$  continuè minuendo binario, quousque perveniatur ad  $n$  vel  $n n$ . Literæ capitales  $A, B, C, D$  &c. ordine denotant coëfficientes ultimarum terminorum pro  $\int n n, \int n^4, \int n^6, \int n^8$  &c.  
nempe

$$A = \frac{1}{6}, B = -\frac{1}{30}, C = \frac{1}{42}, D = -\frac{1}{30}.$$

## 2. BERNOULLI FUNCTIONS

We consider the problem of generalizing  $B_n(z)$  by making  $n$  a complex variable. A motivation for this is given by the wish to generalize a fundamental difference relation of the Bernoulli numbers:

$$B_n(z+1) = B_n(z) + nz^{n-1}, \quad n = 0, 1, 2, \dots, \quad (2.1)$$

to a relation that also holds when  $n$  is replaced by a complex parameter  $\nu$ . A further step then is to interpret such a generalization for negative values of  $z$ . When we now how to interpret another fundamental property:

$$B_n(1-z) = (-1)^n B_n(z) \quad (2.2)$$

when  $n$  is complex and  $z$  is negative the problem can completely be solved.

A second motivation comes from the recent set of papers [4]–[7] by BUTZER *et al.* in which Bernoulli numbers and polynomials (and related quantities) are generalized. It seems that Butzer *et al.* have overlooked several rather old papers (for instance JONQUIÈRE (1891) and BÖHMER (1910)), in which generalizations of Bernoulli polynomials are considered. Part of our analysis is based on these two classical papers.

The difference relation (2.1) is the heart of *difference calculus*, the branch of mathematics that became so important in solving problems from numerical analysis, in particular in solving differential equations. Further information on classical difference calculus can be found in JORDAN (1947), NÖRLUND (1924), and MILNE-THOMSON (1933).

One of the striking occurrences of Bernoulli numbers in special functions is the relation

$$\zeta(2m) = \frac{1}{2}(-1)^{m+1}(2\pi)^{2m} \frac{B_{2m}}{(2m!)}, \quad m = 1, 2, 3, \dots, \quad (2.3)$$

where  $\zeta(s) = \sum_{k=1}^{\infty} k^{-s}$ , the Riemann zeta function. This relation was given by Euler (1735/1739), and Ramanujan used it to define *signless* Bernoulli numbers of arbitrary index  $s$  by writing

$$B_s^* = 2\Gamma(s+1)\zeta(s)(2\pi)^{-s} \quad (2.4)$$

(see BERNDT (1985, pp. 125f, 151f)), ‘signless’ meaning that

$$B_{2m}^* = (-1)^{m+1} B_{2m} > 0, \quad m = 1, 2, 3, \dots$$

In fact, as Berndt (*loc. cit.* p. 125) remarks, already Euler made a very first attempt to introduce (signless) Bernoulli numbers of arbitrary index as above. Apparently, he made no significant use of his idea. The relation (2.3) gave hope to many mathematicians that it would be possible to find values of  $\zeta(2m+1)$ , and the numbers  $B_s^*$  defined in (2.4) might be a convenient starting

point for this when relevant new properties of  $B_s^*$  could be found. Until now this approach to identify  $\zeta(2m+1)$  in terms of simple quantities has not been successful.

In this section we consider a different way of generalizing, by taking another explicit representation. When we generalize this by taking  $n$  complex we write (1.2) in the form

$$B_\nu^\mu(z) = \frac{\Gamma(\nu+1)}{2\pi i} \int_C e^{zt} \left( \frac{t}{e^t - 1} \right)^\mu \frac{dt}{t^{\nu+1}}, \quad (2.5)$$

where  $\Re z > 0$ . Because of the algebraic singularity of  $t^{\nu+1}$  at the origin we assume now that the contour of integration  $C$  runs from  $-\infty, \arg t = -\pi$ , encircles the origin in positive direction (that is, anti-clockwise) terminates at  $-\infty$ , now with  $\arg t = +\pi$ . We assume that all zeros of  $e^t - 1$  (except  $t = 0$ ) are not enclosed by the contour, and, initially, that the many-valued function  $t^\nu$  is real for real values of  $\nu$  and  $t > 0$ .

### 2.1. Bernoulli functions $B_\nu(z)$ in the complex plane.

In this subsection we first consider the analytic continuation of  $B_\nu(z)$  up to the negative  $z$ -axis. Originally the branch cut of the many-valued function  $t^\nu$  in (2.5) runs from 0 to  $-\infty$ . However, this choice is by convention. When  $\arg z \geq 0$ , we may turn the loop  $C$  in clockwise direction into the upper half plane. In this way we redefine the location of the branch cut in the  $t$ -plane. Turning around a positive angle  $\delta$ , we have at one side of the cut  $\arg t = \pi + \delta$ , and on the other side  $\arg t = -\pi + \delta$ . When we take  $\delta \in [0, \frac{1}{2}\pi)$ , the integral remains convergent when we allow  $\arg z$  ranging in the interval  $[0, \pi)$ . A similar method can be used for  $z$  in the lower half plane. This gives the analytic continuation of  $B_\nu(z)$  defined in (2.5) to the sector  $|\arg z| < \pi$ , for any complex value of  $\nu$ .

By using (2.5) it follows easily that the basic *difference property* (2.1) of the Bernoulli polynomials remains valid for the Bernoulli functions:

$$B_\nu(z+1) = B_\nu(z) + \nu z^{\nu-1}, \quad \nu \in \mathbb{C}, \quad |\arg z| < \pi. \quad (2.6)$$

Also the *derivative property*

$$\frac{d}{dz} B_\nu(z) = \nu B_{\nu-1}(z), \quad \nu \in \mathbb{C}, \quad |\arg z| < \pi \quad (2.7)$$

is easily verified by using (2.1). Observe that the analytic continuation of  $B_\nu(z)$  from the half plane  $\Re z > 0$  into the left half of the complex plane also follows from (2.6). Also in this way we cannot reach the negative  $z$ -axis.

Next we want to verify how relation (2.2) transforms when  $n$  becomes a complex parameter. This will give a quite non-trivial property. To obtain information on  $B_\nu(1-z)$  we replace  $z$  with  $-z$  in (2.6). To take into account the many-valuedness of the function  $z^\nu$  and the condition  $|\arg z| < \pi$ , we change

in (2.6)  $z$  into  $ze^{-i\pi}$  when  $z$  is in the upper half plane  $\Im z > 0$  and change  $z$  into  $ze^{+i\pi}$  when  $z$  is in the lower half plane. The result is when  $\Im z > 0$ :

$$B_\nu(1-z) = B_\nu(-z) - \nu e^{-i\pi\nu} z^{\nu-1}.$$

Combining this with (2.6) and eliminating  $z^{\nu-1}$  we obtain the relation

$$e^{i\pi\nu} B_\nu(1-z) - B_\nu(z) = e^{i\pi\nu} B_\nu(-z) - B_\nu(1+z), \quad \Im z > 0,$$

which says that the left-hand side is a periodic function of  $z$  with period 1. In other words,

$$B_\nu(z) = e^{i\pi\nu} B_\nu(1-z) + \omega_\nu^+(z), \quad \Im z > 0, \quad (2.8)$$

where  $\omega_\nu^+(z)$  is a 1-periodic function in the upper half plane. In a similar way we obtain

$$B_\nu(z) = e^{-i\pi\nu} B_\nu(1-z) + \omega_\nu^-(z), \quad \Im z < 0, \quad (2.9)$$

where  $\omega_\nu^-(z)$  is a 1-periodic function in the lower half plane.

The functions  $\omega_\nu^\pm(z)$  can be obtained as follows. Consider (2.5) with  $\mu = 1$  and  $\Im z > 0$ . As we did for the analytic continuation we can turn the path of integration  $\mathcal{C}$  into the upper half plane, even across the poles at  $t_k = 2\pi ik$ ,  $k = 1, 2, 3, \dots$ , and pick up the residues. Summing the residues, which can be done when  $\Im z > 0$ , and taking into account the value of the phases of  $t$  at both sides of the cut when both branches of  $\mathcal{C}$  pass the poles, we obtain

$$B_\nu(z) = \Gamma(\nu+1) \left[ e^{\frac{3}{2}\pi\nu i} - e^{-\frac{1}{2}\pi\nu i} \right] \sum_{k=1}^{\infty} \frac{e^{2\pi ikz}}{(2\pi k)^\nu} + \frac{\Gamma(\nu+1)}{2\pi i} \int_{\mathcal{C}} \frac{e^{zt}}{(e^t-1)t^\nu} dt, \quad (2.10)$$

where  $\mathcal{C}$  runs around the cut, which now occurs in the first quadrant of the  $t$ -plane. When  $\frac{1}{2}\pi < \arg z \leq \pi$  we can take the cut along the positive  $t$ -axis. At the upper part of the cut we have  $\arg t = -2\pi$ , at the lower side  $\arg t = 0$ . The contour starts at  $+\infty$  (at the upper side of the cut) and encircles the origin in positive direction.

In this position of the contour we introduce a new variable of integration by writing  $t = ve^{-i\pi}$ . By using the relation

$$\frac{e^{-zv}}{e^{-v}-1} = -\frac{e^{(1-z)v}}{e^v-1}$$

and interpreting the new integral in terms of  $B_\nu(1-z)$ , we obtain the functional equation (2.8) with

$$\omega_\nu^+(z) = 2i \sin \pi\nu e^{\frac{1}{2}\pi\nu i} \Gamma(\nu+1) \sum_{k=1}^{\infty} \frac{e^{2\pi ikz}}{(2\pi k)^\nu}. \quad (2.11)$$

This relation holds for all values of  $z$  in the upper half plane, since all three terms in (2.8) are analytic functions with respect to  $z$  in this domain;  $\nu$  may be any complex number.

Repeating the procedure for values of  $z$  in the lower half plane, we obtain (2.9) with

$$\omega_{\nu}^{-}(z) = -2i \sin \pi \nu e^{-\frac{1}{2}\pi \nu i} \Gamma(\nu + 1) \sum_{k=1}^{\infty} \frac{e^{-2\pi i k z}}{(2\pi k)^{\nu}}, \quad (2.12)$$

a result as in (2.11), with all quantities  $i$  replaced by  $-i$ .

We can now define the Bernoulli function  $B_{\nu}(x)$  for  $x < 0$ . This will depend on the way we approach the negative  $z$ -axis: from above or from below. Taking the average of the two values obtained so, we define

$$B_{\nu}^{*}(x) := \lim_{y \rightarrow 0} \frac{B_{\nu}(x + iy) + B_{\nu}(x - iy)}{2}, \quad x < 0. \quad (2.13)$$

It easily follows that we have

$$B_{\nu}^{*}(-x) = \cos \pi \nu B_{\nu}(x + 1) + 2\Gamma(\nu + 1) \sin \pi \nu \sum_{k=1}^{\infty} \frac{\sin(2\pi k x - \frac{1}{2}\nu \pi)}{(2\pi k)^{\nu}}, \quad (2.14)$$

where  $x > 0$ ,  $\Re \nu > 1$ , the latter condition being needed to guarantee the convergence of the infinite series. Again, the series is a 1-periodic function on the real line. The function  $B_{\nu}^{*}(x)$  satisfies the following difference property (compare this with (2.1)):

$$B_{\nu}(x + 1) - B_{\nu}(x) = \begin{cases} \nu x^{\nu-1}, & \text{if } x \geq 0; \\ -\nu |x|^{\nu-1} \cos \pi \nu, & \text{if } x < 0, \Re \nu > 1. \end{cases} \quad (2.15)$$

The series in (2.14) is closely connected with the familiar Fourier series for the Bernoulli polynomials:

$$B_n(x) = -2n! \sum_{k=1}^{\infty} \frac{\cos(2\pi k x - \frac{1}{2}n\pi)}{(2\pi k)^n},$$

$n = 1, 2, 3, \dots, x \in [0, 1)$ .

In BUTZER *et al.* (1992) a quite different approach and result is given for defining the value of  $B_{\nu}(z)$  for negative values of  $z$ . Our approach, which leads to (2.14) and (2.15), is based on the crucial functional relations in (2.8) and (2.9), with (2.11) and (2.12). These relations are not available in the cited reference, and there the difference property (2.15) contains for  $x < 0$  the factor  $\cos \pi \nu - \sin \pi \nu$  instead of only  $\cos \pi \nu$ . In our approach the relation for  $x < 0$  links up nicely with the original difference relation in (2.1), because in order to replace  $(-1)^n$  we just take the average of  $e^{\pm i\pi \nu}$ .

## 2.2. Series in powers of $z$

We conclude by giving the Maclaurin series (in powers of  $z$ ) and an asymptotic expansions (in negative powers of  $z$ ) of  $B_{\nu}(z)$ . These expansions have received little or no attention in the literature.

The well-known property

$$B_n(z) = \sum_{k=0}^n B_k \binom{n}{k} z^{n-k} \quad (2.16)$$

holds for the Bernoulli functions in the form of an asymptotic expansion:

$$B_\nu(z) \sim \sum_{k=0}^{\infty} B_k \binom{\nu}{k} z^{\nu-k}, \quad \text{as } z \rightarrow \infty \quad (2.17)$$

in the sector  $|\arg z| < \pi$ . This follows by taking in (2.5)  $\mu = 1$  and expanding

$$\frac{t}{e^t - 1} = \sum_{k=0}^{\infty} B_k \frac{t^k}{k!}.$$

Interchanging the order of summation and integration, applying Watson's Lemma for loop integrals (see OLVER (1974)), and using Hankel's contour integral for the reciprocal gamma function

$$\frac{1}{\Gamma(z)} = \frac{1}{2\pi i} \int_{\mathcal{C}} e^t t^{-z} dt$$

(where  $\mathcal{C}$  is the same as in (2.5)), we obtain (2.17).

It follows that, when  $\Re \nu < 0$ ,

$$B_\nu(z) \rightarrow 0, \quad \text{as } z \rightarrow \infty$$

in the sector  $|\arg z| < \pi$ . An application of this yields an interesting relation with the generalized zeta function, which is defined by

$$\zeta(s, t) = \sum_{n=0}^{\infty} (n+t)^{-s}, \quad \Re s > 1, \quad t \neq 0, -1, -2, \dots, \quad (2.18)$$

and which reduces to the familiar Riemann zeta function when  $t = 1$ :  $\zeta(s) = \zeta(s, 1)$ . Observe that repeated application of (2.6) gives

$$B_\nu(z+m) = B_\nu(z) + \nu \sum_{k=0}^{m-1} (z+k)^{\nu-1}. \quad (2.19)$$

When  $m$  tends to infinity and  $\Re \nu < 0$  the left-hand side vanishes. It follows that

$$B_\nu(z) = -\nu \zeta(1-\nu, z), \quad z \neq 0, -1, -2, \dots \quad (2.20)$$

By using analytic continuation it follows that this relation holds for all complex values of  $\nu$ . The function  $\zeta(s, t)$  has a pole at  $s = 1$ , with residue 1. Hence, the right-hand side of (2.20) is well defined as  $\nu \rightarrow 0$ .



From the expansion

$$\zeta(s, t) = \frac{1}{\Gamma(s)} \sum_{k=0}^{\infty} \Gamma(s+k) \zeta(s+k) \frac{(1-t)^k}{k!}, \quad |t-1| < 1,$$

which easily follows by expanding in (2.18)

$$(n+t)^{-s} = (n+1)^{-s} \left[ 1 + \frac{t-1}{n+1} \right]^{-s}$$

in powers of  $(t-1)$ , and using (2.1), we obtain

$$B_\nu(z) = -\nu z^{\nu-1} + \frac{1}{\Gamma(-\nu)} \sum_{k=0}^{\infty} \Gamma(k+1-\nu) \zeta(k+1-\nu) \frac{(-z)^k}{k!}, \quad |z| < 1.$$

This expansion reduces to the finite (polynomial) representation (2.16) when we take the limit  $\nu \rightarrow n$  (integer).

Both expansions (2.17) and (2.16) are contained in one integral:

$$B_\nu(z+1) = \frac{1}{\Gamma(-\nu) 2\pi i} \int_{\mathcal{L}} \zeta(1-\nu-w) \Gamma(w) \Gamma(1-\nu-w) z^{-w} dw, \quad (2.21)$$

where  $\Re \nu < -1$  and  $\mathcal{L}$  is a vertical in the strip  $0 < \Re w < -\nu$ . This integral follows from the Mellin transform of  $\zeta(s, t+1)$  with respect to  $t$ , which reads:

$$\int_0^\infty \zeta(s, t+1) t^{w-1} dt = \zeta(s-w) B(w, s-w), \quad 0 \Re w < \Re s - 1,$$

where we have used the Beta integral

$$\int_0^\infty t^{x-1} (t+1)^{-x-y} dt = B(x, y) = \Gamma(x) \Gamma(y) / \Gamma(x+y), \quad \Re x, y > 0.$$

Upon inverting the Mellin transform we obtain (2.21).

The expansions (2.17) and (2.16) follow from (2.21) by shifting the contour  $\mathcal{L}$  to the left, across the poles of the gamma function  $\Gamma(w)$ , and picking up the residues to obtain the Maclaurin expansion (2.16), and shifting to the right across the pole of  $\zeta(1-\nu-w)$  at  $w = -\nu$  and the poles of  $\Gamma(1-\nu-w)$  at  $w = k - \nu + 1, k = 0, 1, 2, \dots$ , to obtain the asymptotic expansion (2.17).

### 3. ASYMPTOTICS OF $B_\nu^\mu$

Our current interest in the asymptotic behaviour of the generalized Bernoulli numbers  $B_\nu^\mu$  stems from our earlier research on *Stirling numbers*, as published recently in TEMME (1993). Indeed, the quantities  $B_\nu^\mu$  are related with Stirling numbers. First we explain this relationship.

The Stirling numbers of the first and second kind, respectively denoted by  $S_n^{(m)}$  and  $\mathcal{S}_n^{(m)}$ , are usually defined through the finite generating functions

$$x(x-1)\cdots(x-n+1) = \sum_{m=0}^n S_n^{(m)} x^m, \quad (3.1)$$

$$x^n = \sum_{m=0}^n \mathcal{S}_n^{(m)} x(x-1)\cdots(x-m+1), \quad (3.2)$$

where we give the left-hand side of (3.1) the value 1 if  $n = 0$ . Similarly, the factors on the right-hand side of (3.2) have the value 1 if  $m = 0$ . This gives the ‘boundary values’

$$S_n^{(n)} = \mathcal{S}_n^{(n)} = 1, \quad n \geq 0, \quad \text{and} \quad S_n^{(0)} = \mathcal{S}_n^{(0)} = 0, \quad n \geq 1.$$

Furthermore it is convenient to agree on  $S_n^{(m)} = \mathcal{S}_n^{(m)} = 0$  if  $m > n$ .

Several other generating functions are available for Stirling numbers. We have

$$\frac{[\ln(x+1)]^m}{m!} = \sum_{n=m}^{\infty} S_n^{(m)} \frac{x^n}{n!}, \quad (3.3)$$

$$\frac{(e^x - 1)^m}{m!} = \sum_{n=m}^{\infty} \mathcal{S}_n^{(m)} \frac{x^n}{n!}. \quad (3.4)$$

These two equations give the link with the generating functions of the generalized Bernoulli numbers given in (1.1). The relations are

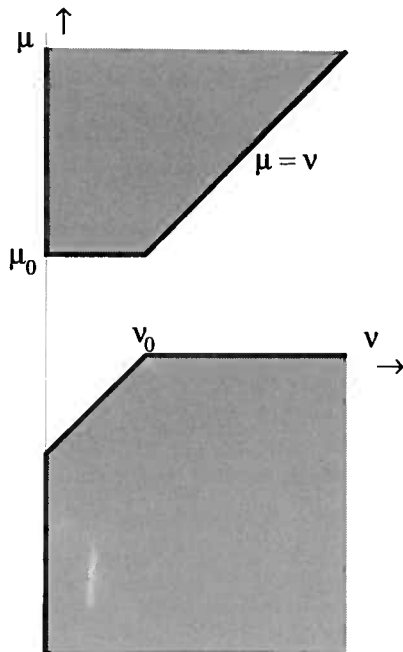
$$S_n^{(m)} = \binom{n-1}{m-1} B_{n-m}^n, \quad \mathcal{S}_n^{(m)} = \binom{n}{m} B_{n-m}^{-m}. \quad (3.5)$$

To explain this for the numbers of the first kind, we write

$$S_n^{(m)} = \frac{1}{2\pi i} \frac{n!}{m!} \int_{\mathcal{C}} \frac{[\ln(z+1)]^m}{z^{n+1}} dz,$$

where  $\mathcal{C}$  is a small circle around  $z = 0$ . Substituting  $z = e^w - 1$  and integrating by parts gives an integral that is similar to (1.2). For the Stirling numbers of the second kind the relation with the numbers  $B_\nu^\mu$  is quite straightforward.

When we consider the asymptotic problem for  $B_\nu^\mu$  we take  $\nu$  as the large parameter. The parameter  $\mu$  may have any complex value in the definition of  $B_\nu^\mu$ , and the asymptotic behaviour of this number strongly depends on the value of  $\mu$ . In our paper TEMME (1993) we have derived new asymptotic approximations of the Stirling numbers of both kinds, which hold when  $n$  is large and which are uniformly valid for  $m \in [0, n]$ . Although the Stirling numbers are defined for integer values of  $n, m$ , the results and methods can be interpreted for continuous variables. Considering the relations in (3.5), we



**Figure 1.** Parameter domains (shaded) for which the uniform asymptotic expansions of the Stirling numbers can be used to obtain a first order approximation for  $B_\nu^\mu$ ; upper area via the Stirling numbers of the first kind, lower part via the Stirling numbers of the second kind.

observe that the uniform asymptotic results of the Stirling numbers numbers can be used for the generalized Bernoulli numbers  $B_\nu^\mu$  in the shaded areas of the  $(\nu, \mu)$ -plane, given in Figure 1. Here  $\nu_0, \mu_0$  are large numbers,  $\nu_0$  indicating the large  $\nu$ -domain  $[\nu_0, \infty)$  for which the uniform approximations of the Stirling numbers can be used for the generalized Bernoulli numbers  $B_\nu^\mu$ .

In §3.1 and §3.2 we concentrate on the asymptotic behaviour of  $B_\nu^\mu$  for  $(\nu, \mu)$  in the non-shaded area in the upper right half plane, that is,  $\nu$  large and  $0 \leq \mu \leq \nu$ . In fact our goal is to obtain a uniform approximation in this domain, as we obtained for the Stirling numbers in the shaded areas. However, the situation here is quite different from the Stirling case, as will be explained in §3.2. In §3.3 we consider a problem for  $B_\nu(z)$  in which  $z$  is large and  $\nu$  acts as a uniformity parameter on the real axis. First we summarize existing results from the literature.

### 3.1. Nörlund's results.

In NÖRLUND (1961) results are given for a parameter domain that corresponds to the neighbourhood of the diagonal  $\nu = \mu$ . In fact, Nörlund considered the

polynomials  $B_\nu^{\nu+\rho+1}(z)$ , where  $\rho$  and  $z$  are fixed complex numbers (*fixed* means independent of  $\nu$ ). His result is

$$\frac{B_\nu^{\nu+\rho+1}(z)}{\nu!} \sim (-1)^\nu \frac{(\ln \nu)^\rho}{\nu^z} \left[ \sum_{s=0}^{n-1} \binom{\rho}{s} \frac{(-1)^s}{(\ln \nu)^s} A_s(z) + \mathcal{O}((\ln \nu)^{-n}) \right], \quad (3.6)$$

as  $\nu \rightarrow \infty$ . The coefficients  $A_s(z)$  are derivatives of the reciprocal gamma function:

$$A_s(z) = \frac{d^s}{dz^s} \frac{1}{\Gamma(1-z)}.$$

The asymptotic expansion (3.6) shows inverse powers of  $\ln \nu$ , giving a rather slow asymptotic convergence for computations, unless  $\nu$  is very large. When  $z = 0$ , this is in fact the case  $S_{\nu+\rho+1}^{(\rho+1)}$  of Stirling numbers of the first kind, the coefficients  $A_s(z)$  reduce to the coefficients of the Maclaurin expansion of  $1/\Gamma(1-z)$ , which easily follow from those of  $1/\Gamma(z)$  (see, for instance, ABRAMOWITZ AND STEGUN (1964, page 256)).

When  $\rho = 0, 1, 2, \dots$ , the series in (3.6) reduces to a finite number of terms (because the binomial coefficient vanishes when  $s > \rho$ ). In particular, when  $\rho = 0$ , we have the simple case  $B_\nu^{\nu+1}(z) = (z-1)(z-2)\cdots(z-\nu)$ . That is,

$$\frac{B_\nu^{\nu+1}(z)}{\nu!} = (-1)^\nu \frac{\Gamma(\nu+1-z)}{\Gamma(1-z)\Gamma(\nu+1)} \sim \sum_{n=0}^{\infty} \frac{B_n^{1-z}}{n! \Gamma(1-z-n)} \frac{(-1)^n}{\nu^{z+n}},$$

which is a well-known result for the ratio of two gamma functions.

We observe that this expansion is in negative powers of  $\nu$ , because in (3.6) the expansion containing inverse powers of  $\ln \nu$  completely vanishes. What remains was hidden in the  $\mathcal{O}$ -symbol of (3.6) and shows up when  $\rho = 0$  (quantities that are asymptotically negligible with respect to all negative powers of  $\ln \nu$  occurring in the series and the  $\mathcal{O}$ -term in (3.6)). This is a nice example in which 'exponentially small terms' become important when a parameter changes a critical value (in this case: when  $\rho = 0$ ).

For fixed values of  $\mu$  Nörlund gives the expansion

$$\frac{B_\nu^\mu(z)}{\nu!} = \nu^{\mu-1} \frac{2 \cos \pi(2z + \mu - \frac{1}{2}\nu)}{\Gamma(\mu)(2\pi)^\nu} [1 + \mathcal{O}(\nu^{-1})].$$

### 3.2. Saddle point methods for $B_\nu^\mu$ .

We now discuss asymptotic properties of the generalized Bernoulli numbers  $B_\nu^\mu$  in connection with our previous results for the Stirling numbers. Consider (1.2) with  $z = 0$  and  $\nu \neq \mu$ , and integrate by parts. It follows that

$$B_\nu^\mu = \frac{\mu}{\mu - \nu} \frac{\Gamma(\nu + 1)}{2\pi i} \int_C \frac{t^{\mu-\nu} e^t}{(e^t - 1)^{\mu+1}} dt.$$

This integral has better convergence properties when we deform the contour  $\mathcal{C}$  into a path that extends to  $-\infty$ . We write

$$B_\nu^\mu = \frac{\mu}{\mu - \nu} \frac{\Gamma(\nu + 1)}{2\pi i} \int_{\mathcal{C}} e^{\phi(t)} \frac{e^t dt}{e^t - 1}, \quad (3.7)$$

where

$$\phi(t) = (\mu - \nu) \ln t - \mu \ln(e^t - 1).$$

In the saddle point method one tries to deform the contour  $\mathcal{C}$  through one or more saddle points of the integrand. To calculate the saddle points we have to solve the equation  $\frac{d}{dt}\phi(t) = 0$ , which is equivalent to solving

$$1 - e^{-t} = \lambda t, \quad \lambda = \frac{\mu}{\mu - \nu}. \quad (3.8)$$

The solution  $t = 0$  is not of interest, because the contour  $\mathcal{C}$  is not allowed to pass through the origin. To keep the discussion surveyable we assume that  $\nu$  is large and positive, and that  $\mu$  is a real parameter.

We can distinguish three  $\mu$ -domains of interest, which correspond with the three domains indicated in Figure 1.

- (i)  $\mu < 0 \Rightarrow 0 < \lambda < 1$ ; in this case (3.7) has a real positive solution;
- (ii)  $0 < \mu < \nu \Rightarrow \lambda < 0$ ; in this case (3.7) has no real solutions;
- (iii)  $\mu > \nu \Rightarrow \lambda > 1$ ; in this case (3.7) has a real negative solution.

We conclude that in the two shaded areas of Figure 1 (both ‘Stirling cases’) there is a real saddle point, and that in the area that has to be done there is no real saddle point. It will turn out that in the latter case, that is, when  $\lambda < 0$ , equation (3.8) has complex solutions, which occur in complex conjugated pairs, and one pair can be used for the saddle point method.

Equation (3.8) is equivalent to the equation

$$we^w = x, \quad \text{where } w = t - \frac{1}{\lambda}, \quad x = -\frac{1}{\lambda} e^{-\frac{1}{\lambda}}.$$

When  $\lambda$  ranges through the interval  $(-\infty, 0)$  the quantity  $x$  ranges through the interval  $(0, +\infty)$ . The trivial solutions  $w = -\frac{1}{\lambda}$  is not of interest. The equation  $we^w = x$  has received quite some attention in the literature. MAPLE, the package for symbolic computations, has the solution  $w(x)$  as a standard function. To give more insight on the location of the complex solutions of this equation, we give a few steps in solving the equation for real positive  $x$ .

We write  $w = u + iv$ , with  $u, v$  real, and see that the equation  $we^w = x$  is equivalent to

$$v = -xe^{-u} \sin v, \quad u = -v \cot v.$$

For positive values of  $x$  solutions occur in the  $v$ -intervals

$$(\pi, 2\pi), (3\pi, 4\pi), \dots$$

and in similar negative  $v$ -intervals. When  $x$  is small, that is,  $-\lambda$  is a large positive number, a conjugate pair of saddle points  $t^\pm$  has imaginary parts near  $\pm\pi$  and the real parts satisfy  $\Re t^\pm \sim -\ln(-\lambda)$ . Because of the convergence of the integral in (3.9) at  $t = \pm\infty$ , the contour  $\mathcal{C}$  can be deformed into two separate conjugate paths  $\mathcal{C}^\pm$ ,  $\mathcal{C}^-$  running from  $-\infty$  to  $+\infty$  with  $\Im t \in (-\pi, -2\pi)$ , and  $\mathcal{C}^+$  from  $+\infty$  to  $-\infty$  with  $\Im t \in (\pi, 2\pi)$ , such that  $\mathcal{C}^\pm$  run through the saddle points  $t^\pm$ . Locally at  $t = t^\pm$  we can approximate  $\phi(t)$  up to the quadratic term of its Maclaurin expansion, and we obtain the asymptotic result

$$B_\nu^\mu \sim \frac{\lambda\Gamma(\nu+1)}{2\pi i} \sum_{(+,-)} \frac{e^{\phi(t^\pm)}}{\lambda t^\pm} \int_{\mathcal{C}^\pm} e^{\frac{1}{2}\phi''(t^\pm)(t-t^\pm)^2} dt.$$

That is,

$$B_\nu^\mu \sim \frac{\Gamma(\nu+1)}{\sqrt{2\pi i}} \left[ \frac{e^{\phi(t^-)}}{t^- \sqrt{-\phi''(t^-)}} - \frac{e^{\phi(t^+)}}{t^+ \sqrt{-\phi''(t^+)}} \right]. \quad (3.10)$$

We have for the second derivative of  $\phi$ :

$$\phi''(t) = -\frac{\mu-\nu}{t^2} + \frac{\mu e^t}{(e^t-1)^2}.$$

Evaluating this at the saddle points, using  $1 - \lambda t^\pm = e^{-t^\pm}$ , see (3.11), we have

$$\phi''(t^\pm) = \frac{\nu-\mu}{\lambda t^\pm} [\lambda(1+t^\pm) - 1].$$

These quantities have negative real parts when  $-\lambda$  is a large positive number.

The first approximation given in (3.10) can be supplied with more terms by using standard techniques of the saddle point method, but we omit the details here. Also, it is possible to repeat the analysis for the generalized Bernoulli polynomials  $B_\nu^\mu(z)$ , and to compare the results with Nörlund's results. All this is outside the scope of the present paper, because the elaborations are rather technical and complicated. Moreover, further investigations are needed to determine the range of the parameters for which the expansion holds. We expect that (3.10) will be uniformly valid for  $\lambda = \mu/(\mu - \nu)$  belonging to compact sets of the interval  $(-\infty, 0)$ , and  $\nu \rightarrow +\infty$ . When indeed this is true, we can fill a large part of the unshaded area in the first quadrant of Figure 1.

### 3.3. Uniform asymptotics for large values of $z$

We return to  $B_\nu(z)$  and consider the problem of obtaining an expansion for large values of  $z$  and  $\nu$ . When  $\nu$  is fixed the expansion in (2.17) is applicable. In this subsection we give two expansions, one holding uniformly with respect to  $\nu \in [0, \infty)$ , and a similar expansion holding uniformly with respect to  $\nu \in (-\infty, 0]$ . The approach is based on earlier work discussed in TEMME (1983).

The asymptotic problem in that paper is to obtain an expansion of the integral

$$F_\lambda(z) = \frac{1}{\Gamma(\lambda)} \int_0^\infty t^{\lambda-1} e^{-zt} f(t) dt, \quad z, \lambda > 0, \quad (3.12)$$

that holds uniformly with respect to  $\lambda \in [0, \infty)$ . Laplace integrals can be expanded by invoking Watson's Lemma (see OLVER (1974) or WONG (1989)): expand  $f$  at the origin and interchange summation and integration. That is,

$$f(t) = \sum_{n=0}^{\infty} c_n t^n \quad \Rightarrow \quad F_\lambda(z) \sim \sum_{n=0}^{\infty} c_n \frac{\Gamma(\lambda + n)}{\Gamma(\lambda)} z^{-n-\lambda}$$

as  $z \rightarrow \infty$ ,  $\lambda$  fixed. When  $\lambda$  is not fixed (say,  $\lambda$  is depending on  $z$ ) this becomes invalid. It is better to expand at  $t = \kappa := \lambda/z$ , the saddle point of the dominant part  $t^\lambda e^{-zt}$  of the integrand. We have

$$f(t) = \sum_{n=0}^{\infty} a_n(\kappa) (t - \kappa)^n \quad \Rightarrow \quad F_\lambda(z) \sim z^{-\lambda} \sum_{n=0}^{\infty} a_n(\kappa) P_n(\lambda) z^{-n}, \quad (3.13)$$

where

$$P_n(\lambda) = \frac{1}{\Gamma(\lambda)} \int_0^\infty t^{\lambda-1} e^{-zt} (t - \kappa)^n dt. \quad (3.14)$$

That is,

$$P_0(\lambda) = 1, \quad P_1(\lambda) = 0, \quad P_2(\lambda) = \lambda, \quad P_3(\lambda) = 2\lambda, \dots$$

It is quite easy to obtain the recursion:

$$P_{n+1}(\lambda) = n [P_n(\lambda) + \lambda P_{n-1}(\lambda)],$$

and the estimate

$$P_n(\lambda) = \mathcal{O}(\lambda^{[n/2]}), \quad \lambda \rightarrow \infty. \quad (3.15)$$

This expansion is, under mild conditions on  $a_n(\kappa)$ , that is, on  $f$ , uniformly valid with respect to  $\lambda \in [0, \infty)$ . For instance, when  $f(t) = 1/(t + 1)$ , the coefficients  $a_n(\kappa)$  are given by

$$\frac{1}{t + 1} = \sum_{n=0}^{\infty} a_n(\kappa) (t - \kappa)^n, \quad a_n(\kappa) = \frac{(-1)^n}{(1 + \kappa)^{n+1}}, \quad (3.16)$$

and we see from (3.15) that, in this case, the terms  $a_n(\kappa) P_n(\lambda) z^{-n}$  in the expansion of  $F_\lambda(z)$  given in (3.13) do not lose their asymptotic character when  $\lambda$  runs through the domain  $[0, \infty)$ . This is not a proof of the asymptotic nature of the expansion. but an indication that the expansion has some robustness with respect to large values of  $\lambda$ . For a proof we refer to TEMME (1983).

We can apply this method by writing  $B_\nu(z)$  in the form (3.12). This is possible when  $\nu$  is negative. We observe that in that case we can integrate in (2.5) along both sides of the negative real axis (using different phases  $\pm\pi i$  of  $t$ ), and obtain

$$B_\nu(z) = \frac{1}{\Gamma(-\nu)} \int_0^\infty t^{-\nu-1} e^{-zt} f(t) dt, \quad f(t) = \frac{t}{1 - e^{-t}}. \quad (3.17)$$

We define  $\kappa = -\nu/z$  and expand  $f$  at  $t = \kappa$  as in (3.13). In this case the expansion has a different asymptotic character than in the example with  $f(t) = 1/(t + 1)$ . To explain this, we have in the latter case the lucky situation that  $\{a_n\}$  constitute an asymptotic scale as  $\kappa \rightarrow \infty$ . That is,

$$a_{n+1}/a_n = \mathcal{O}(1/\kappa) \quad \text{as } \kappa \rightarrow \infty.$$

In fact, when this is the case, the expansion in of  $F_\lambda(z)$  in (3.13) has a double asymptotic property: it is also valid when  $\lambda \rightarrow \infty$ , uniformly with respect to  $z \in [z_0, \infty)$ , where  $z_0$  is a fixed positive number.

Let us now consider  $f$  defined in (3.17) for the case of the Bernoulli functions. We have, as  $t \rightarrow \infty$ ,

$$f(t) = t(1 + e^{-t} + e^{-2t} + \dots)$$

and

$$f'(t) = 1 + \mathcal{O}(te^{-t}), \quad f^{(n)}(t) = \mathcal{O}(te^{-t}), \quad n = 2, 3, 4, \dots$$

Hence, for  $n \geq 2$ , the coefficients  $a_n(\kappa)$  are asymptotically small. The only snag is that the coefficients do not constitute an asymptotic scale.

We conclude with giving a similar expansion for positive values of  $\nu$ . The starting point is the contour integral (2.5)

$$B_\nu(z) = \frac{\Gamma(\nu + 1)}{2\pi i} \int_{\mathcal{C}} e^{zt} f(t) \frac{dt}{t^{\nu+1}},$$

with

$$f(t) = \frac{t}{e^t - 1}.$$

Again, there is a saddle point at  $t = \kappa := \nu/z$  and we obtain

$$f(t) = \sum_{n=0}^{\infty} b_n(\kappa) (t - \kappa)^n \quad \Rightarrow \quad B_\nu(z) \sim z^{-\nu} \sum_{n=0}^{\infty} b_n(\kappa) Q_n(\nu) z^{-n}, \quad (3.18)$$

where

$$Q_n(\nu) = \frac{\Gamma(\lambda + 1)}{2\pi i} \int_{\mathcal{C}} e^{zt} (t - \kappa)^n \frac{dt}{t^{\nu+1}}.$$

It is easily verified that

$$Q_n(\nu) = (-1)^n P_n(-\nu), \quad n = 0, 1, 2, \dots,$$

where the polynomials  $P_n$  are given in (3.14), and that

$$a_0(\kappa) = b_0(\kappa) + \kappa, \quad a_1(\kappa) = b_1(\kappa) + 1, \quad a_n(\kappa) = b_n(\kappa), \quad n \geq 2.$$

That's why I call the expansions in (3.13) and (3.18) quite similar. Also, the expansion for  $B_\nu(z)$  for positive values of  $\nu$  has the same asymptotic nature



as the one for negative values of  $\nu$  given in (3.13). When  $n \geq 2$  the coefficients  $b_n(\kappa)$  are exponentially small when  $\kappa$  is large, and do not constitute an asymptotic scale.

I looked at him. There was so much I wanted to ask him, so much I wanted to say; but somehow I knew there wasn't time and even if there was, that it was all, somehow, beside the point.

Donna Tart  
*The Secret History*

#### BIBLIOGRAPHY

- [ 1 ] ABRAMOWITZ, M. & I.A. STEGUN (1964). *Handbook of mathematical functions with formulas, graphs and mathematical tables*, Nat. Bur. Standards Appl. Series, **55**, U.S. Government Printing Office, Washington, D.C.
- [ 2 ] BERNDT, B.C. (1985). *Ramanujan's Notebook, Part I*, Springer Verlag, Berlin and New York.
- [ 3 ] BÖHMER, P. (1910). Über die Bernoullischen Funktionen, *Math. Ann.*, **68** 338–360.
- [ 4 ] BUTZER, P.L. & M. HAUSS (1991). On Stirling functions of the second kind, *Studies Appl. Math.*, **84** 71–91.
- [ 5 ] BUTZER, P.L., M. HAUSER & M. LECLERC (1992). Bernoulli numbers and polynomials of arbitrary complex indices, *Appl. Math. Lett.*, **42** 83–88.
- [ 6 ] BUTZER, P.L., M. HAUSER & M. LECLERC (1992). *Extension of Euler polynomials to Euler functions  $E_\alpha(z)$  with complex indices*, Arbeitsbericht, Rheinisch-Westfälische Techn. Hochschule Aachen, Report 414/222-9-2/XII. Lehrstuhl A für Mathematik.
- [ 7 ] BUTZER, P.L., M. HAUSS & M. SCHMIDT (1989). Factorial functions and Stirling numbers, *Results in Mathematics*, **16** 16–33.
- [ 8 ] EDWARDS, A.W.F. (1982). Sums of powers of integers: a little of the history, *Mat. Gaz.*, **66**, no. 435, 22–28.
- [ 9 ] JONQUIÈRE, A. (1891). Eine Verallgemeinerung der Bernoulli'schen Funktionen und ihren Zusammenhang mit der verallgemeinerten Riemann'schen Reihe, *Bihang Till K. Svenska Vet.-Akad. Handlingar*, **16**, Afd. I, no. 6. 1–28.

- [ 10] JORDAN, C. (1947). *The calculus of finite differences*, Chelsea Publishing Company, New York.
- [ 11] MAGNUS, W., F. OBERHETTINGER & R.P. SONI (1966). *Formulas and theorems for the special functions of mathematical physics*, Springer Verlag, Berlin and New York.
- [ 12] MILNE-THOMSON, L.M. (1933). *The calculus of finite differences*, Macmillan, London.
- [ 13] NÖRLUND, N.E. (1924). *Vorlesungen über Differenzenrechnung*, Springer Verlag, Berlin and New York.
- [ 14] NÖRLUND, N.E. (1961). Sur les valeurs asymptotiques des nombres et des polynômes de Bernoulli, *Rend. Circ. Mat. Palermo*, **10**, no. 1, 27–44.
- [ 15] OLVER, F.W.J. (1974). *Asymptotics and special functions*. Academic Press, New York.
- [ 16] TEMME, N.M. (1983). Uniform asymptotic expansions of Laplace integrals, *Analysis*, **3**, 221–249.
- [ 17] TEMME, N.M. (1993). Asymptotic estimates of Stirling numbers, *Stud. Appl. Math.*, **89**, 233–243.
- [ 18] WONG, R. (1989). *Asymptotic approximations of integrals*. Academic Press, New York.

# Kleene's Realizability

*for Cor Baayen*

A.S.Troelstra

*Faculteit Wiskunde en Informatica*

*Universiteit van Amsterdam*

*Plantage Muidergracht 24, 1018 TV Amsterdam (NL).*

This paper is a prepublication of the first section of an introductory survey on realizability, for the *Handbook of Proof Theory*, edited by S.Buss, to appear with North-Holland Publ. Co. S. Buss, U. Kohlenbach, H. Luckhardt, J.R. Moschovakis and J. van Oosten have commented on earlier drafts of this paper.

## 1 INTRODUCTION

*1.1.* The realizability interpretation of intuitionistic arithmetic was first introduced by S.C.Kleene (1945). It has turned out to be an extremely fruitful interpretation, widely applicable to axiomatic systems based on constructive logic, and yielding interesting results such as the consistency of Church's thesis with intuitionistic formalisms. Nowadays there is not just a single notion of realizability, but a whole family of notions, which of course resemble each other in certain respects.

Here we present a streamlined development of the formalized version of Kleene's original notion. We presuppose some (not much) familiarity with intuitionistic first-order predicate logic, classical Peano arithmetic, as well as elementary recursion theory; for the rest the paper is self-contained.

For the history of the topic, see (Troelstra 1973, Dragalin 1988).

*1.2.* Realizability by numbers introduced by Kleene as a semantics for intuitionistic arithmetic, by defining for arithmetical sentences  $A$  a notion "the number  $\mathbf{n}$  realizes  $A$ ", intended to capture some essential aspects of the intuitionistic meaning of  $A$ . Here  $\mathbf{n}$  is not a term of the arithmetical formalism, but an element of the natural numbers  $\mathbb{N}$ . The definition is by induction on the complexity of  $A$ :

- $\mathbf{n}$  realizes  $t = s$  iff  $t = s$  holds;
- $\mathbf{n}$  realizes  $A \wedge B$  iff  $\mathbf{p}_0\mathbf{n}$  realizes  $A$  and  $\mathbf{p}_1\mathbf{n}$  realizes  $B$ ;

- $\mathbf{n}$  realizes  $A \vee B$  iff  $\mathbf{p}_0\mathbf{n} = 0$  and  $\mathbf{p}_1\mathbf{n}$  realizes  $A$  or  $\mathbf{p}_0\mathbf{n} = 1$  and  $\mathbf{p}_1\mathbf{n}$  realizes  $B$ ;
- $\mathbf{n}$  realizes  $A \rightarrow B$  iff for all  $\mathbf{m}$  realizing  $A$ ,  $\mathbf{n}\bullet\mathbf{m}$  is defined and realizes  $B$ ;
- $\mathbf{n}$  realizes  $\neg A$  if for no  $\mathbf{m}$ ,  $\mathbf{m}$  realizes  $A$ ;
- $\mathbf{n}$  realizes  $\exists y A$  iff  $\mathbf{p}_1\mathbf{n}$  realizes  $A[y/\overline{\mathbf{p}_0\mathbf{n}}]$ .
- $\mathbf{n}$  realizes  $\forall y A$  iff  $\mathbf{n}\bullet\mathbf{m}$  is defined and realizes  $A[y/\overline{\mathbf{m}}]$ , for all  $\mathbf{m}$ .

Here  $\mathbf{p}_1$  and  $\mathbf{p}_0$  are the inverses of some standard primitive recursive pairing function  $\mathbf{p}$  coding  $\mathbb{N}^2$  onto  $\mathbb{N}$ , and  $\overline{\mathbf{m}}$  is the standard term  $S^{\mathbf{m}}0$  (numeral) in the language of intuitionistic arithmetic corresponding to  $\mathbf{m}$ ;  $\bullet$  is partial recursive function application, i.e.  $\mathbf{n}\bullet\mathbf{m}$  is the result of applying the function with code  $\mathbf{n}$  to  $\mathbf{m}$ . (Later on we also use  $\overline{m}, \overline{n}, \dots$  for numerals.) The definition may be extended to formulas with free variables by stipulating that  $\mathbf{n}$  realizes  $A$  if  $\mathbf{n}$  realizes the universal closure of  $A$ .

Reading “there is a number realizing  $A$ ” as “ $A$  is constructively true”, we see that a realizing number provides witnesses for the constructive truth of existential quantifiers and disjunctions, and in implications carries this type of information from premise to conclusion by means of partial recursive operators. In short, realizing numbers “hereditarily” encode information about the realization of existential quantifiers and disjunctions.

1.3. Realizability, as an interpretation of “constructively true” is reminiscent of the well-known Brouwer-Heyting-Kolmogorov explanation (BHK for short) of the intuitionistic meaning of the logical connectives. BHK explains “ $p$  proves  $A$ ” for compound  $A$  in terms of the provability of the components of  $A$ . For prime formulas the notion of proof is supposed to be given. Examples of the clauses of BHK are:

- $p$  proves  $A \rightarrow B$  iff  $p$  is a construction transforming any proof  $c$  of  $A$  into a proof  $p(c)$  of  $B$ ;
- $p$  proves  $A \wedge B$  iff  $p = (p_0, p_1)$  and  $p_0$  proves  $A$ ,  $p_1$  proves  $B$ ;
- $p$  proves  $A \vee B$  iff  $p = (p_0, p_1)$  with  $p_0 \in \{0, 1\}$ , and  $p_1$  proves  $A$  if  $p_0 = 0$ ,  $p_1$  proves  $B$  if  $p_0 \neq 0$ .

Realizability corresponds to BHK if (a) we concentrate on (numerical) information concerning the realizations of existential quantifiers and the choices for disjunctions, and (b) the constructions considered for  $\forall, \rightarrow$  are assumed to be encoded by (partial) recursive operations.

1.4. Realizability gives a classically meaningful definition of intuitionistic truth; the set of realizable statements is closed under deduction and must be consistent, since  $1=0$  cannot be realizable. It is to be noted that decidedly non-classical principles are realizable, for example

$$\neg\forall x[\exists yTxy \vee \forall y\neg Txy]$$

is easily seen to be realizable. ( $T$  is Kleene's T-predicate, which is assumed to be available in our language;  $Txyz$  is primitive recursive in  $x, y, z$  and expresses that the algorithm with code  $x$  applied to argument  $y$  yields a computation with code  $z$ ;  $U$  is a primitive recursive function extracting from a computation code  $z$  the result  $Uz$ .) For  $\neg A$  is realizable iff no number realizes  $A$ , and realizability of  $\forall x[\exists yTxy \vee \forall y\neg Txy]$  requires a total recursive function deciding  $\exists yTxy$ , which does not exist (more about this below). In this way realizability shows how in constructive mathematics principles may be incorporated which cause it to diverge from the corresponding classical theory, instead of just being included in the classical theory.

1.5. Some notational habits adopted in this paper are: dropping of distinguishing sub- and superscripts where the context permits; saving on parentheses, e.g. for a binary predicate  $R$  applied to  $x, y$  we often write  $Rxy$  instead of  $R(x, y)$  (this habit has just been demonstrated above). The symbol  $\equiv$  is used for literal identity of expressions modulo renaming of bound variables.  $\Rightarrow$  is used as metamathematical consequence relation, and in particular  $\mathcal{A}, \mathcal{B} \Rightarrow \mathcal{C}$  expresses a rule which derives  $\mathcal{C}$  from premises  $\mathcal{A}, \mathcal{B}$ .  $FV(\mathcal{A})$  is the set of free variables of expression  $\mathcal{A}$ .

## 2 FORMALIZING REALIZABILITY IN HA

2.1. In order to exploit realizability proof-theoretically, we have to formalize it. Let us first discuss its formalization in ordinary intuitionistic first-order arithmetic HA ("Heyting's Arithmetic"), based on intuitionistic predicate logic with equality, and containing symbols for all primitive recursive functions, with their recursion equations as axioms. Induction and successor axioms  $Sx = Sy \rightarrow x = y$ ,  $Sx \neq 0$  are present as usual.

$x, y, z, \dots$  are numerical variables,  $S$  is successor. We use the notation  $\bar{n}$  for the term  $S^n 0$ ; such terms are called *numerals*.  $\mathbf{p}_0, \mathbf{p}_1$  bind stronger than infix binary operations, i.e.  $\mathbf{p}_0 t + s$  is  $(\mathbf{p}_0 t) + s$ . For primitive recursive predicates  $R, Rt_1 \dots t_n$  may be treated as a prime formula since the formalism contains a symbol for the characteristic function  $\chi_R$ .

Now we are ready for a formalized definition of " $x$  realizes  $A$ " in HA.

2.2. DEFINITION. By recursion on the complexity of  $A$  we define  $x \underline{rn} A$ ,  $x \notin FV(A)$ , " $x$  numerically realizes  $A$ " :

$$\begin{aligned}
x \underline{\text{rn}}(t = s) &:= (t = s) \\
x \underline{\text{rn}}(A \wedge B) &:= (\mathbf{p}_0 x \underline{\text{rn}} A) \wedge (\mathbf{p}_1 x \underline{\text{rn}} B), \\
x \underline{\text{rn}}(A \rightarrow B) &:= \forall y (y \underline{\text{rn}} A \rightarrow \exists z (Txyz \wedge Uz \underline{\text{rn}} B)), \\
x \underline{\text{rn}} \forall y A &:= \forall y \exists z (Txyz \wedge Uz \underline{\text{rn}} A), \\
x \underline{\text{rn}} \exists y A &:= \mathbf{p}_1 x \underline{\text{rn}} A[y/\mathbf{p}_0 x].
\end{aligned}$$

Note that  $\text{FV}(x \underline{\text{rn}} A) \subset \{x\} \cup \text{FV}(A)$ .  $\square$

2.3. REMARKS. (i) We have omitted clauses for negation and disjunction, since in arithmetic we can take  $\neg A := A \rightarrow 1 = 0$ ,  $A \vee B := \exists x((x = 0 \rightarrow A) \wedge (x \neq 0 \rightarrow B))$ . If we spell out  $x \underline{\text{rn}}(A \vee B)$  on the basis of this definition we find:

$$x \underline{\text{rn}}(A \vee B) \leftrightarrow (\mathbf{p}_0 x = 0 \rightarrow (\mathbf{p}_0 \mathbf{p}_1 x) 0 \underline{\text{rn}} A) \wedge (\mathbf{p}_0 x \neq 0 \rightarrow (\mathbf{p}_1 \mathbf{p}_1 x) 0 \underline{\text{rn}} B),$$

(ii) The definition of realizability permits slight variations, e.g. for the first clause we might have taken

$$x \underline{\text{rn}}'(t = s) := (x = t \wedge t = s).$$

However, it is routine to see that this variant  $\underline{\text{rn}}'$ -realizability is *equivalent* to  $\underline{\text{rn}}$ -realizability in the following sense: for each formula  $A$  there are two partial recursive functions  $\phi_A$  and  $\psi_A$  such that

$$\begin{aligned}
&\vdash x \underline{\text{rn}} A \rightarrow \phi_A(x) \underline{\text{rn}}' A \\
&\vdash x \underline{\text{rn}}' A \rightarrow \psi_A(x) \underline{\text{rn}} A.
\end{aligned}$$

(If in the future we shall call two versions of a realizability notion equivalent, it will always be in this or a similar sense.) Similarly, if we treat  $\vee$  as a primitive, the clause for  $x \underline{\text{rn}}(A \vee B)$  given above may be simplified to

$$x \underline{\text{rn}}(A \vee B) := (\mathbf{p}_0 x = 0 \wedge \mathbf{p}_1 x \underline{\text{rn}} A) \vee (\mathbf{p}_0 x \neq 0 \wedge \mathbf{p}_1 x \underline{\text{rn}} B),$$

which yields an equivalent notion of realizability.

(iii) In terms of partial recursive function application  $\bullet$  and the definedness predicate  $\downarrow$  ( $t \downarrow$  means “ $t$  is defined”), we can write more succinctly:

$$\begin{aligned}
x \underline{\text{rn}}(A \rightarrow B) &:= \forall y (y \underline{\text{rn}} A \rightarrow x \bullet y \downarrow \wedge x \bullet y \underline{\text{rn}} B), \\
x \underline{\text{rn}} \forall y A &:= \forall y (x \bullet y \downarrow \wedge x \bullet y \underline{\text{rn}} B).
\end{aligned}$$

where  $t \downarrow$  expresses that  $t$  is defined (cf. next subsection). Of course, the partial operation  $\bullet$  and the definedness predicate  $\downarrow$  are not part of the language, but expressions containing them may be treated as abbreviations, using the following equivalences:

$$\begin{aligned}
t_1 = t_2 &\leftrightarrow \exists x (t_1 = x \wedge t_2 = x), \\
t_1 \bullet t_2 = x &\leftrightarrow \exists y z u (t_1 = y \wedge t_2 = z \wedge T y z u \wedge U u = x), \\
t \downarrow &\leftrightarrow \exists z (t = z).
\end{aligned}$$

( $t_1, t_2$  terms containing  $\bullet$ ,  $x, y, z, u$  not free in  $t_1, t_2$ ). However, note that the logical complexity of  $A(t)$ , where  $t$  is an expression containing  $\bullet$ , depends on the complexity of  $t!$  (On the other hand,  $t\downarrow$  is always expressible in  $\Sigma_1^0$ -form.) For metamathematical investigations it is therefore more convenient to formalize realizability in a conservative extension  $\mathbf{HA}^*$  of  $\mathbf{HA}$  in which we can treat “ $\bullet$ ” as a primitive. Treating  $t_1 = t_2$  for partially defined  $t_1, t_2$  as an abbreviation in a rigorous way is possible, but involves a good deal of lengthy inductions, as demonstrated in (Kleene 1969). Since ordinary logic deals with total functions only, we first need to extend our logic to the (intuitionistic) logic of partial terms LPT, or intuitionistic  $E^+$ -logic, in the terminology of Troelstra and van Dalen(1988, 2.2.3). LPT first appeared in (Beeson 1981).

### 3 INTUITIONISTIC PREDICATE LOGIC WITH PARTIAL TERMS LPT

3.1. Variables are supposed to range over the objects of the domain considered, so always denote; arbitrary terms need not denote, so we need a predicate  $\mathbf{E}$ , expressing definedness;  $\mathbf{E}t$  reads “ $t$  denotes” or “ $t$  is defined”. Instead of  $\mathbf{E}t$  we shall write  $t\downarrow$ , in the notation commonly used in recursion theory.

If we also have equality in our logic, and read  $t = s$  as “ $t$  and  $s$  are both defined and equal”, we can express  $t\downarrow$  as  $t = t$ .

3.2. The following axiomatization is a convenient (but not canonical) choice for arguments proceeding by induction on the length of formal deductions:

- L1  $A \rightarrow A,$
- L2  $A, A \rightarrow B \Rightarrow B,$
- L3  $A \rightarrow B, B \rightarrow C \Rightarrow A \rightarrow C,$
- L4  $A \wedge B \rightarrow A, A \wedge B \rightarrow B,$
- L5  $A \rightarrow B, A \rightarrow C \Rightarrow A \rightarrow B \wedge C,$
- L6  $A \rightarrow A \vee B, B \rightarrow A \vee B,$
- L7  $A \rightarrow C, B \rightarrow C \Rightarrow A \vee B \rightarrow C,$
- L8  $A \wedge B \rightarrow C \Rightarrow A \rightarrow (B \rightarrow C),$
- L9  $A \rightarrow (B \rightarrow C) \Rightarrow A \wedge B \rightarrow C,$
- L10  $\perp \rightarrow A,$
- L11  $B \rightarrow A \Rightarrow B \rightarrow \forall x A \quad (x \notin \text{FV}(B)),$
- L12  $\forall x A \wedge t\downarrow \rightarrow A[x/t] \quad (t \text{ free for } x \text{ in } A),$
- L13  $A[x/t] \wedge t\downarrow \rightarrow \exists x A \quad (t \text{ free for } x \text{ in } A),$
- L14  $A \rightarrow B \Rightarrow \exists x A \rightarrow B \quad (x \notin \text{FV}(B))$

where  $t\downarrow := t = t$ . For equality we have ( $F$  function symbol,  $R$  relation symbol of the language):

$$\text{EQ} \quad \begin{cases} \forall xy(x = y \rightarrow y = x), & \forall xyz(x = y \wedge y = z \rightarrow x = z), \\ \forall \vec{x}\vec{y}(\vec{x} = \vec{y} \wedge F\vec{x}\downarrow \rightarrow F\vec{x} = F\vec{y}), & \forall \vec{x}\vec{y}(R\vec{x} \wedge \vec{x} = \vec{y} \rightarrow R\vec{y}) \end{cases}$$

Basic predicates and functions of the language are assumed to be strict:

STR  $F(t_1, \dots, t_n) \downarrow \rightarrow t_i \downarrow, \quad R(t_1, \dots, t_n) \rightarrow t_i \downarrow$

Note that this logic reduces to ordinary first-order intuitionistic logic if all functions are total, i.e.  $\forall \vec{x}(f\vec{x} \downarrow)$ , since then  $t \downarrow$  for all terms  $t$ .

For the notion “*equally defined and equal if defined*” introduced by

$$t \simeq s := (t \downarrow \vee s \downarrow) \rightarrow t = s,$$

we can prove the replacement schema for arbitrary formulas  $A$

$$t \simeq s \wedge A[x/t] \rightarrow A[x/s].$$

#### 4 CONSERVATIVENESS OF DEFINED FUNCTIONS

Relative to the logic of partial terms, the following conservative extension result is easily proved. Let  $\Gamma$  be a theory based on LPT, such that

$$\Gamma \vdash A(\vec{x}, y) \wedge A(\vec{x}, z) \rightarrow y = z.$$

Then we may introduce a symbol  $\phi_A$  for a partial function with axiom

$$\text{Ax}(\phi_A) \text{A}(\vec{x}, y) \leftrightarrow y = \phi_A(\vec{x}).$$

The conservativeness of this addition can be proved in a straightforward syntactic way; the easiest method, however, uses completeness for Kripke models, see Troelstra and van Dalen (1988, 2.7).

Let  $\Gamma^*$  consist of  $\Gamma$  and all substitution instances of the axiom schemata w.r.t. the extended language, and let  $\phi(\Gamma^*)$  be the result of systematically eliminating the function symbol  $\phi_A$  from the elements of  $\Gamma$ , and assume  $\phi(\Gamma^*)$  to be provable from  $\Gamma$ , then the conservative extension result still holds in the form: “ $\Gamma^* + \text{Ax}(\phi_A)$  is conservative over  $\Gamma$ ”.

This extended result applies to  $\mathbf{HA}^*$  defined below, since eliminating the symbol for partial recursive function application from instances of induction yields instances of induction in the language of  $\mathbf{HA}$ .

#### 5 FORMALIZING ELEMENTARY RECURSION THEORY IN $\mathbf{HA}^*$

**5.1.**  $\mathbf{HA}^*$  is the conservative extension of  $\mathbf{HA}$ , formulated in the intuitionistic logic of partial terms, with a primitive binary partial operation  $\bullet$  of partial recursive function application.  $t_1 \bullet t_2 \bullet t_3 \dots$  abbreviates  $(\dots((t_1 \bullet t_2) \bullet t_3) \dots)$  (association to the left).

Note that strictness entails in particular  $t \bullet t' \downarrow \rightarrow t \downarrow \wedge t' \downarrow$  for the application operation. Of course we have to require totality for the primitive recursive functions; it suffices to demand  $0 \downarrow, Sx \downarrow$ . In all other case the primitive recursive functions satisfy equations with  $=$ , characterizing them inductively in terms of functions introduced before (e.g.  $x + 0 = x, x + Sy = S(x + y)$ ). By induction one can then prove  $Fx_1 \dots x_n \downarrow$  for each primitive recursive function symbol  $F$ .



A formalization of elementary recursion theory in  $\mathbf{HA}^*$  can be given by using Kleene's index method in combination with the theory of elementary inductive definitions in arithmetic (Troelstra and van Dalen 1988, 3.6, 3.7). The idea behind this formalization is the following: one gives an elementary inductive definition of the relation  $\Omega := \{(n, x, m) : x \bullet m \simeq n\}$ . An elementary inductive definition of a predicate  $P_A$  is given by a predicate  $A(X, z)$  in the language of  $\mathbf{HA}^*$  extended with an extra predicate variable  $X$ , such that  $A$  is in a class  $\mathcal{P}$  generated by the following clauses:

- all arithmetical formulas are in  $\mathcal{P}$ ;
- $Xt \in \mathcal{P}$  for all numerical terms  $t$ ;
- $\mathcal{P}$  is closed under  $\wedge, \vee, \exists$  and bounded universal quantification  $\forall x < t$  with  $x \notin \text{FV}(t)$ .

The predicate  $P_A$  then satisfies

$$\forall x(A(P_A, x) \rightarrow P_A(x), \text{ and } \forall x(A(Q, x) \rightarrow Qx) \rightarrow \forall x(P_A(x) \rightarrow Qx),$$

for all predicates  $Q$  definable in  $\mathbf{HA}^*$  extended with  $P_A$ . Predicates introduced by elementary inductive definitions are in fact explicitly definable in arithmetic, and the principles for  $P_A$  stated above are provable in arithmetic.

This leads to a smooth formalization of elementary recursion theory; in particular we obtain the smn-theorem, the recursion theorem (Kleene's fixed-point theorem): for some primitive recursive  $\phi$

$$\forall \vec{x} \vec{y} z (\phi(z, \vec{x}) \bullet (\vec{y}) \simeq z \bullet (\vec{x}, \vec{y}))$$

(where  $(\vec{u})$  is some standard encoding of the sequence  $\vec{u}$ ), the Kleene normal form theorem, etc. Moreover, by the normal form theorem, every partial recursive function is definable by a term of the language of  $\mathbf{HA}^*$ .

**5.2. NOTATION.** If  $t$  is a term in the language of  $\mathbf{HA}^*$ , then  $\Lambda x.t$  is a canonically chosen code number for  $t$  as a partial recursive function of  $x$ , uniformly in the other free variables; by the smn-theorem we may therefore assume  $\Lambda x.t$  to be primitive recursive in  $\text{FV}(t) \setminus \{x\}$ .  $\Lambda x_1 \dots x_n.t$  abbreviates  $\Lambda x_1(\Lambda x_2 \dots (\Lambda x_n.t) \dots)$ .  $\square$

We note the following

**5.3. LEMMA.** *In  $\mathbf{HA}^*$  the  $\Sigma_1^0$ -formulas of  $\mathbf{HA}$  are equivalent to prime formulas of the form  $t = t$  for suitable  $t$ , and each formula  $t = s$  is equivalent to a  $\Sigma_1^0$ -formula of  $\mathbf{HA}$ .*

**PROOF.** Systematically using the equivalences mentioned above transforms any formula  $t = s$  of  $\mathbf{HA}^*$  into a  $\Sigma_1^0$ -formula of  $\mathbf{HA}$ . Conversely, let a  $\Sigma_1^0$ -formula be given; by the normal form results of recursion theory, we can write this in the form  $\exists z T(\bar{n}, \langle \vec{x} \rangle, z)$  for a numeral  $\bar{n}$ ; this is equivalent to  $\bar{n} \bullet \langle \vec{x} \rangle = \bar{n} \bullet \langle \vec{x} \rangle$ .  $\square$

We are now ready to formalize  $x \text{ r}\bar{n} A$  directly in  $\mathbf{HA}^*$ .

## 6 FORMALIZING $\underline{\text{rn}}$ -REALIZABILITY IN $\mathbf{HA}^*$

6.1. DEFINITION.  $x \underline{\text{rn}} A$  is defined by induction on the complexity of  $A$ ,  $x \notin \text{FV}(A)$ .

$$\begin{aligned} x \underline{\text{rn}} P &:= P \wedge x \downarrow \text{ for } P \text{ prime,} \\ x \underline{\text{rn}} (A \wedge B) &:= \mathbf{p}_0 x \underline{\text{rn}} A \wedge \mathbf{p}_1 x \underline{\text{rn}} B, \\ x \underline{\text{rn}} (A \rightarrow B) &:= \forall y (y \underline{\text{rn}} A \rightarrow x \bullet y \underline{\text{rn}} B) \wedge x \downarrow, \\ x \underline{\text{rn}} \forall y A &:= \forall y (x \bullet y \underline{\text{rn}} A), \\ x \underline{\text{rn}} \exists y A &:= \mathbf{p}_1 x \underline{\text{rn}} A[y/\mathbf{p}_0 x]. \end{aligned}$$

We also define a combination of realizability with truth,  $x \underline{\text{rnt}} A$ ; the clauses are the same as for  $\underline{\text{rn}}$ , the clause for implication excepted, which now reads:

$$x \underline{\text{rnt}} (A \rightarrow B) := \forall y (y \underline{\text{rnt}} A \rightarrow x \bullet y \underline{\text{rnt}} B) \wedge x \downarrow \wedge (A \rightarrow B). \quad \square$$

6.2. REMARKS. (i)  $t \underline{\text{rn}} A$  is  $\exists$ -free (i.e. does not contain  $\exists$ ) for all  $A$ . Note that, by our definition of  $\forall$  in terms of the other operators,  $\exists$ -free implies  $\forall$ -free.

(ii) The clauses “ $\wedge x \downarrow$ ” have been added for the cases of prime formulas and implications, in order to guarantee the truth of part (i) of the following lemma.

(iii) For negations we have  $x \underline{\text{rn}} \neg A \leftrightarrow \forall y (\neg y \underline{\text{rn}} A) \wedge x \downarrow$ , and  $x \underline{\text{rn}} \neg \neg A \leftrightarrow \forall y (\neg y \underline{\text{rn}} \neg A) \wedge x \downarrow \leftrightarrow \forall y \neg \forall z \neg (z \underline{\text{rn}} A) \wedge x \downarrow \leftrightarrow \neg \neg \exists z (z \underline{\text{rn}} A) \wedge x \downarrow$ .

The following lemmas are easily proved by induction on  $A$ .

6.3. LEMMA. (Definedness of realizing terms; Substitution Property) For  $\mathbf{R} \in \{\underline{\text{rn}}, \underline{\text{rnt}}\}$

$$(i) \vdash t \mathbf{R} A \rightarrow t \downarrow,$$

$$(ii) (x \mathbf{R} A)[y/t] \equiv x \mathbf{R} (A[y/t]) \quad (x \notin \text{FV}(A) \cup \text{FV}(t), y \neq x).$$

PROOF. By induction on the complexity of  $A$ . Let e.g.  $t \underline{\text{rn}} \exists y A$ , then  $\mathbf{p}_1 t \underline{\text{rn}} A[y/\mathbf{p}_0 t]$ , hence by induction hypothesis  $\mathbf{p}_1 t \downarrow$ , and so by strictness  $t \downarrow$ .  $\square$

6.4. LEMMA.  $\mathbf{HA}^* \vdash t \underline{\text{rnt}} A \rightarrow A$ .

A similar lemma holds for all combinations of realizability with truth (i.e. realizabilities with  $\underline{\text{t}}$  in their mnemonic code) we shall encounter in the sequel; we shall not bother to state it explicitly in the future. We can readily prove that realizability is sound for  $\mathbf{HA}^*$ :

## 7 SOUNDNESS

7.1. THEOREM. (Soundness theorem)

$$\mathbf{HA}^* \vdash A \Rightarrow \mathbf{HA}^* \vdash t \underline{\text{rn}} A \wedge t \underline{\text{rnt}} A$$

for a suitable term  $t$  with  $FV(t) \subset FV(A)$ .

PROOF. The proof proceeds by induction on the length of derivations; that is to say, we have to find realizing terms for the axioms, and for the rules we must show how to find a realizing term for the conclusion from realizing terms for the premises. We check some cases.

L5. Assume  $t \underline{rn} (A \rightarrow B)$ ,  $t' \underline{rn} (A \rightarrow C)$ , and let  $x \underline{rn} A$ ; then  $\mathbf{p}(t \bullet x, t' \bullet x) \underline{rn} (B \wedge C)$ , so  $\Lambda x. \mathbf{p}(t \bullet x, t' \bullet x) \underline{rn} (A \rightarrow B \wedge C)$ .

L14. Assume  $t \underline{rn} (A \rightarrow B)$ ,  $x \notin FV(B)$ , and let  $y \underline{rn} \exists x A$ , then  $\mathbf{p}_1 y \underline{rn} A[x/\mathbf{p}_0 y]$ , hence  $t[x/\mathbf{p}_0 y] \bullet (\mathbf{p}_1 y) \underline{rn} B$ , so  $\Lambda y. t[x/\mathbf{p}_0 y] \bullet (\mathbf{p}_1 y) \underline{rn} (\exists x A \rightarrow B)$ .

Of the non-logical axioms, only induction requires attention. Suppose

$$x \underline{rn} (A[y/0] \wedge \forall y (A \rightarrow A[y/Sy])).$$

Then

$$\mathbf{p}_0 x \underline{rn} A[y/0], \quad z \underline{rn} A \rightarrow (\mathbf{p}_1 x) \bullet y \bullet z \underline{rn} A[y/Sy].$$

So let  $t$  be such that

$$t \bullet 0 \simeq \mathbf{p}_0 x, \quad t \bullet (Sy) \simeq (\mathbf{p}_1 x) \bullet y \bullet (t \bullet y).$$

The existence of  $t$  follows either by an application of the recursion theorem, or is immediate if closure under recursion has been built directly into the definition of recursive function. It is now easy to prove by induction that  $t$  realizes induction for  $A$ .  $\square$

A statement weaker than soundness is  $\vdash A \Rightarrow \vdash \exists x (x \underline{rn} A)$ ; we might call this *weak soundness*. We can also prove a stronger version of soundness:

7.2. THEOREM. (*Strong Soundness Theorem*) For closed  $A$

$$\mathbf{HA}^* \vdash A \Rightarrow \mathbf{HA}^* \vdash \bar{n} \underline{rn} A \wedge \bar{n} \underline{rnt} A \quad \text{for some numeral } \bar{n}.$$

PROOF. Let  $\mathbf{HA}^* \vdash A$ ; from the soundness theorem we find a term  $t$  such that

$$t \underline{rn} A, \quad \text{hence } t \downarrow.$$

$t \downarrow$ , i.e.  $t = t$  is equivalent to a  $\Sigma_1^0$ -formula of  $\mathbf{HA}$ , say  $\exists x (s = 0)$ , and  $\mathbf{HA}$  proves only true  $\Sigma_1^0$ -formulas, from which we see that  $t = \bar{n}$  must be provable in  $\mathbf{HA}^*$  for some numeral  $\bar{n}$ . Similarly for  $\underline{rnt}$ .  $\square$

7.3. REMARK. If one formalizes the proof of the soundness theorem, it is easy to see that there are primitive recursive functions  $\psi, \phi$  such that

$$\mathbf{HA} \vdash \text{Prf}(x, \ulcorner A \urcorner) \rightarrow \text{Prf}(\phi(x), \text{Sub}(\ulcorner y \underline{rn} A \urcorner, y, \psi(x)))$$

where “Prf” is the formalized proof-predicate of  $\mathbf{HA}^*$ ,  $\ulcorner \xi \urcorner$  is the gödelnumber of expression  $\xi$ , and  $\text{Sub}(\ulcorner B \urcorner, x, \ulcorner s \urcorner)$  is the gödelnumber of  $B[x/s]$ .

In fact, the whole implication is provable even in primitive recursive arithmetic. But the statement expressing a formalized version of the *strong* completeness theorem:

$$\text{Prf}(x, \ulcorner A \urcorner) \rightarrow \text{Prf}(\phi(x), \overline{\ulcorner \psi(x) \urcorner} \underline{\text{rn}} A \urcorner)$$

( $A$  closed, for suitable provably recursive  $\phi, \psi$ ) is not provable in **HA** (see 10.6).

The following lemma will be used in the sequel, but is also interesting in its own right:

**7.4. LEMMA.** (Self-realizing formulas) For  $\exists$ -free formulas, canonical realizers exist, that is to say for each  $\exists$ -free  $A$  we have in **HA**\*

$$(i) \vdash \exists x(x \underline{\text{rn}} A) \rightarrow A,$$

$$(ii) \vdash A \rightarrow t_A \underline{\text{rn}} A \text{ for some term } t_A \text{ with } \text{FV}(t_A) \subset \text{FV}(A).$$

(iii) A formula  $A$  is provably equivalent to its own realizability, i.e.  $A \leftrightarrow \exists x(x \underline{\text{rn}} A)$ , iff  $A$  is provably equivalent to an existentially quantified  $\exists$ -free formula.

(iv) Realizability is idempotent, i.e.  $\exists x(x \underline{\text{rn}} \exists y(y \underline{\text{rn}} A)) \leftrightarrow \exists x(x \underline{\text{rn}} A)$ ; in fact, even  $\exists x(x \underline{\text{rn}} (A \leftrightarrow \exists y(y \underline{\text{rn}} A)))$  holds.

**PROOF.** Take  $t_{s=s'} := 0$ ,  $t_{A \wedge B} := \mathbf{p}(t_A, t_B)$ ,  $t_{\forall x.A} := \Lambda x.t_A$ ,  $t_{A \rightarrow B} := \Lambda x.t_B$  ( $x \notin \text{FV}(t_B)$ ), and prove (i) and (ii) by simultaneous induction on  $A$ . (iii) and (iv) are immediate corollaries.  $\square$

**7.5. REMARK.** An observation of practical usefulness is the following. For any definable predicate with canonical realizers (i.e. a predicate  $A$  definable by an  $\exists$ -free formula) we obtain an equivalent realizability if we read restricted quantifiers  $\forall x(A(x) \rightarrow \dots)$  and  $\exists x(A(x) \wedge \dots)$  as quantifiers  $\forall x \in A$ ,  $\exists x \in A$  over a new domain with realizability clauses copied from numerical quantification, i.e.

$$\begin{aligned} x \underline{\text{rn}} \forall y \in A. B &:= \forall y \in A(x \bullet y \underline{\text{rn}} B) \wedge x \downarrow, \\ x \underline{\text{rn}} \exists y \in A. B &:= \mathbf{p}_1 x \underline{\text{rn}} B[x/\mathbf{p}_0 x] \wedge A(\mathbf{p}_0 x). \end{aligned}$$

In short, we may simply forget about the canonical realizers.

## 8 AXIOMATIZING PROVABLE REALIZABILITY

**8.1.** As we have seen already in the introduction, realizability validates more than what is provable in **HA**; in fact, we can formally prove realizability of in **HA**\* an intuitionistic version of Church's thesis:

$$\text{CT}_0 \quad \forall x \exists y A(x, y) \rightarrow \exists z \forall x (A(x, z \bullet x) \wedge z \bullet x \downarrow).$$

$\text{CT}_0$  is certainly not *provable in HA*, since it is in fact refutable in classical arithmetic. This version of Church's thesis is in fact a combination of the well-known version which states "Each humanly computable function is recursive" and the intuitionistic reading of  $\forall x \exists y A(x, y)$  which states that there is a method

for constructing, for each given  $x$ , a  $y$  such that  $A(x, y)$ . Such a method describes a humanly computable function.

We now ask ourselves: is there a reasonably simple axiomatization (by a few axiom schemata say) of the formulas provably realizable in **HA**? The answer is yes, the provably realizable formulas can be axiomatized by a generalization of  $\text{ECT}_0$ , namely “*Extended Church’s Thesis*”:

$$\text{ECT}_0 \forall x(Ax \rightarrow \exists y Bxy) \rightarrow \exists z \forall x(Ax \rightarrow z \bullet x \downarrow \wedge B(x, z \bullet x)) \quad (A \text{ } \exists\text{-free}).$$

**8.2. LEMMA.** *Each instance of  $\text{ECT}_0$  is  $\mathbf{HA}^*$ -realizable.*

**PROOF.** Suppose

$$u \underline{\mathbf{rn}} \forall x(Ax \rightarrow \exists y Bxy)$$

Then  $\forall xv(v \underline{\mathbf{rn}} Ax \rightarrow u \bullet x \bullet v \underline{\mathbf{rn}} \exists y Bxy)$ , and since  $A$  is  $\exists$ -free, in particular  $\forall x(Ax \rightarrow u \bullet x \bullet t_A \underline{\mathbf{rn}} \exists y Bxy)$ , so  $\forall x(Ax \rightarrow \mathbf{p}_1(u \bullet x \bullet t_A) \underline{\mathbf{rn}} B(x, \mathbf{p}_0(u \bullet x \bullet t_A)))$ . Then it is straightforward to see that

$$\mathbf{p}(\Lambda x. \mathbf{p}_0(u \bullet x \bullet t_A), \Lambda xv. \mathbf{p}(0, \mathbf{p}_1(u \bullet x \bullet t_A)))$$

realizes the conclusion.  $\square$

**REMARK.** The condition “ $A$  is  $\exists$ -free” in  $\text{ECT}_0$  cannot be dropped: applying unrestricted  $\text{ECT}_0$  to  $Ax := \exists z Txxz \vee \neg \exists z Txxz$ ,  $Bxy := (y = 0 \wedge \exists z Txxz) \vee (y = 1 \wedge \neg \exists z Txxz)$  yields a contradiction. In fact, this example can be used to show that even unrestricted  $\text{ECT}_0!$  fails ( $\text{ECT}_0!$  is like  $\text{ECT}_0$  except that  $\exists y$  in the premise is replaced by  $\exists!y$ ;  $\exists!y$  means “there is a unique  $y$  such that”).

**8.3. THEOREM.** (*Characterization Theorem for  $\underline{\mathbf{rn}}$ -realizability*)

$$(i) \quad \mathbf{HA}^* + \text{ECT}_0 \vdash A \leftrightarrow \exists x(x \mathbf{R} A) \text{ for } \mathbf{R} \in \{\underline{\mathbf{rn}}, \underline{\mathbf{rnt}}\},$$

$$(ii) \quad \text{For closed } A, \mathbf{HA}^* + \text{ECT}_0 \vdash A \leftrightarrow \mathbf{HA}^* \vdash \bar{n} \underline{\mathbf{rn}} A \text{ for some numeral } \bar{n}.$$

**PROOF.** (i) is proved by a straightforward induction on  $A$ . The crucial case is  $A \equiv B \rightarrow C$ ; then  $B \rightarrow C \leftrightarrow (\exists x(x \underline{\mathbf{rn}} B) \rightarrow \exists y(y \underline{\mathbf{rn}} C))$  (by the induction hypothesis)  $\leftrightarrow \forall x(x \underline{\mathbf{rn}} B \rightarrow \exists y(y \underline{\mathbf{rn}} C))$  (by pure logic)  $\leftrightarrow \exists z \forall x(x \underline{\mathbf{rn}} B \rightarrow z \bullet x \underline{\mathbf{rn}} C)$  (by  $\text{ECT}_0$ , since  $x \underline{\mathbf{rn}} B$  is  $\exists$ -free)  $\equiv \exists z(z \underline{\mathbf{rn}} (B \rightarrow C))$ .

(ii). The direction  $\Rightarrow$  follows from the strong soundness theorem plus the lemma;  $\Leftarrow$  is an immediate consequence of (i).  $\square$

Curiosity prompts us to ask which formulas are classically provably realizable, i.e. provably realizable in first-order Peano Arithmetic **PA**, which is just **HA** with classical logic. The answer is contained in the following

8.4. PROPOSITION.  $\mathbf{PA} \vdash \exists x(x \underline{\text{rn}} A) \Leftrightarrow \mathbf{HA} + \text{M} + \text{ECT}_0 \vdash \neg\neg A$ ,  
 where M is Markov's principle:

$$\text{M} \quad \forall x(A \vee \neg A) \wedge \neg\neg\exists x A \rightarrow \exists x A.$$

PROOF. Let  $\mathbf{PA} \vdash \exists x(x \underline{\text{rn}} A)$ , and let  $B$  be a negative formula (i.e. a formula in the  $\wedge, \forall, \rightarrow$ -fragment) such that  $\mathbf{HA} + \text{M} \vdash x \underline{\text{rn}} A \leftrightarrow B(x)$ . Then  $\mathbf{PA} \vdash \neg\forall x\neg(x \underline{\text{rn}} A)$ , and since  $\mathbf{PA}$  is conservative over  $\mathbf{HA}$  for negative formulas (in consequence of Gödel's negative translation), also  $\mathbf{HA} \vdash \neg\forall x\neg B$ , i.e.  $\mathbf{HA} + \text{M} \vdash \neg\neg\exists x(x \underline{\text{rn}} A)$ , and thus it follows that  $\mathbf{HA} + \text{M} + \text{ECT}_0 \vdash \neg\neg A$ . The converse is simpler.  $\square$

## 9 EXTENSIONS OF $\mathbf{HA}^*$

9.1. For suitable sets  $\Gamma$  of extra axioms, we may replace  $\mathbf{HA}^*$  in the soundness and characterization theorem by  $\mathbf{HA}^* + \Gamma$ . Weak soundness and the characterization theorem require for all  $A \in \Gamma$

$$(1) \quad \mathbf{HA}^* + \Gamma \vdash \exists x(x \underline{\text{rn}} A).$$

Soundness requires for all  $A \in \Gamma$

$$(2) \quad \mathbf{HA}^* + \Gamma \vdash t \underline{\text{rn}} A \text{ for some term } t,$$

and Strong Soundness requires (2) and in addition:  $\mathbf{HA}^* + \Gamma$  proves only true  $\Sigma_1^0$ -formulas.

## 9.2. EXAMPLES

(a) For  $\Gamma$  any set of  $\exists$ -free formulas soundness and the characterization theorem extend. If  $\mathbf{HA}^* + \Gamma$  proves only true  $\Sigma_1^0$ -formulas, strong soundness holds. The next two examples permit characterization and strong soundness.

(b) Let  $\prec$  be a primitive recursive well-ordering of  $\mathbb{N}$ , provably total and linear in  $\mathbf{HA}^*$ ; for  $\Gamma$  we take all instances of *transfinite induction over*  $\prec$ :

$$\text{TI}(\prec) \forall y(\forall x \prec y A \rightarrow A[x/y]) \rightarrow \forall x A.$$

(c)  $\Gamma$  is the set of instances of Markov's principle (cf. the last proposition in 8). In fact, in the presence of  $\text{CT}_0$ , which is valid under realizability,  $\Gamma$  may be replaced by a single axiom:

$$\forall xy(\neg\neg\exists zTxyz \rightarrow \exists zTxyz).$$

It is also worth noting that in the presence of M, we can use the following variant of  $\text{ECT}_0$  which is equivalent to  $\text{ECT}_0$ :

$$\text{ECT}'_0 \forall x(\neg A \rightarrow \exists yBxy) \rightarrow \exists z\forall x(\neg A \rightarrow z \bullet x \downarrow \wedge B(x, z \bullet y)).$$

(d) An extension of another kind is obtained if we enrich the language with constants for inductively defined predicates, e.g. the tree predicate  $\text{Tr}$ . Intuitively,  $\text{Tr}$  is the least set containing the (code of the) single-node tree (i.e.  $\langle \rangle \in \text{Tr}$ ), and with every recursive sequence of tree codes  $n \bullet 0, n \bullet 1, \dots, n \bullet m, \dots$  in  $\text{Tr}$ ,  $\text{Tr}$  also contains a code for the infinite tree having the trees with codes  $n \bullet m$  as immediate subtrees, namely  $\mathbf{p}(1, n)$ . Thus if

$$A(X, x) := (x = 0) \vee (\mathbf{p}_0 x = 1 \wedge \forall m (\mathbf{p}_1 x \bullet m \in X))$$

we have

$$\begin{aligned} A(\text{Tr}, x) &\rightarrow x \in \text{Tr}, \\ \forall x (A(\lambda y. B, x) \rightarrow B[y/x]) &\rightarrow \forall x \in \text{Tr}. B[y/x] \end{aligned}$$

for all  $B$  in the language extended with the new primitive predicate  $\text{Tr}$ . Then we can extend  $\underline{\text{rn}}$ -realizability simply by putting

$$x \underline{\text{rn}} (t \in \text{Tr}) := t \in \text{Tr}.$$

Let us check that the soundness theorem extends.  $A(\text{Tr}, x)$  is equivalent to an  $\exists$ -free formula, so its realizability implies its truth, and  $x \in \text{Tr}$  follows. As to the schema, assume

$$\begin{aligned} u \underline{\text{rn}} \forall x (A(\lambda y. B, x) \rightarrow B[y/x]), \text{ or} \\ u \underline{\text{rn}} \forall x [(x = 0 \rightarrow B(0)) \wedge (\mathbf{p}_0 x = 1 \wedge \forall y B(\mathbf{p}_1 x \bullet y) \rightarrow Bx)]. \end{aligned}$$

So

$$\begin{aligned} \mathbf{p}_0(u \bullet 0) \bullet (0, 0) \underline{\text{rn}} B(0), \\ \mathbf{p}_1(u \bullet x) \bullet v \underline{\text{rn}} B(x) \text{ if } \mathbf{p}_0 x = 1 \text{ and } v \underline{\text{rn}} (\mathbf{p}_0 x = 1 \wedge \forall y B(\mathbf{p}_1 x \bullet y)). \end{aligned}$$

Assume  $\forall y (e \bullet (\mathbf{p}_1 x \bullet y) \underline{\text{rn}} B(\mathbf{p}_1 x \bullet y))$ ,  $\mathbf{p}_0 x = 1$ . Then

$$v = \mathbf{p}(0, \Lambda y. e \bullet (\mathbf{p}_1 x \bullet y)) \underline{\text{rn}} (\mathbf{p}_0 x = 1 \wedge \forall y B(\mathbf{p}_1 x \bullet y)).$$

Therefore

$$\begin{aligned} \text{if } \mathbf{p}_0 x = 1 \text{ and } \forall y (e \bullet (\mathbf{p}_1 x \bullet y) \underline{\text{rn}} B(\mathbf{p}_1 x \bullet y)) \\ \text{then } \mathbf{p}_1(u \bullet x) \bullet (0, \Lambda y. e \bullet (\mathbf{p}_1 x \bullet y)) \underline{\text{rn}} B(x). \end{aligned}$$

Now we construct by the recursion theorem an  $e$  such that

$$e \bullet x \simeq \begin{cases} \mathbf{p}_0(u \bullet 0) \bullet 0 & \text{if } x = 0, \\ \mathbf{p}_1(u \bullet x) \bullet \mathbf{p}(0, \Lambda y. e \bullet (\mathbf{p}_1 x \bullet y)) & \text{if } \mathbf{p}_0 x = 1, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

We then prove by induction on  $\text{Tr}$  that  $\forall x \in \text{Tr} (e \bullet x \underline{\text{rn}} B(x))$ . This is straightforward. This example is capable of considerable generalization, namely to arithmetic enriched with constants for predicates introduced by iterated inductive definitions of higher level; see e.g. Buchholz, Feferman, Pohlers and Sieg (1981, IV, section 6).

The examples just mentioned also permit extension of  $\underline{\text{rnt}}$ -realizability. We end the section with some applications of  $\underline{\text{rn}}$ - and  $\underline{\text{rnt}}$ -realizability.

## 10 APPLICATIONS

### 10.1. PROPOSITION. (Consistency and inconsistency results)

- (i)  $\mathbf{HA}^* + \mathbf{ECT}_0$  is consistent relative to  $\mathbf{HA}^*$  (and hence also relative to  $\mathbf{PA}$ ).
- (ii)  $\neg\forall x(A \vee \neg A), \neg(\forall x\neg\neg B \rightarrow \neg\neg\forall xB)$  are consistent with  $\mathbf{HA}^*$  for certain arithmetical  $A, B$ .
- (iii) The schema “Independence of Premise”

$$\text{IP} \quad (\neg A \rightarrow \exists zB) \rightarrow \exists z(\neg A \rightarrow B)$$

is not derivable in  $\mathbf{HA}^* + \mathbf{CT}_0 + \mathbf{M}$ ; in fact,  $\mathbf{HA}^* + \text{IP} + \mathbf{CT}_0 + \mathbf{M} \vdash 1 = 0$ .

PROOF. (i) Immediate from the characterization theorem.

(ii) is a corollary of the realizability of  $\mathbf{CT}_0$ : take  $A \equiv \exists yTxxxy$ ,  $B \equiv \exists yTxxxy \vee \neg\exists yTxxxy$ .

(iii) By  $\mathbf{M}$ ,  $\neg\neg\exists yTxxxy \rightarrow \exists zTxxz$ ; apply  $\text{IP}$  to obtain  $\forall x\exists z(\neg\neg\exists yTxxxy \rightarrow Txxz)$ , then by  $\mathbf{CT}_0$  there is a total recursive  $F$  such that  $\neg\neg\exists yTxxxy \rightarrow T(x, x, Fx)$ , and this would make  $\exists yTxxxy$  recursive in  $x$ .  $\square$

We next give an example of a conservative extension result.

10.2. DEFINITION.  $\text{CC}(\underline{\mathbf{rn}})$  (the  $\underline{\mathbf{rn}}$ -Conservative Class) is the class of formulas  $A$  such that whenever  $B \rightarrow C$  is a subformula of  $A$ , then  $B$  is  $\exists$ -free.  $\square$

10.3. LEMMA. For  $A \in \text{CC}(\underline{\mathbf{rn}})$  we have  $\vdash \exists x(x \underline{\mathbf{rn}} A) \rightarrow A$ .

PROOF. By induction on the structure of  $A$ . Consider the case  $A \equiv B \rightarrow C$ ; then  $B$  is  $\exists$ -free, so there is a  $t_B$  such that  $\vdash B \rightarrow t_B \underline{\mathbf{rn}} B$ . Assume  $B$  and  $x \underline{\mathbf{rn}} (B \rightarrow C)$ , then  $x \bullet t_B \downarrow \wedge x \bullet t_B \underline{\mathbf{rn}} C$ , hence by the induction hypothesis  $C$ ; therefore  $(x \underline{\mathbf{rn}} (B \rightarrow C)) \rightarrow (B \rightarrow C)$ .  $\square$

The lemma in combination with the characterization theorem yields

10.4. PROPOSITION.  $\mathbf{HA}^* + \mathbf{ECT}_0$  is conservative over  $\mathbf{HA}^*$  w.r.t. formulas in  $\text{CC}(\underline{\mathbf{rn}})$ :

$$(\mathbf{HA}^* + \mathbf{ECT}_0) \cap \text{CC}(\underline{\mathbf{rn}}) = \mathbf{HA}^* \cap \text{CC}(\underline{\mathbf{rn}}).$$

The following proposition follows from  $\underline{\mathbf{rnt}}$ -realizability.

10.5. PROPOSITION. (Derived rules) In  $\mathbf{HA}^*$

- (i) For sentences  $\vdash A \vee B \Rightarrow \vdash A$  or  $\vdash B$  (Disjunction property DP),
- (ii) For sentences  $\vdash \exists xA \Rightarrow \vdash A[x/\bar{n}]$  for some numeral  $\bar{n}$  (Explicit Definability for Numbers EDN),
- (iii) Extended Church’s Rule: for  $\exists$ -free  $A$



$$\text{ECR} \vdash \forall x(A \rightarrow \exists yBxy) \Rightarrow \vdash \exists z\forall x(A \rightarrow z\bullet x \downarrow \wedge B(x, z\bullet x)).$$

PROOF. (i) follows from (ii) (actually, (i) and (ii) are equivalent for systems containing a minimum of arithmetic, see Friedman (1975)). As to (ii), let  $\vdash \exists xA$ , then, by the strong soundness for  $\underline{\text{rnt}}$ -realizability,  $\vdash \bar{m} \underline{\text{rnt}} \exists xA$  for some numeral  $\bar{m}$ , so  $\vdash \mathbf{p}_1 \bar{m} \underline{\text{rnt}} A[x/\mathbf{p}_0 \bar{m}]$ , and hence  $\vdash A[x/\mathbf{p}_0 \bar{m}]$ .

(iii) Assume  $\vdash \forall x(A \rightarrow \exists yBxy)$ , then for a suitable  $t \vdash t \underline{\text{rnt}} \forall x(A \rightarrow \exists yBxy)$ , i.e.

$$\vdash \forall x\forall z(z \underline{\text{rnt}} A \rightarrow \mathbf{p}_1(t\bullet x\bullet z) \underline{\text{rnt}} B(x, \mathbf{p}_0(t\bullet x\bullet z))).$$

Since  $t_A \underline{\text{rnt}} A$ ,

$$\vdash \forall x(A \rightarrow \mathbf{p}_1(t\bullet x\bullet t_A) \underline{\text{rnt}} B(x, \mathbf{p}_0(t\bullet x\bullet t_A))),$$

and therefore  $\vdash \forall x(A \rightarrow B(x, \mathbf{p}_0(t\bullet x\bullet t_A)))$ . So we can take  $z = \Lambda x.\mathbf{p}_0(t\bullet x\bullet t_A)$ .  $\square$

10.6. REMARK. The DP cannot be formalized in any consistent extension of  $\mathbf{HA}$  itself (Myhill (1973), Friedman (1977)). We sketch Myhill's argument (the result of Friedman is even stronger). Assume that there is a provably recursive function  $f$  satisfying

$$\vdash \text{Prf}(x, \ulcorner A \vee B \urcorner) \rightarrow ((fx = 0 \wedge \text{Pr}(\ulcorner A \urcorner)) \vee ((fx = 1 \wedge \text{Pr}(\ulcorner B \urcorner))).$$

where  $\text{Pr}(x) := \exists y\text{Prf}(y, x)$ . So  $f = \{\bar{p}\}$ , and  $\vdash \forall x\exists yT\bar{p}xy$ . Let  $F$  enumerate all primitive recursive functions, i.e.  $\lambda n.F(i, n)$  is the  $i$ -th primitive recursive function. Put

$$D(n) := \bar{p}\bullet F(n, n) \neq 0,$$

then  $\vdash \forall n(Dn \vee \neg Dn)$  (i.e.  $\text{Prf}(\bar{k}, \ulcorner \forall n(Dn \vee \neg Dn) \urcorner)$  for a specific  $\bar{k}$ ), from which we can find a particular primitive recursive  $\lambda n.F(\bar{m}, n)$  such that  $\vdash \text{Prf}(F(\bar{m}, \bar{n}), \ulcorner D\bar{n} \vee \neg D\bar{n} \urcorner)$ . Then  $D\bar{m} \rightarrow \bar{p}\bullet F(\bar{m}, \bar{m}) \neq 0 \rightarrow \text{Prf}(F(\bar{m}, \bar{m}), \ulcorner D\bar{m} \vee \neg D\bar{m} \urcorner) \wedge \text{Pr}(\ulcorner \neg D\bar{m} \urcorner)$ , hence  $\neg D\bar{m}$  follows, since  $\mathbf{HA}^*$  is consistent. If we start assuming  $\neg D\bar{m}$ , we similarly obtain a contradiction.

From this we see that DP cannot be proved in  $\mathbf{HA}^*$  itself; for if DP were provable in  $\mathbf{HA}^*$ , then a function  $f$  as above would be given by

$$\begin{aligned} f(x) &:= \mathbf{p}_0(\text{the least } y \text{ s.t. } (x \text{ does not prove a closed disjunction and } y = 0) \\ &\text{or (for some closed } \ulcorner A \vee B \urcorner, \text{Prf}(x, \ulcorner A \vee B \urcorner) \wedge \mathbf{p}_0 y = 0 \wedge \text{Prf}(\mathbf{p}_1 y, \ulcorner A \urcorner)) \\ &\text{or (for some closed } \ulcorner A \vee B \urcorner, \text{Prf}(x, \ulcorner A \vee B \urcorner) \wedge \mathbf{p}_1 y = 1 \wedge \text{Prf}(\mathbf{p}_1 y, \ulcorner B \urcorner))). \end{aligned}$$

This in turn implies that the strong soundness theorem is not formalizable in  $\mathbf{HA}^*$ , since strong soundness for  $\underline{\text{rnt}}$ -realizability immediately implies EDN for  $\mathbf{HA}^* + \text{ECT}_0$ .

#### REFERENCES

- Beeson, M. (1981). Formalizing constructive mathematics: Why and how?, in F. Richman (ed.), *Constructive mathematics*, Springer Verlag, Berlin, Heidelberg, New York, pp. 146–190.
- Buchholz, W., Feferman, S., Pohlers, W. and Sieg, W. (1981). *Iterated Inductive Definitions and Subsystems of Analysis: Recent Proof-Theoretical Studies*, Springer Verlag, Berlin, Heidelberg, New York.
- Dragalin, A. G. (1988). *Mathematical Intuitionism*, American Mathematical Society, Providence, Rhode Island. Translation of the Russian original from 1979.
- Friedman, H. M. (1975). The disjunction property implies the numerical existence property, *Proceedings of the National Academy of Sciences of the United States of America* **72**: 2877–2878.
- Friedman, H. M. (1977). On the derivability of instantiation properties, *JSL* **42**: 506–514.
- Kleene, S. C. (1945). On the interpretation of intuitionistic number theory, *JSL* **10**: 109–124.
- Kleene, S. C. (1969). *Formalized recursive functionals and formalized realizability*, Vol. 89 of *Memoirs of the American Mathematical Society*, American Mathematical Society, Providence, Rhode Island.
- Myhill, J. R. (1973). A note on indicator-functions, *Proceedings of the American Mathematical Society* **29**: 181–183.
- Troelstra, A. S. and van Dalen, D. (1988). *Constructivism in Mathematics*, North-Holland, Amsterdam. 2 volumes.
- Troelstra, A. S. (ed.) (1973). *Metamathematical investigation of intuitionistic arithmetic and analysis*, Springer Verlag, Berlin, Heidelberg, New York. With contributions by A. S. Troelstra, C. A. Smorynski, J. I. Zucker and W. A. Howard.

# Verification of a Distributed Summation Algorithm

Frits W. Vaandrager

CWI

P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

fritsv@cwi.nl

University of Amsterdam

Programming Research Group

Kruislaan 403, 1098 SJ Amsterdam, The Netherlands

## 1 INTRODUCTION

Reasoning about distributed algorithms appears to be intrinsically difficult and will probably always require a great deal of ingenuity. Nevertheless, research on formal verification has provided a whole range of well-established concepts and techniques that may help us to tackle problems in this area. It seems that by now the basic principles for reasoning about distributed algorithms have been discovered and that the main issue that remains is the problem of scale: we know how to analyze small algorithms but are still lacking methods and tools to manage the complexity of the the bigger ones (in this context we can take “small” to mean “fits on one or two pages”).

Not everybody agrees with this view, however, and frequently one can hear claims that existing approaches cannot deal (or cannot deal in a natural way) with certain types of distributed algorithms. A new approach is then proposed to address this problem. A recent example of this is a paper by Chou [3], who offers a rather pessimistic view on the state-of-the-art in formal verification:

At present, reasoning about distributed algorithms is still an *ad hoc*, trial-and-error process that needs a great deal of ingenuity. What is lacking is a practical method that supports, on the one hand, an *intuitive* way to think about and understand distributed algorithms and, on the other hand, a *formal* technique for reasoning about distributed algorithms using that intuitive understanding.

To illustrate the shortcoming of the assertional methods of [2, 5, 6, 7, 8, 10, 13], Chou discusses a variant of Segall’s PIF (Propagation of Information with Feedback) protocol [18]. A complex and messy classical proof of this algorithm is contrasted with a slightly simpler but definitely more structured proof based on the new method advocated by the author.

I think that Chou’s view of existing assertional methods is much too pessimistic. First of all these methods are not ad-hoc, but provide significant guidance and structure to verifications. After one has described both the algorithm and its specification as abstract programs, it is usually not so difficult

to come up with a first guess of a simulation relation from the state space of the algorithm to the state space of the specification. In order to state this simulation it is sometimes necessary to add auxiliary history and prophecy variables to the low-level program. By just starting to prove that the guessed simulation relation is indeed a simulation, i.e., that for each execution of the low-level program there exists a corresponding execution of the high-level program, one discovers the need for certain invariants, properties that are valid for all reachable states of the programs. To prove these invariant properties it is sometimes convenient or even necessary to introduce auxiliary state variables. Frequently one also has to prove other auxiliary invariants first. The existence of a simulation relation guarantees that the algorithm is safe with respect to the specification: all the finite behaviors of the algorithm are allowed by the specification. The concepts of invariants, history and prophecy variables, and simulation relations are so powerful that in most cases they allow one to formalize the intuitive reasoning about safety properties of distributed algorithms. When a simulation relation (and thereby the safety properties) has been established, this relation often provides guidance in the subsequent proof that the algorithm satisfies the required liveness properties: typically one proves that the simulation relates each fair execution of the low-level program to a fair execution of the high-level program. Here modalities from temporal logic such as “eventually” and “leads to” often make it quite easy to formalize intuitions about the liveness properties of the algorithm.

As an illustration of the use of “classical” assertional methods, I present in this paper a verification of the algorithm discussed by Chou [3]. Altogether, it took me about two hours to come up with a detailed sketch of the proof (during a train ride from Leiden to Eindhoven), and less than two weeks to work it out and write this paper. The proof is completely routine, except for a few nice invariants and the idea to use a prophecy variable. Unlike history variables, which date back to the sixties [9], prophecy variables have been introduced only recently [1], and there are not that many examples of their use. My proof is not particularly short, but it does formalize in a direct way my own intuitions about the behavior of this algorithm.

It might very well be the case that for more complex distributed algorithms, such as [17], new methods will pay off and lead to shorter proofs that are closer to intuition. This paper shows that, unlike what is claimed by Chou [3], the old methods still work very well for a variant of Segall’s PIF protocol.

## 2 LABELED TRANSITION SYSTEMS AND SIMULATIONS

In this paper we use a very simple and well-known transition system model. The model is a simplified version of the I/O automata model [10, 11]: it does not deal with fairness or other forms of liveness and there is no distinction between input and output actions. In this section we review some basic definitions and results concerning automata and simulation proof techniques. For a more extensive introduction we refer to [12].

DEFINITION 1 A *labeled transition system* or *automaton*  $A$  consists of four components:

- A (finite or infinite) set  $states(A)$  of states.
- A nonempty set  $start(A) \subseteq states(A)$  of start states.
- A pair  $(ext(A), int(A))$  of disjoint sets of external and internal actions, respectively. The derived set  $acts(A)$  of actions is defined as the union of  $ext(A)$  and  $int(A)$ .
- A set  $steps(A) \subseteq states(A) \times acts(A) \times states(A)$  of steps.

We let  $s, s', u, u', \dots$  range over states, and  $a, \dots$  over actions. We write  $s \xrightarrow{a}_A s'$ , or just  $s \xrightarrow{a} s'$  if  $A$  is clear from the context, as a shorthand for  $(s', a, s) \in steps(A)$ .

An *execution fragment* of an automaton  $A$  is a finite or infinite alternating sequence,  $\alpha = s_0 a_1 s_1 a_2 s_2 \dots$ , of states and actions of  $A$ , beginning with a state, and if it is finite also ending with a state, such that for all  $i$ ,  $s_i \xrightarrow{a_{i+1}} s_{i+1}$ . The function *first* gives the first state of an execution fragment and, for finite execution fragments, the function *last* gives the final state. An *execution* of  $A$  is an execution fragment that begins with a start state. A state  $s$  of  $A$  is *reachable* if  $s = last(\alpha)$  for some finite execution  $\alpha$  of  $A$ .

The *trace* of an execution fragment  $\alpha$ , written  $trace(\alpha)$ , is the sequence of external actions occurring in  $\alpha$ . A sequence  $\beta$  of actions is a *trace* of automaton  $A$  if there is an execution  $\alpha$  of  $A$  with  $\beta = trace(\alpha)$ . The set of traces of  $A$  is denoted by  $traces(A)$ . Suppose  $s$  and  $s'$  are states of  $A$ , and  $\beta$  is a finite sequence of external actions of  $A$ . We write  $s \xrightarrow{\beta}_A s'$ , or just  $s' \xrightarrow{\beta} s$ , if  $A$  has a finite execution fragment  $\alpha$  with  $first(\alpha) = s$ ,  $trace(\alpha) = \beta$  and  $last(\alpha) = s'$ .

DEFINITION 2 Let  $A$  and  $B$  be automata with the same external actions.

1. A *refinement* from  $A$  to  $B$  is a function  $r$  from states of  $A$  to states of  $B$  that satisfies the following two conditions:
  - (a) If  $s$  is a start state of  $A$  then  $r(s)$  is a start state of  $B$ .
  - (b) If  $s \xrightarrow{a}_A s'$  and both  $s$  and  $r(s)$  are reachable, then  $r(s) \xrightarrow{\beta}_B r(s')$ , where  $\beta = trace((s, a, s'))$ .
2. A *forward simulation* from  $A$  to  $B$  is a relation between states of  $A$  and states of  $B$  that satisfies the following two conditions:
  - (a) If  $s$  is a start state of  $A$  then there exists a start state  $u$  of  $B$  with  $(s, u) \in f$ .
  - (b) If  $s \xrightarrow{a}_A s'$ ,  $(s, u) \in f$  and  $s$  and  $u$  are reachable, then there exists a state  $u'$  of  $B$  such that  $u \xrightarrow{\beta}_B u'$  and  $(s', u') \in f$ , where  $\beta = trace((s, a, s'))$ .

3. A *history relation* from  $A$  to  $B$  is a forward simulation from  $A$  to  $B$  whose inverse is a refinement from  $B$  to  $A$ .
4. A *backward simulation* from  $A$  to  $B$  is a relation between states of  $A$  and states of  $B$  that satisfies the following three conditions:
  - (a) If  $s$  is a start state of  $A$  and  $u$  is a reachable state of  $B$  with  $(s, u) \in b$ , then  $u$  is a start state of  $B$ .
  - (b) If  $s \xrightarrow{a}_A s'$ ,  $(s', u') \in b$  and  $s$  and  $u'$  are reachable, then there exists a reachable state  $u$  of  $B$  such that  $u \xrightarrow{\beta}_B u'$  and  $(s, u) \in b$ , where  $\beta = \text{trace}((s, a, s'))$ .
  - (c) If  $s$  is a reachable state of  $A$  then there exists a reachable state  $u$  of  $B$  with  $(s, u) \in b$ .
5. A *prophecy relation* from  $A$  to  $B$  is a backward simulation from  $A$  to  $B$  whose inverse is a refinement from  $B$  to  $A$ .

A refinement, forward simulation, etc. is called *strong* if in each case where one automaton is required to simulate a step from the other automaton, this is possible with an execution fragment consisting of *exactly* one step.<sup>1</sup>

A relation  $R$  over  $S_1$  and  $S_2$  is *image-finite* if for all elements  $s_1$  of  $S_1$  there are only finitely many elements  $s_2$  of  $S_2$  such that  $(s_1, s_2) \in R$ .

**THEOREM 1** *Let  $A$  and  $B$  be automata with the same external actions.*

1. *If there is a refinement from  $A$  to  $B$  then  $\text{traces}(A) \subseteq \text{traces}(B)$ .*
2. *If there is a forward simulation from  $A$  to  $B$  then  $\text{traces}(A) \subseteq \text{traces}(B)$ .*
3. *If there is a history relation from  $A$  to  $B$  then  $\text{traces}(A) = \text{traces}(B)$ .*
4. *If there is an image-finite backward simulation from  $A$  to  $B$  then  $\text{traces}(A) \subseteq \text{traces}(B)$ .*
5. *If there is an image-finite prophecy relation from  $A$  to  $B$  then  $\text{traces}(A) = \text{traces}(B)$ .*

### 3 DESCRIPTION OF THE ALGORITHM

Consider a graph  $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ , where  $\mathbf{V}$  is a nonempty, finite collection of nodes and  $\mathbf{E} \subseteq \mathbf{V} \times \mathbf{V}$  is a collection of links. We assume that graph  $\mathbf{G}$  is undirected, i.e.,  $(v, w) \in \mathbf{E} \Leftrightarrow (w, v) \in \mathbf{E}$ , and also strongly connected. To each node  $v$  in the graph a value  $\text{weight}(v)$  is associated, taken from some set  $\mathbf{M}$ . We assume that  $\mathbf{M}$  contains an element *unit* and that there is a binary operator  $\circ$

<sup>1</sup>Here we use the word “strong” in the sense of [14]. Actually, the notions of simulation that we consider here are *weak* in the sense of [12] since their definitions include reachability conditions.

on  $\mathbf{M}$ , such that  $(\mathbf{M}, \circ, \text{unit})$  is an Abelian monoid (so  $\circ$  is commutative and associative and has unit element  $\text{unit}$ ).

Nodes of  $\mathbf{G}$  represent autonomous processors and links represent communication channels via which these processors can send messages to each other. We assume that the communication channels are reliable and that messages are received in the same order as they are sent. We discuss a simple distributed algorithm to compute the sum of the weights of all the nodes in the network. The algorithm is a minor rephrasing of an algorithm described by Chou [3], which in turn is a variant of Segall's PIF (Propagation of Information with Feedback) protocol [18].

The only messages that are required by the algorithm are elements from  $\mathbf{M}$ . A node in the network enters the protocol when it receives a first message from one of its neighbors. Initially, the communication channels for all the links are empty, except the channel associated to the link  $e_0$  from a fixed root node  $v_0$  to itself, which contains a single message.<sup>2</sup> When an arbitrary node  $v$  receives a first message, it marks the node  $w$  from which this message was received. It then sends a unit message to all its neighbors, except  $w$ . Upon receiving subsequent messages, the values of these messages are added to the weight of  $v$ . As soon as, for a non-root node, the total number of received messages equals the total number of neighbors, the value that has been computed is sent back to the node from which the first message was received. When, for root node  $v_0$ , the total number of received messages equals the total number of neighbors, the value that has been computed by  $v_0$  is produced as the final outcome of the algorithm.

In Figure 1, the algorithm is specified as an automaton  $SUM$  using the standard precondition/effect style of the I/O automata model [10, 11, 4]. A minor subtlety is the occurrence of the variable  $v$  in the definition of the step relation, which is neither a state variable nor a formal parameter of the actions. Semantically, the meaning of this  $v$  is determined by an implicit existential quantification: an action  $a$  is enabled in a state  $s$  if there exists a valuation  $\xi$  of all the variables (including  $v$ ) that agrees with  $s$  on the state variables and with  $a$  on the parameters of the actions, such that the precondition of  $a$  holds under  $\xi$ . If action  $a$  is enabled in  $s$  under  $\xi$  then the effect part of  $a$  together with  $\xi$  determine the resulting state  $s'$ .

For each link  $e=(v, w)$ , the source  $v$  is denoted  $\text{source}(e)$ , the target  $w$  is denoted  $\text{target}(e)$ , and the reverse link  $(w, v)$  is denoted  $e^{-1}$ . For each node  $v$ ,  $\text{from}(v)$  gives the set of links with source  $v$  and  $\text{to}(v)$  gives the set of links with target  $v$ , so  $e \in \text{from}(v) \Leftrightarrow \text{source}(e)=v$  and  $e \in \text{to}(v) \Leftrightarrow \text{target}(e)=v$ . All the other data types and operation symbols used in the specification have the obvious meaning. The states of  $SUM$  are interpretations of five state variables in their domains. The first four of these variables represent the values of program variables at each node:

<sup>2</sup>The assumption that  $e_0 = (v_0, v_0) \in \mathbf{E}$  is not required, but allows for a more uniform description of the algorithm for each node.

**Internal:** *MSG*  
*REPORT*  
**External:** *RESULT*

**State Variables:**  $busy \in \mathbf{V} \rightarrow \mathbf{Bool}$   
 $parent \in \mathbf{V} \rightarrow \mathbf{E}$   
 $total \in \mathbf{V} \rightarrow \mathbf{M}$   
 $cnt \in \mathbf{V} \rightarrow \mathbf{Int}$   
 $mq \in \mathbf{E} \rightarrow \mathbf{M}^*$

**Init:**  $\bigwedge_v \neg busy[v]$   
 $\bigwedge_e mq[e] = \text{if } e=e_0 \text{ then append(unit, empty) else empty}$

**MSG**( $e : \mathbf{E}, m : \mathbf{M}$ )  
**Precondition:**  
 $v = \text{target}(e) \wedge m = \text{head}(mq[e])$   
**Effect:**  
 $mq[e] := \text{tail}(mq[e])$   
**if**  $\neg busy[v]$  **then**  $busy[v] := \text{true}$   
 $parent[v] := e$   
 $total[v] := \text{weight}(v)$   
 $cnt[v] := \text{size}(\text{from}(v)) - 1$   
**for**  $f \in \text{from}(v) / \{e^{-1}\}$  **do**  $mq[f] := \text{append}(\text{unit}, mq[f])$   
**else**  $total[v] := total[v] \circ m$   
 $cnt[v] := cnt[v] - 1$

**REPORT**( $e : \mathbf{E}, m : \mathbf{M}$ )  
**Precondition:**  
 $v = \text{source}(e) \neq v_0 \wedge busy[v] \wedge cnt[v] = 0 \wedge e^{-1} = parent[v] \wedge m = total[v]$   
**Effect:**  
 $busy[v] := \text{false}$   
 $mq[e] := \text{append}(m, mq[e])$

**RESULT**( $m : \mathbf{M}$ )  
**Precondition:**  
 $busy[v_0] \wedge cnt[v_0] = 0 \wedge m = total[v_0]$   
**Effect:**  
 $busy[v_0] := \text{false}$

FIGURE 1. Automaton *SUM*.



- *busy* tells for each node whether or not it is currently participating in the protocol; initially *busy*[*v*] equals *false* for each *v*;
- *parent* is used to remember the link via which a node has been activated;
- *total* records the sum of the values seen by a node during a run of the protocol;
- *cnt* gives the number of values that a node still wants to see before it will terminate.

State variable *mq*, finally, represents the contents of the message queue for each link. Initially, *mq*[*e*] is empty for each link *e* except  $e_0$ .

Automaton *SUM* has three types of actions: an action *MSG*, which describes the receipt and processing of a message, an action *REPORT*, by which a non root node sends the final value that it has computed to its parent, and an action *RESULT*, which is the last action of the algorithm, used by the root node to output the final result of the computation.

#### 4 CORRECTNESS PROOF

The correctness property  $\Phi$  of *SUM* that we want to establish is that each maximal execution of the automaton consists of a finite number of internal actions followed by the single output action  $RESULT(\sum_{v \in \mathbf{V}} \text{weight}(v))$ .

Intuitively, propagation of messages occurs in two phases. First unit messages are sent from node  $v_0$  into the network, and then partial sums flow back from the network to  $v_0$ . In the first phase a spanning tree is constructed with root  $v_0$  and this spanning tree is used to accumulate values in the second phase.

##### 4.1 Adding a History Variable

A first important observation about the algorithm is that in each run at most one message travels on each link. In order to state this property formally as an invariant, we add a so-called “history variable” *sent* to automaton *SUM* that records for each link *e* how many messages have been sent on *e*. Figure 2 describes the automaton  $SUM^h$  obtained in this way. Variable *sent* is an auxiliary/history variable in the sense of Owicki and Gries [16] because it does not occur in conditions nor at the right-hand-side of assignments to other variables. Clearly, adding *sent* does not change the behavior of automaton *SUM*. This can be formalized via the following trivial lemma, which in turn implies that *SUM* satisfies correctness property  $\Phi$  if and only if  $SUM^h$  does.

**LEMMA 2** *The inverse of the projection function that maps states from  $SUM^h$  to states of *SUM* is a strong history relation from *SUM* to  $SUM^h$ .*

Invariant 1 below gives a basic sanity property of  $SUM^h$ : at any time the number of messages in a link is at most equal to the number of messages that have been sent on that link.

<b>Internal:</b>	<i>MSG</i> <i>REPORT</i>
<b>External:</b>	<i>RESULT</i>
<b>State Variables:</b>	$busy \in \mathbf{V} \rightarrow \mathbf{Bool}$ $parent \in \mathbf{V} \rightarrow \mathbf{E}$ $total \in \mathbf{V} \rightarrow \mathbf{M}$ $cnt \in \mathbf{V} \rightarrow \mathbf{Int}$ $mq \in \mathbf{E} \rightarrow \mathbf{M}^*$ $sent \in \mathbf{E} \rightarrow \mathbf{Int}$
<b>Init:</b>	$\bigwedge_v \neg busy[v]$ $\bigwedge_e mq[e] = \text{if } e=e_0 \text{ then append(unit, empty) else empty}$ $\bigwedge_e sent[e] = \text{if } e=e_0 \text{ then 1 else 0}$
<b>MSG(<math>e : \mathbf{E}, m : \mathbf{M}</math>)</b>	<p><b>Precondition:</b> <math>v = \text{target}(e) \wedge m = \text{head}(mq[e])</math></p> <p><b>Effect:</b>  <math>mq[e] := \text{tail}(mq[e])</math>  <b>if</b> <math>\neg busy[v]</math> <b>then</b> <math>busy[v] := \text{true}</math>  <math>parent[v] := e</math>  <math>total[v] := \text{weight}(v)</math>  <math>cnt[v] := \text{size}(\text{from}(v)) - 1</math>  <b>for</b> <math>f \in \text{from}(v) / \{e^{-1}\}</math> <b>do</b> <math>mq[f] := \text{append}(\text{unit}, mq[f])</math>  <math>sent[f] := sent[f] + 1</math></p> <p><b>else</b> <math>total[v] := total[v] \circ m</math>  <math>cnt[v] := cnt[v] - 1</math></p>
<b>REPORT(<math>e : \mathbf{E}, m : \mathbf{M}</math>)</b>	<p><b>Precondition:</b> <math>v = \text{source}(e) \neq v_0 \wedge busy[v] \wedge cnt[v] = 0 \wedge e^{-1} = parent[v] \wedge m = total[v]</math></p> <p><b>Effect:</b>  <math>busy[v] := \text{false}</math>  <math>mq[e] := \text{append}(m, mq[e])</math>  <math>sent[e] := sent[e] + 1</math></p>
<b>RESULT(<math>m : \mathbf{M}</math>)</b>	<p><b>Precondition:</b> <math>busy[v_0] \wedge cnt[v_0] = 0 \wedge m = total[v_0]</math></p> <p><b>Effect:</b> <math>busy[v_0] := \text{false}</math></p>

FIGURE 2. Automaton  $SUM^h$  obtained from  $SUM$  by adding history variable *sent*.

INVARIANT 1 For all reachable states of  $SUM^h$  and for all  $e$ :

$$\text{len}(mq[e]) \leq \text{sent}[e]$$

At first sight, Invariant 2 below may look a bit complicated. It is however easy to give intuition for it. The key part of the invariant is the first conjunct, which states that at most one message travels on each link. The other conjuncts are only needed to get the induction to work in the invariant proof. The second and third conjunct imply that if in a *MSG* step a value is sent into some channel, this channels must have been empty in the start state of that step. The fourth conjunct allows to prove a similar property for *REPORT* steps. The routine proof of Invariant 2, which has been omitted here, uses Invariant 1.

INVARIANT 2 For all reachable states of  $SUM^h$  and for all  $v$  and  $e$ :

$$\begin{aligned} & \wedge \text{sent}[e] \leq 1 \\ & \wedge \text{len}(mq[e_0])=1 \rightarrow (\forall f \in \text{from}(v_0)/\{e_0\} : \text{sent}[f]=0) \\ & \wedge v \neq v_0 \wedge \neg \text{busy}[v] \wedge e \in \text{to}(v) \wedge \text{len}(mq[e])=1 \rightarrow (\forall f \in \text{from}(v) : \text{sent}[f]=0) \\ & \wedge v \neq v_0 \wedge \text{busy}[v] \wedge e = \text{parent}[v] \rightarrow \text{sent}[e^{-1}]=0 \end{aligned}$$

Invariant 2 is quite powerful and implies in particular that the algorithm will always terminate.

COROLLARY 3 Automaton  $SUM^h$  has no infinite executions.

PROOF: Define the state function *Norm* as follows:

$$\text{Norm} \triangleq \sum_{e \in \mathbf{E}} 2 \cdot \text{sent}[e] - \text{len}(mq[e])$$

Since both sending and receiving a value increases *Norm*, each step of  $SUM^h$  with label *MSG* or *REPORT* increases *Norm*. By Invariant 2, *Norm* can be at most  $2 \cdot \text{size}(\mathbf{E})$ , for any reachable state. Therefore there can be at most finitely many steps labeled by an internal actions in any execution of  $SUM^h$ . Since each *RESULT* step changes the value of *busy*[ $v_0$ ] from true to false, there can be at most one *RESULT* step after the last internal step.  $\square$

A next property that we will established is that each node can be activated only once in any run of the algorithm. We say that node  $v$  is *activated* in a step if *busy*[ $v$ ] changes from false to true in that step. This implies that  $v$  has been activated iff it has received at least one message. The number of messages received by a node  $v$  equals the number of messages that have been sent to  $v$  minus the number of messages still in transit, and is therefore given by the state function:

$$\text{Received}(v) \triangleq \sum_{e \in \text{to}(v)} \text{sent}[e] - \text{len}(mq[e])$$

The following Invariant 3 gives a characterization of the value of *Received*( $v$ ) for reachable states. The proof is straightforward and uses Invariant 2.

INVARIANT 3 For all reachable states of  $SUM^h$  and for all  $v$ :

$$\begin{aligned} \wedge \text{ busy}[v] &\rightarrow \text{Received}(v) = \text{size}(\text{to}(v)) - \text{cnt}[v] > 0 \\ \wedge \neg \text{ busy}[v] &\rightarrow \text{Received}(v) = 0 \vee \text{Received}(v) = \text{size}(\text{to}(v)) \end{aligned}$$

Invariants 2 and 3 together imply that each node is activated at most once in each execution. Because suppose that in some reachable state some node  $v$  is both inactive and activated. This means  $\neg \text{ busy}[v] \wedge \text{Received}(v) > 0$ . Then Invariant 3 gives  $\text{Received}(v) = \text{size}(\text{to}(v))$ . But this implies that no *MSG* action can be enabled, because this would violate Invariant 2.

We conclude this subsection with two simple invariants that we will use later on.

INVARIANT 4 For all reachable states of  $SUM^h$  and for all  $v$ :

$$\text{Received}(v) > 0 \rightarrow v = \text{target}(\text{parent}[v])$$

INVARIANT 5 For all reachable states of  $SUM^h$  and for all  $e$ :

$$e \neq e_0 \wedge \text{mq}[e] \neq \text{empty} \rightarrow \text{Received}(\text{source}(e)) > 0$$

#### 4.2 Adding a Prophecy Variable

Intuitively, in the first phase of the algorithm a spanning tree is constructed with root  $v_0$ , and this spanning tree is used to accumulate values in the second phase. When the algorithm starts, it not clear how the spanning tree is going to look like and in fact any spanning tree is still possible. While the algorithm proceeds, the spanning tree is constructed step by step. The choice whether an arbitrary link will be part of the spanning tree depends on the relative speeds of the processors, and is entirely nondeterministic. Such unpredictable, nondeterministic behavior is typical for distributed computation but often complicates analysis. Fortunately, the concept of *prophecy variables* of Abadi and Lamport [1] allows us to drastically reduce the nondeterminism of the algorithm or, more precisely, to push nondeterminism backwards to the initial state. We add to  $SUM^h$  a new variable *tree*, which records an initial guess of the full spanning tree and is used to enforce that the actual tree that is constructed during execution is equal to this initial guess. Figure 3 describes the automaton  $SUM^{hp}$  obtained in this way. In Figure 3, *tree* is the function that tells for each set of links whether or not it is a tree. More formally, for  $T \subseteq \mathbf{E}$  and  $E = \{\text{source}(e), \text{target}(e) \mid e \in T\}$ ,  $\text{tree}(T) = \text{true}$  iff either  $T = \emptyset$  or there exists a node  $v \in E$  such that for all  $v' \in E$  there is a unique path of links in  $T$  leading from  $v$  to  $v'$ .

In order to show that *tree* is a prophecy variable in the sense of [1, 12], we establish a prophecy relation from  $SUM^h$  to  $SUM^{hp}$ . For this, we need three more invariants. The proof of Invariant 6 uses Invariants 3, 4 and 5. Invariants 7 and 8 are completely trivial.

<b>Internal:</b>	<i>MSG</i>
	<i>REPORT</i>
<b>External:</b>	<i>RESULT</i>
<b>State Variables:</b>	$busy \in \mathbf{V} \rightarrow \mathbf{Bool}$
	$parent \in \mathbf{V} \rightarrow \mathbf{E}$
	$total \in \mathbf{V} \rightarrow \mathbf{M}$
	$cnt \in \mathbf{V} \rightarrow \mathbf{Int}$
	$mq \in \mathbf{E} \rightarrow \mathbf{M}^*$
	$sent \in \mathbf{E} \rightarrow \mathbf{Int}$
	$tree \in \mathbf{V} \rightarrow \mathbf{E}$
<b>Init:</b>	$\bigwedge_v \neg busy[v]$ $\bigwedge_e mq[e] = \text{if } e=e_0 \text{ then append(unit, empty) else empty}$ $\bigwedge_e sent[e] = \text{if } e=e_0 \text{ then 1 else 0}$ $\bigwedge_v tree[v_0] = e_0 \wedge v = \text{target}(tree[v]) \wedge \text{tree}(\{tree[v] \mid v \in \mathbf{V}/\{v_0\}\})$
<i>MSG</i> ( $e : \mathbf{E}, m : \mathbf{M}$ )	
<b>Precondition:</b>	$v = \text{target}(e) \wedge m = \text{head}(mq[e]) \wedge (\neg busy[v] \rightarrow e = tree[v])$
<b>Effect:</b>	$mq[e] := \text{tail}(mq[e])$ <b>if</b> $\neg busy[v]$ <b>then</b> $busy[v] := \text{true}$ $parent[v] := e$ $total[v] := \text{weight}(v)$ $cnt[v] := \text{size}(\text{from}(v)) - 1$ <b>for</b> $f \in \text{from}(v)/\{e^{-1}\}$ <b>do</b> $mq[f] := \text{append}(\text{unit}, mq[f])$ $sent[f] := sent[f] + 1$ <b>else</b> $total[v] := total[v] \circ m$ $cnt[v] := cnt[v] - 1$
<i>REPORT</i> ( $e : \mathbf{E}, m : \mathbf{M}$ )	
<b>Precondition:</b>	$v = \text{source}(e) \neq v_0 \wedge busy[v] \wedge cnt[v] = 0 \wedge e^{-1} = parent[v] \wedge m = total[v]$
<b>Effect:</b>	$busy[v] := \text{false}$ $mq[e] := \text{append}(m, mq[e])$ $sent[e] := sent[e] + 1$
<i>RESULT</i> ( $m : \mathbf{M}$ )	
<b>Precondition:</b>	$busy[v_0] \wedge cnt[v_0] = 0 \wedge m = total[v_0]$
<b>Effect:</b>	$busy[v_0] := \text{false}$

FIGURE 3. Automaton  $SUM^{hp}$  obtained from  $SUM^h$  by adding prophecy variable *tree*.

INVARIANT 6 Let  $T$  be the state function defined by

$$T \triangleq \{\text{parent}[v] \mid v \neq \mathbf{v}_0 \wedge \text{Received}(v) > 0\}$$

Then  $\text{tree}(T)$  holds for all reachable states of  $SUM^h$ .

INVARIANT 7 For all reachable states of  $SUM^{hp}$  and for all  $v$ :

$$\text{Received}(v) > 0 \rightarrow \text{parent}[v] = \text{tree}[v]$$

INVARIANT 8 For all reachable states of  $SUM^{hp}$  and for all  $v$ :

$$\text{tree}[\mathbf{v}_0] = \mathbf{e}_0 \wedge v = \text{target}(\text{tree}[v]) \wedge \text{tree}(\{\text{tree}[v] \mid v \in \mathbf{V}/\{\mathbf{v}_0\}\})$$

LEMMA 4 The inverse of the projection function  $\pi$  that maps states of  $SUM^{hp}$  to states of  $SUM^h$  is a strong image-finite prophecy relation from  $SUM^h$  to  $SUM^{hp}$ .

PROOF: Mapping  $\pi$  is trivially a strong refinement from  $SUM^{hp}$  to  $SUM^h$ . Since the domain of variable  $\text{tree}$  is finite,  $\pi^{-1}$  is image-finite. We prove that  $\pi^{-1}$  satisfies the three conditions of a backward simulation (condition (b) in the strong sense).

For condition (a), suppose that  $s$  is a start state of  $SUM^h$  and  $u$  is a reachable state of  $SUM^{hp}$  with  $\pi(u) = s$ . Then it follows by Invariant 8 that  $u$  is a start state of  $SUM^{hp}$ .

To prove that  $\pi^{-1}$  satisfies conditions (b) and (c) we need the following claim: a state  $u$  of  $SUM^{hp}$  is reachable iff  $\pi(u)$  is reachable and  $u$  satisfies the properties of Invariants 7 and 8. Direction “ $\Rightarrow$ ” of this claim follows by induction on the length of the shortest execution to  $u$ , and uses the fact that  $\pi$  is a strong refinement together with Invariants 7 and 8. Direction “ $\Leftarrow$ ” of the claim follows by induction on the length of the shortest execution to  $\pi(u)$ .

Using the claim, it is routine to prove condition (b). Condition (c) follows from the claim together with Invariant 6.  $\square$

Note that as a direct corollary of Lemma 4 all invariants of  $SUM^h$  are also invariants of  $SUM^{hp}$ .

### 4.3 A Refinement

In this subsection we will prove that there exists a refinement from automaton  $SUM^{hp}$  to the automaton  $S$  defined in Figure 4. Automaton  $S$  is extremely simple. It has only two states: an initial state where  $\text{done}=\text{false}$  and a final state where  $\text{done}=\text{true}$ . There is one step, which starts in the initial state, has label  $\text{RESULT}(\sum_{v \in \mathbf{V}} \text{weight}(v))$ , and ends in the final state.

Define state functions  $\text{Init}$  and  $\text{Done}$  by

$$\begin{aligned} \text{Init}(v) &\triangleq \neg \text{busy}[v] \wedge \text{Received}(v) = 0 \\ \text{Done}(v) &\triangleq \neg \text{busy}[v] \wedge \text{Received}(v) = \text{size}(\text{to}(v)) \end{aligned}$$

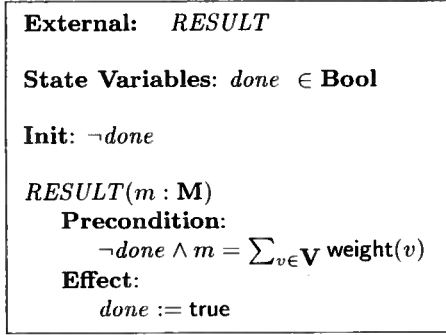


FIGURE 4. Automaton *S*.

As a consequence of Invariant 3, each reachable state of  $SUM^{hp}$  satisfies, for each  $v$ , either  $Init(v)$  or  $busy[v]$  or  $Done(v)$ . In order to establish a refinement from  $SUM^{hp}$  to *S*, we again need two extra invariants. Invariant 9 states that, until the moment where computation has finished, there is a conservation of weight in the network. Invariant 10 allows us to prove that in a state where *RESULT* is enabled,  $Done(v)$  holds for all nodes except  $v_0$ .

INVARIANT 9 *For all reachable states of  $SUM^{hp}$ :*

$$\neg Done(v_0) \rightarrow \sum_{v \in \mathbf{V}} \text{weight}(v) = \sum_{\{v \in \mathbf{V} \mid Received(v)=0\}} \text{weight}(v) + \sum_{\{v \in \mathbf{V} \mid busy[v]\}} total[v] + \sum_{\{e \in \mathbf{E} \mid mq[e] \neq \text{empty}\}} head(mq[e])$$

INVARIANT 10 *For all reachable states of  $SUM^{hp}$  and for all  $v$  and  $e$ :*

$$v \neq v_0 \wedge e = tree[v] \wedge sent[e^{-1}] = 1 \rightarrow Done(v)$$

LEMMA 5 *The function  $r$  from states of  $SUM^{hp}$  to states of *S* given by*

$$r(s) \models done \Leftrightarrow s \models Done(v_0)$$

*is a refinement from  $SUM^{hp}$  to *S*.*

#### 4.4 Absence of Deadlock

The existence of a refinement mapping from  $SUM^{hp}$  to *S* does not guarantee that automaton  $SUM^{hp}$  will produce any output: the automaton still may

have an infinite loop of internal actions or get into a state of deadlock before an output step has been done. We can easily prove the absence of infinite loops by using the result of Corollary 3 that  $SUM^h$  has no infinite executions and the fact that there is a strong prophecy relation from  $SUM^h$  to  $SUM^{hp}$ . The proof that  $SUM^{hp}$  has no premature deadlocks is more involved and requires three additional invariants.

INVARIANT 11 *For all reachable states of  $SUM^{hp}$ ,  $sent[e_0] = 1$ .*

INVARIANT 12 *For all reachable states of  $SUM^{hp}$  and for all  $v$  and  $e$ :*

$$e = tree[v] \wedge Init(v) \wedge mq[e] = \text{empty} \rightarrow Init(\text{source}(e))$$

INVARIANT 13 *For all reachable states of  $SUM^{hp}$  and for all  $v$  and  $e$ :*

$$\neg Init(v) \wedge \text{source}(e) = v \wedge e^{-1} \neq tree[v] \rightarrow sent[e] = 1$$

LEMMA 6 *A reachable state of  $SUM^{hp}$  has no outgoing steps if and only if  $Done(v_0)$  holds in that state.*

PROOF: (Sketch)

“ $\Leftarrow$ ” If  $Done(v_0)$  holds then we can prove using Invariant 10 that  $Done(v)$  holds for all nodes  $v$ . Then Invariants 2 and 3 together imply that no message is in transit. Consequently, no step of  $SUM^{hp}$  is enabled.

“ $\Rightarrow$ ” Suppose that some given state is deadlocked. Then no message can be in transit on the spanning tree, otherwise a *MSG* step would be enabled. This implies, by Invariants 11 and 13, that  $\neg Init(v)$  holds for all nodes  $v$ . This in turn implies that no message can be in transit on *any* link in the network (otherwise a *MSG* action would be enabled). Next we use Invariant 13 to infer that exactly one message has been sent on each link in the network, except those on the reversed spanning tree. Finally, we prove for all nodes  $v$  of the network, starting with the leaves of the tree, that  $v$  has received a message over all incoming links; since no *REPORT* or *RESULT* action is enabled in  $v$  this implies  $Done(v)$ .  $\square$

THEOREM 7 *Automaton  $SUM$  satisfies property  $\Phi$ .*

PROOF: Follows from the fact that  $SUM^{hp}$  satisfies  $\Phi$  and the existence of a strong history relation from  $SUM$  to  $SUM^h$  and a strong prophecy relation from  $SUM^h$  to  $SUM^{hp}$ .  $\square$

## 5 CONCLUDING REMARKS

The verification of this paper has not yet been proof-checked by computer, but I expect that this will be a routine exercise, building on earlier work on mechanical checking of I/O automata proofs [19, 4, 15]. Although I have carried out the verification using a simple version of the I/O automaton model, it is probably trivial to translate this story to other state based models, such as Lamport’s Temporal Logic of Actions [8].



## REFERENCES

1. M. Abadi and L. Lamport. The existence of refinement mappings. *Theoretical Computer Science*, 82(2):253–284, 1991.
2. K.M. Chandy and J. Misra. *Parallel Program Design. A Foundation*. Addison-Wesley, 1988.
3. C. Chou. Practical use of the notions of events and causality in reasoning about distributed algorithms. CS Report #940035, UCLA, October 1994.
4. L. Helmink, M.P.A. Sellink, and F.W. Vaandrager. Proof-checking a data link protocol. In H. Barendregt and T. Nipkow, editors, *Proceedings International Workshop TYPES'93*, Nijmegen, The Netherlands, May 1993, volume 806 of *Lecture Notes in Computer Science*, pages 127–165. Springer-Verlag, 1994. Full version available as Report CS-R9420, CWI, Amsterdam, March 1994.
5. B. Jonsson. Compositional specification and verification of distributed systems. *ACM Transactions on Programming Languages and Systems*, 16(2):259–303, March 1994.
6. S.S. Lam and A.U. Shankar. Protocol verification via projections. *IEEE Transactions on Software Engineering*, 10(4):325–342, July 1984.
7. L. Lamport. Specifying concurrent program modules. *ACM Transactions on Programming Languages and Systems*, 5(2):190–222, 1983.
8. L. Lamport. The temporal logic of actions. *ACM Transactions on Programming Languages and Systems*, 16(3):872–923, March 1994.
9. P. Lucas. Two constructive realizations of the block concept and their equivalence. Technical Report 25.085, IBM Laboratory, Vienna, June 1968.
10. N.A. Lynch and M.R. Tuttle. Hierarchical correctness proofs for distributed algorithms. In *Proceedings of the 6<sup>th</sup> Annual ACM Symposium on Principles of Distributed Computing*, pages 137–151, August 1987. A full version is available as MIT Technical Report MIT/LCS/TR-387.
11. N.A. Lynch and M.R. Tuttle. An introduction to input/output automata. *CWI Quarterly*, 2(3):219–246, September 1989.
12. N.A. Lynch and F.W. Vaandrager. Forward and backward simulations – part I: Untimed systems. Report CS-R9313, CWI, Amsterdam, March 1993. Also, MIT/LCS/TM-486.b, Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA. To appear in *Information and Computation*.
13. Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. Springer-Verlag, 1992.
14. R. Milner. *Communication and Concurrency*. Prentice-Hall International, Englewood Cliffs, 1989.
15. T. Nipkow and K. Slind. I/O automata in Isabelle/HOL, 1994. Draft paper.
16. S. Owicki and D. Gries. An axiomatic proof technique for parallel programs. *Acta Informatica*, 6(4):319–340, 1976.
17. P. Humblet R. Gallager and P. Spira. A distributed algorithm for minimum-weight spanning trees. *ACM Transactions on Programming Languages and Systems*, 5(1):66–77, January 1983.

18. A. Segall. Distributed network protocols. *IEEE Transactions on Information Theory*, IT-29(2):23–35, January 1983.
19. J. Sogaard-Andersen, S. Garland, J. Guttag, N.A. Lynch, and A. Pogoyants. Computer-assisted simulation proofs. In C. Courcoubetis, editor, *Proceedings of the 5th International Conference on Computer Aided Verification*, Elounda, Greece, volume 697 of *Lecture Notes in Computer Science*, pages 305–319. Springer-Verlag, 1993.

# Stability Analysis of a Difference Scheme for Three-Dimensional Advection-Diffusion Problems <sup>1</sup>

*Dedicated to Cor Baayen at the Occasion of his Retirement*

*as our Scientific Director*

J.G. Verwer  
B.P. Sommeijer  
CWI

## 1 INTRODUCTION

### 1.1 General

The authors of this contribution belong to the research group *Discretization of Evolution Problems* of CWI's *Numerical Mathematics Department*. This research group focuses on fundamental and applied research into numerical methods for evolutionary differential equations. Both ordinary and partial differential equations are treated. In recent years much attention is devoted to large-scale applications and high performance computing. In this connection, an important research subject concerns *Transport Problems in Environmental Applications* which are constituted by systems of time-dependent partial differential equations of the advection-diffusion-reaction type. Numerical research for this type of problems is important for the simulation and prediction of the chemistry and transport of hazardous pollutants in the atmosphere, groundwater and shallow water. Because the systems are usually three-dimensional in space and usually contain many components, one for each chemical or biological constituent in the model, they are extremely CPU and memory intensive and in fact belong to the computationally most expensive models in environmental research and fluid dynamics. Consequently, high performance computing on powerful vector and parallel computers is an important field of research for these applications.

Moreover, when new methods and techniques designed for high-performance use on such computers are developed, also their fundamental numerical properties need to be investigated, notably their stability, consistency and convergence properties. The present contribution provides an example of such a theoretical investigation. This paper deals with a linear stability analysis of a method recently designed in our group for the numerical integration of transport problems in shallow water on vector and parallel computers. To appreciate the

---

<sup>1</sup>This research was supported by Cray Research Inc. under grant CRG 94.04 via the Stichting Nationale Computerfaciliteiten (National Computing Facilities Foundation, NCF).

complete paper the reader should have a numerical background. Fortunately, the linear stability analysis for difference schemes of the type considered here is based on the well-known Fourier method as proposed by J. Von Neumann (see [8], which is one of the earliest papers where the Fourier method is applied to finite-difference equations). This means that an important part of the paper, viz. Section 3, should be accessible, and hopefully is of some interest, for many readers without any numerical background.

Section 3 is almost self-contained. Here we study the problem of determining the location of the zeros of a polynomial relative to the unit circle in the complex plane. This problem is of long standing (see Schur [11]) and of great practical relevance in applied mathematics (see Miller [6]). In our case we have to deal with a quadratic polynomial whose coefficients are complex-valued functions of a real variable, a phase angle. These functions are determined by the difference scheme and contain so-called advection and diffusion parameters. The question is what conditions should be imposed on these coefficient functions, and hence on their defining parameters, such that the two zeros lie on the unit disc for all phase angles. The resulting conditions determine the critical stepsize for the linear time step stability of the difference scheme. The analysis to solve this stability question shows interesting aspects and surprising results.

### *1.2 Research contents*

In [12] and [13] an odd-even-line hopscotch (OELH) method is developed and implemented for the efficient numerical solution of three-space dimensional advection-diffusion problems modeling the transport of pollutants and suspended material in shallow water. A special feature of this OELH method is that it is explicit for the horizontal transport and implicit for the vertical transport. The implicitness in the vertical direction is necessary to avoid a too stringent stability restriction on the time step. This implicitness gives rise to the solution of a large set of tridiagonal systems, one for every grid point in the horizontal plane. The solution of this large set of tridiagonal systems can be vectorized and parallelized over the horizontal grid, which results in a very good performance [13]. In the comparison with other techniques discussed in [12, 13], the method has been shown superior.

In neither of the aforementioned two papers a comprehensive stability analysis is given. The purpose of the present paper is to fill up this gap. For the general, constant coefficient, linear advection-diffusion model problem we will derive sufficient and necessary conditions for von Neumann stability in the strict sense. Strict means that the stability property we investigate requires the absolute value of amplification factors less than or equal to one. The stability analysis is based on an equivalence with an associated scheme which is composed of the leap-frog, the Du Fort-Frankel, and the Crank-Nicholson scheme. The actual Fourier analysis is carried out for this associated scheme and appears to be rather intricate. For example, the resulting expressions for critical stepsizes reveal that the presence of horizontal diffusion generally leads to a

smaller value, in spite of the fact that we have unconditional stability for pure diffusion problems.

## 2 THE OELH METHOD FORMULATED FOR THE MODEL PROBLEM

We consider the 3D, constant coefficient, scalar advection-diffusion model problem

$$u_t + q_1 u_x + q_2 u_y + q_3 u_z = \epsilon_1 u_{xx} + \epsilon_2 u_{yy} + \epsilon_3 u_{zz}. \quad (1)$$

Let

$$\frac{d}{dt} U_{ijk} = L_h U_{ijk} \quad (2)$$

be the semi-discrete approximation, resulting from the use of 2nd-order central differences at the uniformly spaced gridpoints

$$(x_i, y_j, z_k) = (ih_1, jh_2, kh_3).$$

The basic formula [1, 2, 3, 4] defining the OELH method studied in [12, 13] then reads

$$U_{\underline{i}}^{n+1} = U_{\underline{i}}^n + \tau \theta_{\underline{i}}^n L_h U_{\underline{i}}^n + \tau \theta_{\underline{i}}^{n+1} L_h U_{\underline{i}}^{n+1}, \quad (3)$$

where  $\underline{i} = (i, j, k)$ ,  $\tau = t_{n+1} - t_n$ , and the hopscotch parameter  $\theta_{\underline{i}}^n$  is defined by

$$\theta_{\underline{i}}^n = \begin{cases} 1 & \text{for odd values of } n + i + j, \\ 0 & \text{for even values of } n + i + j. \end{cases} \quad (4)$$

Notice that the subscript  $k$  is not involved in this definition, i.e., all gridpoints on a vertical gridline have the same  $\theta$ -value. If we consider only the odd points (in the space-time grid), then the forward Euler rule results,

$$U_{\underline{i}}^{n+1} = U_{\underline{i}}^n + \tau L_h U_{\underline{i}}^n, \quad (5)$$

and at the even points, for the same  $n$ , we have the backward Euler rule

$$U_{\underline{i}}^{n+1} = U_{\underline{i}}^n + \tau L_h U_{\underline{i}}^{n+1}. \quad (6)$$

Consequently, by first applying the explicit forward Euler method at all odd points, and subsequently the implicit backward Euler method at all even points, we have carried out one step with (2.3). The merit of the method lies in the fact that the implicit step is only implicit for the vertical direction. This follows from the 3-point coupling in the horizontal directions and from the definition of the  $\theta_{\underline{i}}^n$ . If we remove the third dimension, then we recover the odd-even-hopscotch scheme (OEH) which is scalarly implicit. Note that the OEH scheme for the 3D problem results if we replace  $(n + i + j)$  in  $\theta_{\underline{i}}^n$  by  $(n + i + j + k)$ . The stability of the OEH scheme applied to (2.1) has been studied in [14].

The von Neumann stability approach cannot be carried out for (2.3) as it stands. Following [3, 14], we therefore derive an equivalent formula which

does admit Fourier analysis. First introduce, for  $m = 1, 2, 3$ , the advection parameter  $c_m$  and the diffusion parameter  $\sigma_m$ ,

$$c_m = \frac{\tau q_m}{h_m}, \quad \sigma_m = \frac{\tau \epsilon_m}{h_m^2}, \quad (7)$$

and the difference operators  $H_m$  and  $\delta_m^2$ ,

$$H_1 U_{\underline{i}} = U_{i+1jk} - U_{i-1jk}, \text{ etc.} \quad (8)$$

$$\delta_1^2 U_{\underline{i}} = U_{i+1jk} - 2U_{ijk} + U_{i-1jk}, \text{ etc.} \quad (9)$$

We then may express  $\tau L_h U_{\underline{i}}$  as

$$\tau L_h U_{\underline{i}} = \sum_{m=1}^3 \left( -\frac{1}{2} c_m H_m + \sigma_m \delta_m^2 \right) U_{\underline{i}}. \quad (10)$$

Next introduce, in addition to (2.3), the OELH formula for the next time step

$$U_{\underline{i}}^{n+2} = U_{\underline{i}}^{n+1} + \tau \theta_{\underline{i}}^{n+1} L_h U_{\underline{i}}^{n+1} + \tau \theta_{\underline{i}}^{n+2} L_h U_{\underline{i}}^{n+2}. \quad (11)$$

Using (2.3), (2.4) and (2.11), for the odd points we then can write, considering time levels  $n$  and  $n + 2$ ,

$$U_{\underline{i}}^{n+2} = U_{\underline{i}}^n + \tau L_h \left( U_{\underline{i}}^n + U_{\underline{i}}^{n+2} \right). \quad (12)$$

Likewise, for the even points we find

$$U_{\underline{i}}^{n+2} = 2U_{\underline{i}}^{n+1} - U_{\underline{i}}^n. \quad (13)$$

Next we elaborate the odd-point formula (2.12). Using (2.13) to eliminate variables at even points, an elementary calculation with (2.10) shows that (2.12) can be written as

$$\begin{aligned} (1 + \sigma) U_{\underline{i}}^{n+2} &= (1 - \sigma) U_{\underline{i}}^n + (4\sigma_1 \mu_1 + 4\sigma_2 \mu_2) U_{\underline{i}}^{n+1} - \\ & (c_1 H_1 + c_2 H_2) U_{\underline{i}}^{n+1} + \left( -\frac{1}{2} c_3 H_3 + \sigma_3 \delta_3^2 \right) \left( U_{\underline{i}}^n + U_{\underline{i}}^{n+2} \right), \end{aligned} \quad (14)$$

where  $\mu_m$  is the averaging operator

$$\mu_1 U_{\underline{i}} = \frac{1}{2} (U_{i+1jk} + U_{i-1jk}), \text{ etc.} \quad (15)$$

and

$$\sigma = 2(\sigma_1 + \sigma_2). \quad (16)$$

It is important to note that in (2.14) only variables at odd numbered points appear. This means that the solution defined by (2.3), can first be computed

by means of (2.14) at the complete set of odd points, and thereafter at the complete set of even points by means of (cf. (2.13))

$$U_{\underline{i}}^{n+1} = \frac{1}{2} (U_{\underline{i}}^n + U_{\underline{i}}^{n+2}). \quad (17)$$

Hence for the stability analysis we may proceed with the odd-point scheme (2.14), because the sets of even and odd points are decoupled.

We see that this odd-point scheme is composed of the leap-frog scheme for the horizontal advection part,

$$U_{\underline{i}}^{n+2} = U_{\underline{i}}^n - (c_1 H_1 + c_2 H_2) U_{\underline{i}}^{n+1}, \quad (18)$$

of the Du Fort-Frankel scheme for the horizontal diffusion part,

$$(1 + \sigma) U_{\underline{i}}^{n+2} = (1 - \sigma) U_{\underline{i}}^n + (4\sigma_1 \mu_1 + 4\sigma_2 \mu_2) U_{\underline{i}}^{n+1}, \quad (19)$$

and of the Crank-Nicholson scheme, with stepsize  $2\tau$ , for the vertical advection and diffusion part,

$$U_{\underline{i}}^{n+2} = U_{\underline{i}}^n + \left(-\frac{1}{2}c_3 H_3 + \sigma_3 \delta_3^2\right) (U_{\underline{i}}^n + U_{\underline{i}}^{n+2}). \quad (20)$$

Consequently, in view of the unconditional stability of the Crank-Nicholson and Du Fort-Frankel scheme, at first sight one might expect that the critical stepsize for stability equals that of the leap-frog scheme (2.18). In the next section we will prove that this is indeed true if there is no horizontal diffusion. However, if horizontal diffusion terms are present, then the situation turns out to be more complicated. We will show that in this case the critical stepsize is generally smaller.

### 3 STRICT VON NEUMANN STABILITY

Substitution of the Fourier mode

$$U_{\underline{i}}^n = \xi^n e^{I(\omega_1 x_i + \omega_2 y_j + \omega_3 z_k)}, \quad I^2 = -1, \quad (21)$$

into scheme (2.14) leads to the characteristic polynomial

$$f(\xi) = a_0 + a_1 \xi + a_2 \xi^2 \quad (22)$$

with coefficients

$$\begin{aligned} a_0 &= -1 + \sigma - 2\sigma_3 (\cos \theta_3 - 1) + I c_3 \sin \theta_3, \\ a_1 &= \sum_{m=1}^2 -4\sigma_m \cos \theta_m + 2I c_m \sin \theta_m, \\ a_2 &= 1 + \sigma - 2\sigma_3 (\cos \theta_3 - 1) + I c_3 \sin \theta_3, \end{aligned} \quad (23)$$

where  $\theta_m = \omega_m h_m$  denotes the phase angle. The specific stability property we will investigate is von Neumann stability in the strict sense:

DEFINITION 1 Method (2.14) is called von Neumann stable if the zeroes  $\xi_1, \xi_2$  of the characteristic polynomial (3.2) satisfy

$$|\xi_1|, |\xi_2| \leq 1 \text{ for all } |\theta_m| \leq \pi, \quad m = 1, 2, 3. \quad (24)$$

Hence strict means that the stability property we investigate requires the absolute value of amplification factors less than or equal to one. In literature, this is also called 'practical' or 'modified' von Neumann stability [9, 7, 5]. Note that the original von Neumann condition is weaker as it requires  $|\xi| \leq 1 + O(\tau)$  [9]. As is well known, for advection-diffusion problems this weaker condition can lead to unacceptably large errors [7]. Strict stability is also more natural here, since Fourier modes of the true solution cannot grow in time either.

For the von Neumann analysis we will use results from [6]. We therefore introduce the polynomial

$$f^*(\xi) = \bar{a}_2 + \bar{a}_1\xi + \bar{a}_0\xi^2, \quad (25)$$

and the so-called first reduced polynomial

$$f_1(\xi) = \bar{a}_2a_1 - \bar{a}_1a_0 + (\bar{a}_2a_2 - \bar{a}_0a_0)\xi, \quad (26)$$

where

$$\begin{aligned} \bar{a}_2a_1 - \bar{a}_1a_0 = & -8 \sum_{m=1}^2 \sigma_m \cos \theta_m + I \left( 8c_3 \sin \theta_3 \sum_{m=1}^2 \sigma_m \cos \theta_m \right) + \\ & I \left( 4(\sigma + 2\sigma_3 - 2\sigma_3 \cos \theta_3) \sum_{m=1}^2 c_m \sin \theta_m \right) \end{aligned} \quad (27)$$

and

$$\bar{a}_2a_2 - \bar{a}_0a_0 = 4(\sigma + 2\sigma_3 - 2\sigma_3 \cos \theta_3). \quad (28)$$

Note that in the pure advection case the first reduced polynomial vanishes, because then  $\sigma_m = 0$  for  $m = 1, 2, 3$ .

In the remainder of this section we will prove and discuss two stability theorems. Theorem 1 deals with the case where horizontal diffusion is absent ( $\epsilon_1 = 0, \epsilon_2 = 0$  and  $\epsilon_3 \geq 0$ ). In Theorem 2 we consider the remaining cases where diffusion exists in at least one of the two horizontal directions ( $\epsilon_1 \geq 0, \epsilon_2 \geq 0, \epsilon_3 \geq 0$  and  $\epsilon_1 + \epsilon_2 > 0$ ). In both theorems all velocities  $c_m$  may take on arbitrary values, including zero.

**THEOREM 1** Suppose  $\epsilon_1 = 0, \epsilon_2 = 0$  and  $\epsilon_3 \geq 0$ . Then we have von Neumann stability if and only if

$$|c_1| + |c_2| \leq 1. \quad (29)$$

**PROOF.** We distinguish the two cases  $\epsilon_3 = 0$  and  $\epsilon_3 > 0$ . First suppose  $\epsilon_3 = 0$ . Then the first reduced polynomial  $f_1 \equiv 0$ , so that according to case (ii) of Th.



6.1 from [6], there holds  $|\xi_1|, |\xi_2| \leq 1$ , if and only if the root  $\xi_0$  of the derivative polynomial  $f'$  satisfies  $|\xi_0| \leq 1$ . Since  $\xi_0 = -a_1/2a_2$  we find

$$|\xi_0|^2 = \frac{\left(\sum_{m=1}^2 c_m \sin \theta_m\right)^2}{1 + c_3^2 \sin^2 \theta_3}, \quad (30)$$

which immediately proves the theorem for the case  $\epsilon_3 = 0$ . Next suppose  $\epsilon_3 > 0$ . Two subcases then must be distinguished, viz. phase angle  $\theta_3 = 0$  and  $\theta_3 \neq 0$ . If  $\theta_3 = 0$ , then again  $f_1 \equiv 0$  and the proof goes the same as above. If  $\theta_3 \neq 0$ , then  $f_1$  does not vanish so that now case (i) of Th. 6.1 from [6] applies. That is,  $|\xi_1|, |\xi_2| \leq 1$ , if and only if

- (a)  $|f^*(0)| > |f(0)|$  and
- (b) The root  $\xi_0$  of  $f_1$  satisfies  $|\xi_0| \leq 1$ .

Condition (a) means  $|\bar{a}_2| > |a_0|$  or, according to (3.8),

$$|a_2|^2 - |a_0|^2 = \bar{a}_2 a_2 - \bar{a}_0 a_0 = 4(\sigma + 2\sigma_3 - 2\sigma_3 \cos \theta_3) > 0. \quad (31)$$

We immediately conclude that condition (a) is unconditionally true because the diffusion parameter  $\sigma_3$  is positive and  $\sigma = 0$ . Generally, condition (b) is true if and only if

$$\left| -2 \sum_{m=1}^2 \sigma_m \cos \theta_m + I \left( 2c_3 \sin \theta_3 \sum_{m=1}^2 \sigma_m \cos \theta_m \right) + I \left( (\sigma + 2\sigma_3 - 2\sigma_3 \cos \theta_3) \sum_{m=1}^2 c_m \sin \theta_m \right) \right| \leq \sigma + 2\sigma_3 - 2\sigma_3 \cos \theta_3. \quad (32)$$

Because  $\sigma_1 = \sigma_2 = 0$  and  $\sigma_3 > 0$ , this inequality simply means that

$$\left| \sum_{m=1}^2 c_m \sin \theta_m \right| \leq 1,$$

which immediately proves the theorem also for the case  $\epsilon_3 > 0$ .  $\square$

In the situation of Theorem 1 the Du Fort-Frankel scheme is absent in (2.14), so that only the leap-frog scheme and the Crank-Nicholson scheme as combined in (2.14) play a role. Theorem 1 nicely shows this. We see that the critical stepsize for von Neumann stability is determined by the familiar CFL condition of the leap-frog scheme (2.18),

$$\tau \left( \frac{|q_1|}{h_1} + \frac{|q_2|}{h_2} \right) \leq 1. \quad (33)$$

This is an optimal result in the sense that the vertical velocity  $q_3$  and the vertical mesh width  $h_3$  are absent in the stability condition, which is due to the unconditional stability of the Crank-Nicholson scheme. Especially  $h_3$  should be absent, since in shallow water transport problems  $h_3$  is significantly smaller than  $h_1$  and  $h_2$ . This, in fact, was the motivation for developing the odd-even-line hopschotch method [12, 13]. Also note that in the case of pure advection ( $\epsilon_m = 0, m = 1, 2, 3$ ) the characteristic polynomial  $f$  is conservative ( $|\xi_1| = |\xi_2| = 1$ ) as long as (3.13) holds (Th. 6.4, [6]). If we impose strict inequality, then  $f$  is simple conservative (conservative and  $\xi_1 \neq \xi_2$ , see [6], Cor. 6.5). This means that in the case of pure advection the OELH scheme does not damp Fourier modes, which is a natural property because the true Fourier modes are not damped either. If  $\epsilon_3 > 0$ , then one of the amplification factors must lie in the open unit disc as long as (3.13) holds, since  $f_1$  does not vanish. If we impose strict inequality in (3.13), then both factors lie in the open unit disc which means damping of Fourier modes similar as for the true solution.

Before we present Theorem 2, we first give a result due to [5] and repeat its proof here for reasons of self-containedness.

LEMMA 1 Consider the finite, real-valued series

$$S = 1 - \sum_{m=1}^M \alpha_m \theta_m^2 + \left( \sum_{m=1}^M c_m \theta_m \right)^2.$$

Suppose  $\alpha_m \geq 0$  for all  $m = 1, \dots, M$ . Then we have  $S \leq 1$  for all  $\theta_m$ , if and only if

$$\sum_{m=1}^M \frac{c_m^2}{\alpha_m} \leq 1.$$

PROOF. Denote

$$\alpha = \text{diag}(\alpha_1, \dots, \alpha_M), \vec{c} = (c_1, \dots, c_M)^T, \vec{\theta} = (\theta_1, \dots, \theta_M)^T.$$

Then  $S$  can be expressed as

$$S = 1 - \vec{\theta}^T (\alpha - \vec{c} \vec{c}^T) \vec{\theta}.$$

Thus, we have  $S \leq 1$  for all  $\vec{\theta}$ , if and only if the matrix  $\beta = \alpha - \vec{c} \vec{c}^T$  is non-negative definite. In particular, its diagonal elements  $\alpha_m - c_m^2$  must be non-negative, so that  $\alpha_m = 0$  implies  $c_m = 0$  and the  $m$ -th dimension can be dropped. Hence in the remainder of the proof we may assume all  $\alpha_m > 0$ . If we then define

$$\gamma = \alpha^{-1/2} = \text{diag}(\alpha_1^{-1/2}, \dots, \alpha_M^{-1/2}),$$

we have  $\beta = \alpha^{1/2} (I_M - \gamma \vec{c} \vec{c}^T \gamma) \alpha^{1/2}$  and the matrix

$$\beta' = I_M - \gamma \vec{c} \vec{c}^T \gamma = I_M - (\gamma \vec{c})(\gamma \vec{c})^T = I_M - \vec{d} \vec{d}^T,$$

where  $\vec{d} = \gamma\vec{c}$ , must also be non-negative. This, in turn, means non-negativity of

$$\vec{z}^T \beta' \vec{z} = \vec{z}^T \vec{z} - (\vec{d}^T \vec{z})^2$$

for all  $\vec{z}$ . We can deduce that this is true if and only if

$$\vec{d}^T \vec{d} \leq 1.$$

Sufficiency follows immediately from the Cauchy-Schwarz inequality

$$(\vec{d}^T \vec{z})^2 \leq (\vec{d}^T \vec{d})(\vec{z}^T \vec{z})$$

and necessity by selecting  $z_m = cd_m$  for  $m = 1, \dots, M$ , where  $c$  is an arbitrary constant. Since  $\vec{d}^T \vec{d} = \sum c_m^2 / \alpha_m$ , the proof is complete.  $\square$

This lemma is used to prove necessity of inequality (3.14) in Theorem 2. Note that in certain cases the sum in (3.14) is infinite (division by  $\sigma_m = 0$ ), implying that the interval for von Neumann stability is empty. This situation is discussed in more detail later on. We wish to emphasize that the proof of this theorem is inspired by the proof of the stability theorem in [5], which also uses the result of Lemma 1.

**THEOREM 2** *Suppose  $\epsilon_1, \epsilon_2, \epsilon_3 \geq 0$  and  $\epsilon_1 + \epsilon_2 > 0$ . Then we have von Neumann stability if and only if*

$$\sum_{m=1}^3 \frac{c_m^2}{2\sigma_m/\sigma} \leq 1. \quad (34)$$

**PROOF.** Because  $\sigma > 0$ , the first reduced polynomial  $f_1$  does not vanish so that case (i) of Th. 6.1 from [6] applies, similar as in the second part of the proof of Theorem 1 above. Hence,  $|\xi_1|, |\xi_2| \leq 1$ , if and only if inequalities (3.11) and (3.12) are true. We immediately conclude that inequality (3.11) is unconditionally true, because  $\sigma > 0$  and  $\sigma_3 \geq 0$ . So our task is to check inequality (3.12). Denote

$$\begin{aligned} \sigma^* &= \sigma + 2\sigma_3 - 2\sigma_3 \cos \theta_3, \\ \sigma_m^* &= 2\sigma_m / \sigma^*, \quad m = 1, 2, \\ c_1^* &= c_1, \quad c_2^* = c_2, \quad c_3^* = c_3 \sum_{m=1,2} \sigma_m^* \cos \theta_m. \end{aligned}$$

Inequality (3.12) is equivalent to  $|\mu| \leq 1$ , where

$$\mu = \frac{\sigma}{\sigma^*} - \sum_{m=1}^2 \sigma_m^* (1 - \cos \theta_m) - \sum_{m=1}^3 I c_m^* \sin \theta_m. \quad (35)$$

Introduce the new diffusion parameter  $\sigma_3^*$  by writing

$$\frac{\sigma}{\sigma^*} = 1 - \sigma_3^* (1 - \cos \theta_3), \quad (36)$$

which implies the same expression as for  $\sigma_1^*$  and  $\sigma_2^*$ ,

$$\sigma_3^* = \frac{2\sigma_3}{\sigma^*}. \quad (37)$$

Note that for zero phase angle  $\theta_3$  the definition of  $\sigma_3^*$  through (3.16) is meaningless. However, from the limiting case

$$\sigma^* = \sigma + \sigma_3\theta_3^2 + O(\theta_3^4), \quad \theta_3 \rightarrow 0$$

it follows, by substitution of (3.17) into (3.16), that expression (3.17) is also valid for  $\theta_3 = 0$ . Hence, for all phase angles we can write

$$\mu = 1 - \sum_{m=1}^3 \sigma_m^* (1 - \cos \theta_m) - \sum_{m=1}^3 I c_m^* \sin \theta_m, \quad (38)$$

so that inequality (3.12) is true if and only if

$$|\mu|^2 = \left( 1 - \sum_{m=1}^3 \sigma_m^* (1 - \cos \theta_m) \right)^2 + \left( \sum_{m=1}^3 c_m^* \sin \theta_m \right)^2 \leq 1. \quad (39)$$

Our task is now to prove that (3.14) is necessary and sufficient for (3.19). We will first establish necessity of (3.14). Consider the limiting case:  $\theta_m \rightarrow 0$  with  $|\theta_m| \leq \theta$  for  $m = 1, 2, 3$ . For  $\theta_3 \rightarrow 0$  we have

$$\sigma_m^* = \frac{2\sigma_m}{\sigma} + O(\theta_3^2) \text{ for } m = 1, 2, 3 \text{ and } c_3^* = c_3 + O(\theta^2),$$

so that in the limiting case  $|\mu|^2$  satisfies

$$|\mu|^2 = 1 - \sum_{m=1}^3 \frac{2\sigma_m}{\sigma} \theta_m^2 + \left( \sum_{m=1}^3 c_m \theta_m \right)^2 + O(\theta^4). \quad (40)$$

Set  $\alpha_m = 2\sigma_m/\sigma$ . Because  $\sigma > 0$ , we have  $\alpha_m \geq 0$  for  $m = 1, 2, 3$  and application of Lemma 1 immediately reveals the necessity of (3.14). In particular, if a  $\alpha_m = 0$ , then the corresponding  $c_m$  must be zero too, which means that the dimension is dropped. Hence, in the sufficiency part of the proof we will assume that all  $\alpha_m$  are positive and observe that for a lower dimension the proof of sufficiency goes entirely similar.

To prove sufficiency of (3.14) we proceed as follows. Write

$$\begin{aligned} \sum_{m=1}^3 c_m^* \sin \theta_m &= \sum_{m=1}^2 \frac{c_m}{\sqrt{\alpha_m}} \sqrt{\alpha_m} \sin \theta_m + \\ &\frac{c_3}{\sqrt{\alpha_3}} \sqrt{\alpha_3} \left( \sum_{m=1}^2 \sigma_m^* \cos \theta_m \right) \sin \theta_3. \end{aligned} \quad (41)$$

The Cauchy-Schwarz inequality then yields

$$\left( \sum_{m=1}^3 c_m^* \sin \theta_m \right)^2 \leq \left( \sum_{m=1}^3 \frac{c_m^2}{\alpha_m} \right) \left( \sum_{m=1}^2 \alpha_m \sin^2 \theta_m + \alpha_3 \left( \sum_{m=1}^2 \sigma_m^* \cos \theta_m \right)^2 \sin^2 \theta_3 \right). \quad (42)$$

Set  $y_m = \cos \theta_m$  and invoke (3.14). Using  $\alpha_1 + \alpha_2 = 1$ , we then can write

$$\left( \sum_{m=1}^3 c_m^* \sin \theta_m \right)^2 \leq 1 - \alpha_1 y_1^2 - \alpha_2 y_2^2 + \alpha_3 (\sigma_1^* y_1 + \sigma_2^* y_2)^2 (1 - y_3^2). \quad (43)$$

Further, using  $\sigma^* = \sigma + 2\sigma_3(1 - y_3)$ , we have

$$\left( 1 - \sum_{m=1}^3 \sigma_m^* (1 - \cos \theta_m) \right)^2 = \frac{1}{\sigma^{*2}} (2\sigma_1 y_1 + 2\sigma_2 y_2)^2, \quad (44)$$

so that there remains to prove

$$\begin{aligned} |\mu|^2 &\leq 1 + \frac{1}{\sigma^{*2}} (2\sigma_1 y_1 + 2\sigma_2 y_2)^2 + \alpha_3 \\ &(\sigma_1^* y_1 + \sigma_2^* y_2)^2 (1 - y_3)^2 - \alpha_1 y_1^2 - \alpha_2 y_2^2 \leq 1 \end{aligned} \quad (45)$$

for all  $y_m \in [-1, 1]$ ,  $m = 1, 2, 3$ . Define  $\vec{y} = (y_1, y_2)^T$  and  $Y = \alpha_3(1 - y_3^2)$ . Then the second inequality can be rewritten as

$$\vec{y}^T A \vec{y} \leq 0, \quad (46)$$

where  $A$  is a symmetric two-by-two matrix with the entries

$$\begin{aligned} A_{11} &= \frac{4(Y+1)}{\sigma^{*2}} \sigma_1^2 - \frac{2\sigma_1}{\sigma}, & A_{12} &= \frac{4(Y+1)}{\sigma^{*2}} \sigma_1 \sigma_2, \\ A_{22} &= \frac{4(Y+1)}{\sigma^{*2}} \sigma_2^2 - \frac{2\sigma_2}{\sigma}. \end{aligned} \quad (47)$$

Note that the entries do depend on  $y_3$ , but not on  $\vec{y}$ . Hence, it is sufficient that  $A$  is non-positive definite for all  $y_3 \in [-1, 1]$ . Because  $A_{12} > 0$ ,  $A$  is non-positive definite if

$$A_{11} + A_{12} \leq 0 \quad \text{and} \quad A_{22} + A_{12} \leq 0.$$

A trivial calculation shows that this is indeed the case for all  $y_3 \in [-1, 1]$ , which completes the proof of the theorem.  $\square$

Any case covered by Theorem 2 involves the Du Fort-Frankel scheme in (2.14) since  $\sigma > 0$ . We emphasize that this gives rise to curious and unexpected

stability results. Substitution of  $\sigma_m, c_m$  in (3.14) shows that the critical stepsize for von Neumann stability in all cases covered by Theorem 2 is determined by

$$\tau^2 \left( \sum_{m=1}^3 \frac{q_m^2}{\epsilon_m} \sum_{l=1}^2 \frac{\epsilon_l}{h_l^2} \right) \leq 1. \quad (48)$$

First, we see that the vertical meshwidth  $h_3$  is absent, which is advantageous as we explained in the discussion of Theorem 1. Second, for zero velocities (the pure diffusion case) we have unconditional stability, which is in complete agreement with the unconditional stability of the Du Fort–Frankel scheme (2.19) and the Crank–Nicholson scheme (2.20). However, if a velocity is not zero, then the corresponding diffusion parameter plays a role. Surprisingly, the critical stepsize determined by (3.28) is generally smaller than the one determined by the CFL condition (3.13) and in fact can be zero.

To see this, let us first suppose that  $\epsilon_1, \epsilon_2, \epsilon_3$  are positive. Application of the Cauchy–Schwarz inequality to the CFL condition (3.13) then leads to (3.28) as follows,

$$\begin{aligned} \left( \sum_{l=1}^2 \frac{\tau |q_l|}{h_l} \right)^2 &= \left( \sum_{l=1}^2 \frac{\tau |q_l| \sqrt{\epsilon_l}}{h_l \sqrt{\epsilon_l}} \right)^2 \leq \\ \sum_{l=1}^2 \frac{\tau^2 q_l^2}{\epsilon_l} \sum_{l=1}^2 \frac{\epsilon_l}{h_l^2} &\sum_{m=1}^3 \frac{\tau^2 q_m^2}{\epsilon_m} \sum_{l=1}^2 \frac{\epsilon_l}{h_l^2} \leq 1. \end{aligned} \quad (49)$$

Generally (3.28) appears to be more restrictive, implying a smaller critical stepsize. We consider this curious because it means, for example, that adding artificial diffusion to the advection problem can have a destabilizing effect for the time integration, rather than working out stabilizing. A similar curious situation has been observed earlier in [10, 14]. Also note that if the three diffusion parameters are equal, then they cancel out in (3.28) so that the critical stepsize then even is independent of the diffusion, but yet smaller than in the case of the CFL condition. Of course, the difference between the two conditions is minor if

$$\frac{|q_1| h_1}{\epsilon_1} \approx \frac{|q_2| h_2}{\epsilon_2} \quad \text{and} \quad \frac{q_3^2}{\epsilon_3} \ll \min \left( \frac{q_1^2}{\epsilon_1}, \frac{q_2^2}{\epsilon_2} \right). \quad (50)$$

The observation that for cases covered by Theorem 2 the critical stepsize can even be zero, follows directly from inspection of (3.28). For example, if we take  $q_1, q_2, q_3 \neq 0$ ,  $\epsilon_1, \epsilon_2$  fixed and  $\epsilon_3 \rightarrow 0$ , then  $\tau \rightarrow 0$  when satisfying the stability inequality. By also taking into account Theorem 1, we thus can formulate:

**THEOREM 3** *For von Neumann stability it is necessary that either both  $\epsilon_1$  and  $\epsilon_2$  are zero or positive and if they are both positive, then it is required to have  $\epsilon_3 > 0$  too.*

#### 4 THE DU FORT-FRANKEL DEFICIENCY

We will further explain this curious stability result by relating it with the well-known Du Fort-Frankel deficiency, which describes the situation that for parabolic problems this method is only conditionally convergent, in spite of its unconditional stability (see [9], Sect.7.5).

The necessity of (3.14) or (3.28) has been established from the asymptotic relation (3.20) where all three phase angles  $\theta_m \rightarrow 0$ . This suggests to compute for this limiting case the maximum of the absolute value of the two amplification factors  $\xi_1, \xi_2$  directly from the polynomial (3.2). Denote  $\xi_{max} = \max(|\xi_1|, |\xi_2|)$ . An elementary calculation then yields

$$\xi_{max} = 1 - \sum_{m=1}^3 \sigma_m \theta_m^2 + \frac{1}{2} \sigma \left( \sum_{m=1}^3 c_m \theta_m \right)^2 + O(\theta^3). \quad (51)$$

Indeed, use of Lemma 1 shows again the necessity of (3.14). However, expression (4.1) also reveals a link with the aforementioned convergence deficiency. To see this, consider the modified equation for scheme (2.14) (cf. [9], Sect. 7.5),

$$u_t + q_1 u_x + q_2 u_y + q_3 u_z = \epsilon_1 u_{xx} + \epsilon_2 u_{yy} + \epsilon_3 u_{zz} - \frac{1}{2} \sigma \tau u_{tt}. \quad (52)$$

This modified equation shows the convergence deficiency through the additional term  $-\frac{1}{2} \sigma \tau u_{tt}$ . To establish the link between our stability deficiency and the convergence deficiency, it now suffices to substitute a Fourier mode into (4.2) and to compute the associated continuous amplification factor for vanishing phase angles, similar as we did in the derivation of (4.1). We then find that the continuous amplification factor just equals (4.1), up to  $O(\theta^3)$ . Further, it then follows that the term which causes the instability, that is,

$$\frac{1}{2} \sigma \left( \sum_{m=1}^3 c_m \theta_m \right)^2, \quad (53)$$

originates from the deficiency term  $-\frac{1}{2} \sigma \tau u_{tt}$ , although this term itself is independent of the velocities  $c_m$ . This means that also the modified equation is unstable if (3.14) is violated, in the sense that it admits growing Fourier modes in the low frequency range. This obviously implies that this then also must happen for scheme (2.14) when subjected to the von Neumann stability test.

Noteworthy is that if we bound the phase angles from below, say  $\theta_m \geq \theta_0 > 0$ , that then an interval  $0 < \tau \leq \tau_0$  exists for which the amplification factors  $\xi_1, \xi_2$  are strictly less than one. This follows from expression (3.18), since its real part is independent of  $\tau$  and can be made  $< 1$  by taking  $\theta_0$  sufficiently small, while the imaginary part can be made sufficiently small by taking  $\tau_0$  small enough. Hence, if we consider a fixed grid, then we can always achieve stability, but of course  $\tau_0$  becomes smaller if the grid is refined.

## 5 PRACTICAL CONSIDERATIONS

Strict von Neumann stability is known to have great practical relevance. There is no doubt that the von Neumann method is the best single technique (cf. [5]) for finding necessary conditions for stability if we are in a non-model situation, which in practice of course always happens. In this connection a natural question is, how bad actually is the stability deficiency for the OELH scheme. In other words, should we in practice consider the CFL condition (3.13) as a 'practical restriction', or should we take the more stringent condition (3.28) really serious.

Let  $\tau_{cfl}$  and  $\tau_{(3.28)}$  denote the critical stepsizes. Because the necessity of condition (3.14) shows up in the limiting case  $\theta_m \rightarrow 0$ , the maximum  $\xi_{max}$  as derived in (4.1) will be only marginally larger than one if  $\tau_{(3.28)} < \tau \leq \tau_{cfl}$ . However, there is a possibility that other critical combinations of phase angles exist, away from zero, which also lead to (3.14). Therefore we have computed approximate values of  $\xi_{max}$  (the maximum taken over all discrete  $\theta$ -values) as a function of  $\tau$  for several choices of  $\epsilon_m, q_m, h_m$ . We indeed observed other critical  $\theta$ -combinations away from zero. Yet, in all tests  $\xi_{max}$  appeared to become only marginally larger than one in the stepsize range  $\tau_{(3.28)} < \tau \leq \tau_{cfl}$ , similar as in the limiting case which led to (3.14).

Figure 1 shows a plot of  $\xi_{max}(\tau)$  which is characteristic for the tests considered. We see that the overshoot due to violating (3.28) is practically insignificant. In the interval  $\tau_{(3.28)} < \tau \leq \tau_{cfl}$  the overshoot of  $\xi_{max}(\tau)$  is  $\leq 0.001$ . However, as expected, we also see that  $\tau > \tau_{cfl}$  will quickly result in severe instability. The fact that the CFL condition should be satisfied in general, thus also in all cases covered by Theorem 2, can be understood by computing (3.18) for special choices of the  $\theta_m$ . For example, for  $\theta_m = \frac{\pi}{2}, m = 1, 2, 3$ , we get

$$\mu = 1 - \sum_{m=1}^3 \sigma_m^* - \sum_{m=1}^3 I c_m^* = -I(c_1 + c_2), \quad (54)$$

which trivially yields the CFL condition (3.13) for positive  $c_1, c_2$  (cf. (3.19)).

We conclude that the more stringent condition (3.28) is only a theoretical curiosity. For the actual practice it will be of little importance since the instability that will occur by violation is so small that it will not be observed in actual computation, of course as long as the CFL condition (3.13) is satisfied. This condition is highly relevant for the actual practice and should always be obeyed. On the other hand, violation of (3.28) will only be noticeable after an unrealistically large number of time steps. To illustrate this in actual integration, we applied the OELH integrator to the model equation (2.1), discretized on a uniform 40x40x10 grid, using periodic boundary conditions. The parameters in this experiment were set to the same values as in Figure 1 and the grid sizes to  $(h_1, h_2, h_3) = (500, 500, 10)$ . These values yield

$$\tau_{cfl} = 100.0, \quad \tau_{(3.28)} = 37.7.$$



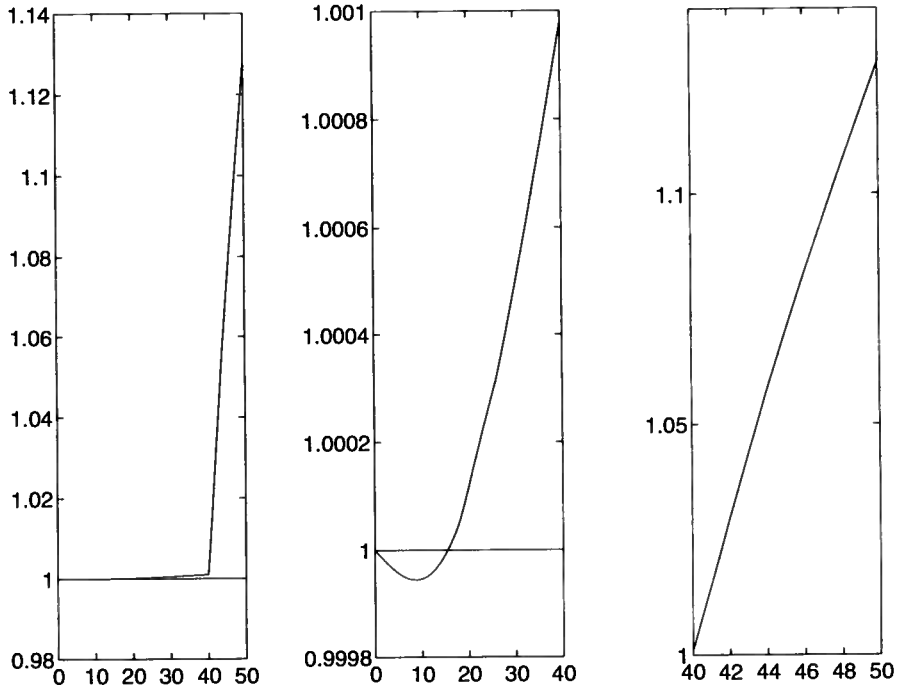


FIGURE 1. Plots of  $\xi_{max}(\tau)$  for the parameters  $(\epsilon_1, \epsilon_2, \epsilon_3) = (1.0, 0.5, 0.01)$ ,  $(q_1, q_2, q_3) = (3, 2, 1)$ . The grid sizes are  $(h_1, h_2, h_3) = (200, 200, 1)$ . This yields  $\tau_{(3,28)} \approx 15.1$  and  $\tau_{cfl} = 40.0$ . The left plot covers the  $\tau$ -interval  $0 \leq \tau \leq 50$ , the middle plot  $0 \leq \tau \leq \tau_{cfl}$  and the right plot  $\tau_{cfl} \leq \tau \leq 50$ . The middle and the right plot show a finer scale in the vertical.

Obviously,  $u \equiv 1$  is an exact solution for the test model. To study the long-term stability behaviour of the OELH method, we slightly perturbed the initial condition to  $u(x, y, z) = 1.0 + \delta g(x, y, z)$ , with  $g$  a smooth function with maximum modulus equal to 1.0 and  $\delta = 10^{-5}$ . Table 1 contains the values of the experimental amplification factors

$$\delta^{-1} \max_i |U_i^N - 1| \tag{55}$$

for various values of  $\tau$  and  $N$ . Here  $U_i^N$  denotes the numerical solution at grid point  $i$  after  $N$  steps of length  $\tau$ . The results are self evident. Violation of the CFL condition is disastrous, whereas violation of (3.28) leads to error growth, but only destroys the solution after an unrealistically large number of time steps.

	$\tau = 37$	$\tau = 100$	$\tau = 100.1$
$N = 10^4$	0.724	3.68	$10^{185}$
$N = 10^5$	0.497	870	
$N = 5 \cdot 10^5$	0.362	$10^{20}$	

Table 1: Experimental amplification factors (5.2).

Finally, it is also of interest to recall the convergence deficiency, from which the OELH scheme also suffers. Presumably, this convergence deficiency is also of little relevance for the shallow water transport application. In this application the regular temporal and spatial truncation errors are expected to be larger than the error induced by the parasitic, non-physical term  $\frac{1}{2}\sigma\tau u_{tt}$ . For example, in the experiments reported in [12, 13] this error plays no role. Experiments where this error is shown, though, can be found in [14].

#### REFERENCES

1. A.R. Gourlay (1970). *Hopscotch: A Fast Second Order Partial Differential Equation Solver*. J. Inst. Math. Appl., 6, 375 - 390.
2. A.R. Gourlay (1971). *Some Recent Methods for the Numerical Solution of Time-Dependent Partial Differential Equations*. Proc. Roy. Soc. London A 323, 219 - 235.
3. A.R. Gourlay, J. LI. Morris (1972). *Hopscotch Difference Methods for Non-linear Hyperbolic Systems*. IBM J. Res. Develop., 16, 349 - 353.
4. A.R. Gourlay (1977). *Splitting Methods for Time-Dependent Partial Differential Equations*. In: D. Jacobs, ed., *The State of the Art in Numerical Analysis*, Academic Press, 757 - 791.
5. A.C. Hindmarsh, P.M. Gresho, D.F. Griffiths (1984). *The Stability of Explicit Euler Time-Integration for Certain Finite-Difference Approximations of the Multi-Dimensional Advection-Diffusion Equation*. Int. J. Numer. Meth. in Fluids, 4, 853 - 897.
6. J.J.H. Miller (1971). *On the Location of Zeros of Certain Classes of Polynomials with Applications to Numerical Analysis*. J. Inst. Maths. Applics., 8, 397 - 406.

7. K.W. Morton (1980). *Stability of Finite Difference Approximations to a Diffusion-Convection Equation*. Int. J. Numer. Meth. in Engr. 15, 677 - 683.
8. J. Von Neumann, R.D. Richtmyer (1950). *A Method for the Numerical Calculations of Hydrodynamical Shocks*. J. Appl. Phys. 21, 232 - 237.
9. R.D. Richtmyer, K.W. Morton (1967). *Difference Methods for Initial Value Problems*. Interscience Publishers, New York.
10. U. Schumann (1975). *Linear Stability of Finite-Difference Equations for Three-Dimensional Flow Problems*. J. Comput. Phys., 18, 465 - 470.
11. J. Schur (1918). *Über Potenzreihen, die im Innern des Einheitskreises beschränkt sind*. J. Reine u. angew. Math. 147, 205 - 232.
12. B.P. Sommeijer, P.J. van der Houwen, J. Kok (1993). *Time Integration of Three-Dimensional Numerical Transport Models*. Report NM-R9316, Centre for Mathematics and Computer Science, Amsterdam (to appear in Appl. Numer. Math.).
13. B.P. Sommeijer, J. Kok (1994). *Implementation and Performance of a Three-Dimensional Numerical Transport Model*. Report NM-R9402, Centre for Mathematics and Computer Science, Amsterdam (to appear in Int. J. Numer. Meth. in Fluids).
14. J.H.M. ten Thije Boonkamp, J.G. Verwer (1987). *On the Odd-Even Hopscotch Scheme for the Numerical Integration of Time-Dependent Partial Differential Equations*. Appl. Numer. Math., 3, 183 - 193.



# Randomness

Paul Vitányi<sup>1</sup>

CWI and Universiteit van Amsterdam

These draft excerpts of the chapter “*Randomness*” in *20th Century Mathematics* in preparation for the ‘*Matematica, Logica, Informatica*’ Volume 12 of the *Storia del XX Secolo*, to be published by the *Instituto della Enciclopedia Italiana*, are dedicated to Cor Baayen. Here we present in a single essay a combination and completion of the several aspects of the problem of randomness of individual objects which of necessity occur scattered in our text [3].

## CONTENTS

<b>1 Introduction</b>	<b>627</b>
<b>2 Randomness as Unpredictability</b>	<b>631</b>
<b>3 Randomness in Terms of Expectations</b>	<b>636</b>
<b>4 Randomness as Incompressibility</b>	<b>639</b>

## 1 INTRODUCTION

P.S. Laplace (1749 – 1827) has pointed out the following reason why intuitively a regular outcome of a random event is unlikely.

“We arrange in our thought all possible events in various classes; and we regard as *extraordinary* those classes which include a very small number. In the game of heads and tails, if head comes up a hundred times in a row then this appears to us extraordinary, because the almost infinite number of combinations that can arise in a hundred throws are divided in regular sequences, or those in which we observe a rule that is easy to grasp, and in irregular sequences, that are incomparably more numerous”.

---

<sup>1</sup>Partially supported by the European Union through NeuroCOLT ESPRIT Working Group Nr. 8556, and by NWO through NFI Project ALADDIN under Contract number NF 62-376. Address: CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands. Email: paulv@cwi.nl

If by 'regularity' we mean that the complexity is significantly less than maximal, then the number of all regular events is small (because by simple counting the number of different objects of low complexity is small). Therefore, the event that anyone of them occurs has small probability (in the uniform distribution). Yet, the classical calculus of probabilities tells us that 100 heads are just as probable as any other sequence of heads and tails, even though our intuition tells us that it is less 'random' than some others. Listen to the redoubtable Dr. Samuel Johnson:

"Dr. Beattie observed, as something remarkable which had happened to him, that he chanced to see both the No. 1 and the No. 1000, of the hackney-coaches, the first and the last; 'Why, Sir', said Johnson, 'there is an equal chance for one's seeing those two numbers as any other two.' He was clearly right; yet the seeing of two extremes, each of which is in some degree more conspicuous than the rest, could not but strike one in a stronger manner than the sight of any other two numbers." [Boswell's *Life of Johnson*]

Laplace distinguishes between the object itself and a cause of the object.

"The regular combinations occur more rarely only because they are less numerous. If we seek a cause wherever we perceive symmetry, it is not that we regard the symmetrical event as less possible than the others, but, since this event ought to be the effect of a regular cause or that of chance, the first of these suppositions is more probable than the second. On a table we see letters arranged in this order C o r B a a y e n, and we judge that this arrangement is not the result of chance, not because it is less possible than others, for if this word were not employed in any language we would not suspect it came from any particular cause, but this word being in use among us, it is incomparably more probable that some person has thus arranged the aforesaid letters than that this arrangement is due to chance." [Slightly paraphrasing Laplace]

Let us try to turn Laplace's argument into a formal one. First we introduce some notation. If  $x$  is a finite binary sequence, then  $l(x)$  denotes the *length* (number of occurrences of binary digits) in  $x$ . For example,  $l(010) = 3$ .

### *Occam's Razor*

Suppose we observe a binary string  $x$  of length  $l(x) = n$  and want to know whether we must attribute the occurrence of  $x$  to pure chance or to a cause. To put things in a mathematical framework, we define *chance* to mean that the literal  $x$  is produced by independent tosses of a fair coin. More subtle is the interpretation of *cause* as meaning that the computer on our desk computes  $x$  from a program provided by independent tosses of a fair coin. The chance of generating  $x$  literally is about  $2^{-n}$ . But the chance of generating  $x$  in the form of a short program  $x^*$ , the cause from which our computer computes  $x$ , is at least  $2^{-l(x^*)}$ . In other words, if  $x$  is regular, then  $l(x^*) \ll n$ , and it is about  $2^{n-l(x^*)}$  times more likely that  $x$  arose as the result of computation from some simple cause (like a short program  $x^*$ ) than literally by a random process.

This approach will lead to an objective and absolute version of the classic maxim of William of Ockham (1290? – 1349?), known as Occam’s razor: “if there are alternative explanations for a phenomenon, then, all other things being equal, we should select the simplest one”. One identifies ‘simplicity of an object’ with ‘an object having a short effective description’. In other words, *a priori* we consider objects with short descriptions more likely than objects with only long descriptions. That is, objects with low complexity have high probability while objects with high complexity have low probability.

This principle is intimately related with problems in both probability theory and information theory. These problems as outlined below can be interpreted as saying that the related disciplines are not ‘tight’ enough; they leave things unspecified which our intuition tells us should be dealt with.

### *Lacuna of Classical Probability Theory*

An adversary claims to have a true random coin and invites us to bet on the outcome. The coin produces a hundred heads in a row. We say that the coin cannot be fair. The adversary, however, appeals to probability theory which says that each sequence of outcomes of a hundred coin flips is equally likely,  $1/2^{100}$ , and one sequence had to come up.

Probability theory gives us no basis to challenge an outcome *after* it has happened. We could only exclude unfairness in advance by putting a penalty side-bet on an outcome of 100 heads. But what about 1010...? What about an initial segment of the binary expansion of  $\pi$ ?

#### **Regular sequence**

$$\Pr(000000000000000000000000) = \frac{1}{2^{26}}$$

#### **Regular sequence**

$$\Pr(01000110110000010100111001) = \frac{1}{2^{26}}$$

#### **Random sequence**

$$\Pr(10010011011000111011010000) = \frac{1}{2^{26}}$$

The first sequence is regular, but what is the distinction of the second sequence and the third? The third sequence was generated by flipping a quarter. The second sequence is very regular: 0, 1, 00, 01, ... The third sequence will pass (pseudo-)randomness tests.

In fact, classical probability theory cannot express the notion of *randomness of an individual sequence*. It can only express expectations of properties of outcomes of random processes, that is, the expectations of properties of the total set of sequences under some distribution.

Only relatively recently, this problem has found a satisfactory resolution by combining notions of computability and statistics to express the complexity of a finite object. This complexity is the length of the shortest binary program from which the object can be effectively reconstructed. It may be called the *algorithmic information content* of the object. This quantity turns out to be an attribute of the object alone, and absolute (in the technical sense of being recursively invariant). It is the *Kolmogorov complexity* of the object.

### *Lacuna of Information Theory*

Claude Shannon's classical information theory assigns a quantity of information to an ensemble of possible messages. All messages in the ensemble being equally probable, this quantity is the number of bits needed to count all possibilities.

This expresses the fact that each message in the ensemble can be communicated using this number of bits. However, it does not say anything about the number of bits needed to convey any individual message in the ensemble. To illustrate this, consider the ensemble consisting of all binary strings of length 9999999999999999.

By Shannon's measure, we require 9999999999999999 bits on the average to encode a string in such an ensemble. However, the string consisting of 9999999999999999 1's can be encoded in about 55 bits by expressing 999999999999999 in binary and adding the repeated pattern '1'. A requirement for this to work is that we have agreed on an algorithm that decodes the encoded string. We can compress the string still further when we note that 9999999999999999 equals  $3^2 \times 1111111111111111$ , and that 1111111111111111 consists of  $2^4$  1's.

Thus, we have discovered an interesting phenomenon: the description of some strings can be compressed considerably, provided they exhibit enough regularity. This observation, of course, is the basis of all systems to express very large numbers and was exploited early on by Archimedes in his treatise *The Sand Reckoner*, in which he proposes a system to name very large numbers:

"There are some, King Golon, who think that the number of sand is infinite in multitude [...or] that no number has been named which is great enough to exceed its multitude. [...] But I will try to show you, by geometrical proofs, which you will be able to follow, that, of the numbers named by me [...] some exceed not only the mass of sand equal in magnitude to the earth filled up in the way described, but also that of a mass equal in magnitude to the universe."

However, if regularity is lacking, it becomes more cumbersome to express large numbers. For instance, it seems easier to compress the number 'one billion,' than the number 'one billion seven hundred thirty-five million two hundred sixty-eight thousand and three hundred ninety-four,' even though they are of the same order of magnitude.

The above example shows that we need too many bits to transmit regular objects. The converse problem, too little bits, arises as well since Shannon's theory of information and communication deals with the specific technology problem of data transmission. That is, with the information that needs to be



transmitted in order to select an object from a previously agreed upon set of alternatives; agreed upon by both the sender and the receiver of the message. If we have an ensemble consisting of the *Odyssey* and the sentence "let's go drink a beer" then we can transmit the *Odyssey* using only one bit. Yet Greeks feel that Homer's book has more information contents. Our task is to widen the limited set of alternatives until it is universal. We aim at a notion of 'absolute' information of individual objects, which is the information which by itself describes the object completely.

Formulation of these considerations in an objective manner leads again to the notion of shortest programs and Kolmogorov complexity.

## 2 RANDOMNESS AS UNPREDICTABILITY

What is the proper definition of a random sequence, the 'lacuna in probability theory' we have identified above? Let us consider how mathematicians test randomness of individual sequences. To measure randomness, criteria have been developed which certify this quality. Yet, in recognition that they do not measure 'true' randomness, we call these criteria 'pseudo' randomness tests. For instance, statistical survey of initial segments of the sequence of decimal digits of  $\pi$  have failed to disclose any significant deviations of randomness. But clearly, this sequence is so regular that it can be described by a simple program to compute it, and this program can be expressed in a few bits.

"Any one who considers arithmetical methods of producing random digits is, of course, in a state of sin. For, as has been pointed out several times, there is no such thing as a random number—there are only methods to produce random numbers, and a strict arithmetical procedure is of course not such a method. (It is true that a problem we suspect of being solvable by random methods may be solvable by some rigorously defined sequence, but this is a deeper mathematical question than we can go into now.)" [von Neumann]

This fact prompts more sophisticated definitions of randomness. In his famous address to the International Mathematical Congress in 1900, D. Hilbert proposed twenty-three mathematical problems as a program to direct the mathematical efforts in the twentieth century. The 6th problem asks for "To treat (in the same manner as geometry) by means of axioms, those physical sciences in which mathematics plays an important part; in the first rank are the theory of probability ..". Thus, Hilbert views probability theory as a physical applied theory. This raises the question about the properties one can expect from typical outcomes of physical random sources, which *a priori* has no relation whatsoever with an axiomatic mathematical theory of probabilities. That is, a mathematical system has no direct relation with physical reality. To obtain a mathematical system that is an appropriate model of physical phenomena one needs to identify and codify essential properties of the phenomena under consideration by empirical observations.

Notably Richard von Mises (1883-1953) proposed notions that approach the very essence of true randomness of physical phenomena. This is related with

the construction of a formal mathematical theory of probability, to form a basis for real applications, in the early part of this century. While von Mises' objective was to justify the applications to the real phenomena, A.N. Kolmogorov's (1903-1987) classic 1933 treatment constructs a purely axiomatic theory of probability on the basis of set theoretic axioms.

"This theory was so successful, that the problem of finding the basis of real applications of the results of the mathematical theory of probability became rather secondary to many investigators. ... [however] the basis for the applicability of the results of the mathematical theory of probability to real 'random phenomena' must depend in some form on the *frequency concept of probability*, the unavoidable nature of which has been established by von Mises in a spirited manner." [Kolmogorov]

The point made is that the axioms of probability theory are designed so that abstract probabilities can be computed, but nothing is said about what probability really means, or how the concept can be applied meaningfully to the actual world. Von Mises analyzed this issue in detail, and suggested that a proper definition of probability depends on obtaining a proper definition of a random sequence. This makes him a 'frequentist'—a supporter of the frequency theory.

The frequency theory to interpret probability says, roughly, that if we perform an experiment many times, then the ratio of favorable outcomes to the total number  $n$  of experiments will, *with certainty*, tend to a limit,  $p$  say, as  $n \rightarrow \infty$ . This tells us something about the *meaning* of probability, namely, the measure of the positive outcomes is  $p$ . But suppose we throw a coin 1000 times and wish to know what to expect. Is 1000 enough for convergence to happen? The statement above does not say. So we have to add something about the rate of convergence. But we cannot assert a *certainty* about a particular number of  $n$  throws, such as 'the proportion of heads will be  $p \pm \epsilon$  for large enough  $n$  (with  $\epsilon$  depending on  $n$ )'. We can at best say 'the proportion will lie between  $p \pm \epsilon$  with at least such and such probability (depending on  $\epsilon$  and  $n_0$ ) whenever  $n > n_0$ '. But now we defined probability in an obviously circular fashion.

In 1919 von Mises proposed to eliminate the problem by simply dividing all infinite sequences into special random sequences (called *collectives*), having relative frequency limits, which are the proper subject of the calculus of probabilities and other sequences. He postulates the existence of random sequences (thereby circumventing circularity) as certified by abundant empirical evidence, in the manner of physical laws and derives mathematical laws of probability as a consequence. In his view a naturally occurring sequence can be nonrandom or unlawful in the sense that it is not a proper collective.

Von Mises views the theory of probabilities insofar as they are numerically representable as a physical theory of definitely observable phenomena, repetitive or mass events, for instance, as found in games of chance, population statistics, Brownian motion. 'Probability' is a primitive notion of the theory comparable to those of 'energy' or 'mass' in other physical theories.

Whereas energy or mass exist in fields or material objects, probabilities exist only in the similarly mathematical idealization of collectives (random sequences). All problems of the theory of probability consist of deriving, according to certain rules, new collectives from given ones and calculating the distributions of these new collectives. The exact formulation of the properties of the collectives is secondary and must be based on empirical evidence. These properties are the existence of a limiting relative frequency and randomness.

The property of randomness is a generalization of the abundant experience in gambling houses, namely, the impossibility of a successful gambling system. Including this principle in the foundation of probability, von Mises argues, we proceed in the same way as the physicists did in the case of the energy principle. Here too, the experience of hunters of fortune is complemented by solid experience of insurance companies and so forth.

A fundamentally different approach is to justify *a posteriori* the application of a purely mathematically constructed theory of probability, such as the theory resulting from the Kolmogorov axioms. Suppose, we can show that the appropriately defined random sequences form a set of measure one, and without exception satisfy all laws of a given axiomatic theory of probability. Then it appears practically justifiable to assume that as a result of an (infinite) experiment only random sequences appear.

Von Mises' notion of infinite random sequence of 0's and 1's (collective) essentially appeals to the idea that no gambler, making a fixed number of wagers of 'heads', at fixed odds [say  $p$  versus  $1 - p$ ] and in fixed amounts, on the flips of a coin [with bias  $p$  versus  $1 - p$ ], can have profit in the long run from betting according to a system instead of betting at random. Says Church: "this definition [below] ... while clear as to general intent, is too inexact in form to serve satisfactorily as the basis of a mathematical theory."

**DEFINITION 1** An infinite sequence  $a_1, a_2, \dots$  of 0's and 1's is a random sequence in the special meaning of *collective* if the following two conditions are satisfied.

1. Let  $f_n$  is the number of 1's among the first  $n$  terms of the sequence. Then

$$\lim_{n \rightarrow \infty} \frac{f_n}{n} = p,$$

for some  $p$ ,  $0 < p < 1$ .

2. A *place-selection rule* is a partial function  $\phi$ , from the finite binary sequences to 0 and 1. It takes the values 0 and 1, for the purpose of selecting one after the other those indices  $n$  for which  $\phi(a_1 a_2 \dots a_{n-1}) = 1$ . We require (1), with the same limit  $p$ , also for every infinite subsequence

$$a_{n_1} a_{n_2} \dots$$

obtained from the sequence by some *admissible* place-selection rule. (We have not yet formally stated which place-selection rules are admissible.)

The existence of a relative frequency limit is a strong assumption. Empirical evidence from long runs of dice throws, in gambling houses, or with death statistics in insurance mathematics, suggests that the relative frequencies are *apparently convergent*. But clearly, no empirical evidence can be given for the existence of a definite limit for the relative frequency. However long the test run, in practice it will always be finite, and whatever the apparent behavior in the observed initial segment of the run, it is always possible that the relative frequencies keep oscillating forever if we continue.

The second condition ensures that no strategy using an admissible place-selection rule can select a subsequence which allows different odds for gambling than a subsequence which is selected by flipping a fair coin. For example, let a casino use a coin with probability  $p = 1/4$  of coming up heads and pay-off heads equal 4 times pay-off tails. This ‘Law of Excluded Gambling Strategy’ says that a gambler betting in fixed amounts cannot make more profit in the long run betting according to a system than from betting at random.

“In everyday language we call random those phenomena where we cannot find a regularity allowing us to predict precisely their results. Generally speaking, there is no ground to believe that random phenomena should possess any definite probability. Therefore, we should distinguish between randomness proper (as absence of any regularity) and stochastic randomness (which is the subject of probability theory). There emerges the problem of finding reasons for the applicability of the mathematical theory of probability to the real world.” [Kolmogorov]

Intuitively, we can distinguish between sequences that are irregular and do not satisfy the regularity implicit in stochastic randomness, and sequences that are irregular but do satisfy the regularities associated with stochastic randomness. Formally, we will distinguish the second type from the first type by whether or not a certain complexity measure of the initial segments goes to a definite limit. The complexity measure referred to is the length of the shortest description of the prefix (in the precise sense of Kolmogorov complexity) divided by its length. It will turn out that almost all infinite strings are irregular of the second type and satisfy all regularities of stochastic randomness.

“In applying probability theory we do not confine ourselves to negating regularity, but from the hypothesis of randomness of the observed phenomena we draw definite positive conclusions.” [Kolmogorov]

Considering the sequence as fair coin tosses with  $p = 1/2$ , the second condition in Definition 1 says there is no *strategy  $\phi$*  (*principle of excluded gambling system*) which assures a player betting at fixed odds and in fixed amounts, on the tosses of the coin, to make infinite gain. That is, no advantage is gained in the long run by following some system, such as betting ‘head’ after each run of seven consecutive tails, or (more plausibly) by placing the  $n$ th bet ‘head’ after the appearance of  $n + 7$  tails in succession. According to von Mises, the above conditions are sufficiently familiar and a uncontroverted empirical generalization to serve as the basis of an applicable calculus of probabilities.

**EXAMPLE 1** It turns out that the naive mathematical approach to a concrete formulation, admitting simply *all* partial functions, comes to grief as follows.

Let  $a = a_1 a_2 \dots$  be any collective. Define  $\phi_1$  as  $\phi_1(a_1 \dots a_{i-1}) = 1$  if  $a_i = 1$ , and undefined otherwise. But then  $p = 1$ . Defining  $\phi_0$  by  $\phi_0(a_1 \dots a_{i-1}) = b_i$ , with  $b_i$  the complement of  $a_i$ , for all  $i$ , we obtain by the second condition of Definition 1 that  $p = 0$ . Consequently, if we allow functions like  $\phi_1$  and  $\phi_0$  as strategy, then von Mises' definition cannot be satisfied at all.  $\diamond$

In the thirties, Abraham Wald proposed to restrict the *a priori* admissible  $\phi$  to any fixed countable set of functions. Then collectives do exist. But which countable set? In 1940, Alonzo Church proposed to choose a set of functions representing 'computable' strategies. According to Church's Thesis, this is precisely the set of *recursive functions*. With recursive  $\phi$ , not only is the definition completely rigorous, and random infinite sequences do exist, but moreover they are abundant since the infinite random sequences with  $p = 1/2$  form a set of measure one. From the existence of random sequences with probability  $1/2$ , the existence of random sequences associated with other probabilities can be derived. Let us call sequences satisfying Definition 1 with recursive  $\phi$  *Mises-Wald-Church random*. That is, the involved *Mises-Wald-Church place-selection rules* consist of the partial recursive functions.

Appeal to a theorem by Wald yields as a corollary that the set of Mises-Wald-Church random sequences associated with any fixed probability has the cardinality of the continuum. Moreover, each Mises-Wald-Church random sequence qualifies as a normal number. (A number is *normal* if each digit of the base, and each block of digits of any length, occurs with equal asymptotic frequency.) Note however, that not every normal number is Mises-Wald-Church random. This follows, for instance, from Champernowne's sequence (or number),

0.1234567891011121314151617181920...

due to D.G. Champernowne, which is normal in the scale of 10 and where the  $i$ th digit is easily calculated from  $i$ . The definition of a Mises-Wald-Church random sequence implies that its consecutive digits cannot be effectively computed. Thus, an existence proof for Mises-Wald-Church random sequences is necessarily nonconstructive. Unfortunately, the von Mises-Wald-Church definition is not yet good enough, as was shown by J. Ville in 1939. There exist sequences that satisfy the Mises-Wald-Church definition of randomness, with limiting relative frequency of ones of  $1/2$ , but nonetheless have the property that

$$\frac{f_n}{n} \geq \frac{1}{2} \text{ for all } n.$$

The probability of such a sequence of outcomes in random flips of a fair coin is zero. Intuition: if you bet '1' all the time against such a sequence of outcomes, then your accumulated gain is always positive! Similarly, other properties of randomness in probability theory such as the Law of the Iterated Logarithm do not follow from the Mises-Wald-Church definition.

### 3 RANDOMNESS IN TERMS OF EXPECTATIONS

For a better understanding of the problem revealed by Ville, and its subsequent solution by P. Martin-Löf in 1966, we look at some aspects of the methodology of probability theory. Consider the sample space of all one-way infinite binary sequences generated by fair coin tosses. Intuitively, we call a sequence ‘random’ iff it is ‘typical’. It is not ‘typical’, say ‘special’, if it has a particular distinguishing property. An example of such a property is that an infinite sequence contains only finitely many ones. There are infinitely many such sequences. But the probability that such a sequence occurs as the outcome of fair coin tosses is zero. ‘Typical’ infinite sequences will have the converse property, namely, they contain infinitely many ones.

In fact, one would like to say that ‘typical’ infinite sequences will have *all converse properties* of the properties which can be enjoyed by ‘special’ infinite sequences. This is formalized as follows. If a particular property, such as containing infinitely many occurrences of ones (or zeros), the Law of Large Numbers, or the Law of the Iterated Logarithm, has been shown to have probability one, then one calls this a *Law of Randomness*.

An infinite sequence is ‘typical’ or ‘random’ if it satisfies all Laws of Randomness. That is, a *particular* ‘random’ infinite sequence possesses all properties which are expected to hold with probability one for the ensemble of *all* infinite sequences. This is the substance of so-called pseudo-randomness tests. For example, to test whether the sequence of digits corresponding to the decimal expansion of  $\pi = 3.1415\dots$  is random one tests whether the initial segment satisfies some properties which hold with probability one for the ensemble of all sequences.

**EXAMPLE 2** One such property is so-called normality. E. Borel (1909) has called an infinite sequence of decimal digits *normal* in the scale of ten if, for each  $k$ , the frequency of occurrences (possibly overlapping) of each block  $y$  of length  $k \geq 1$  in the initial segment of length  $n$  goes to limit  $10^{-k}$  for  $n$  grows unbounded, [1]. It is known that normality is not sufficient for randomness, since Champernowne’s sequence

$$123456789101112\dots$$

is normal in the scale of ten. On the other hand, it is universally agreed that a random infinite sequence must be normal. (If not, then some blocks occur more frequent than others, which can be used to obtain better than fair odds for prediction.)

For a particular binary sequence  $\omega = \omega_1\omega_2\dots$  let  $f_n = \omega_1 + \omega_2 + \dots + \omega_n$ . Of course, we cannot effectively test an infinite sequence. Therefore, a so-called pseudo-randomness test examines increasingly long initial segments of the individual sequence under consideration.

We can define a pseudo randomness test for the normality property with  $k = 1$  to test a candidate infinite sequence for increasing  $n$  whether the deviations

from one half 0's and 1's become too large. For example, by checking for each successive  $n$  whether

$$\left|f_n - \frac{n}{2}\right| > \sqrt{\frac{n \log \log n}{2}}.$$

(The Law of the Iterated Logarithm states that this inequality should not hold for infinitely many  $n$ ). If within  $n$  trials in this process we find that the inequality holds  $k$  times, then we assume the original infinite sequence to be random with confidence at most, say,  $\sum_{i=1}^n 1/2^i - \sum_{i=1}^k 1/2^i$ . (The sequence is random if the confidence is greater than zero for  $n$  goes to infinity, and not random otherwise.)

Clearly, the number of pseudo-randomness tests we can devise is infinite. Namely, just for the normality property alone there is a similar pseudo-randomness test for each  $k \geq 1$ .  $\diamond$

But now we are in trouble. Each individual infinite sequence induces its very own pseudo-randomness test which tests whether a candidate infinite sequence is in fact that individual sequence. Each infinite sequence forms a singleton set in the sample space of all infinite sequences. *All* complements of singleton sets in the sample space have probability one. The intersection of all complements of singleton sets is clearly empty. Therefore, the intersection of all sets of probability one is empty. Thus, there are no random infinite sequences!

Martin-Löf, using ideas related to Kolmogorov complexity, succeeded in defining random infinite sequences in a manner which is free of such difficulties. His starting point is to observe that all laws which are proven in probability theory to hold with probability one are effective. That is, we can effectively test whether a particular infinite sequence does not satisfy a particular Law of Randomness by effectively testing whether the law is violated on increasingly long initial segments of the sequence.

The natural formalization is to identify the effective test with a partial recursive function. This suggests that one ought to consider not the intersection of all sets of measure one, but only the intersection of all sets of measure one with recursively enumerable complements. (Such a complement set is expressed as the union of a recursively enumerable set of cylinders). It turns out that this intersection has again measure one. Hence, almost all infinite sequences satisfy all effective Laws of Randomness with probability one. This notion of infinite random sequences turns out to be related to infinite sequences of which all finite initial segments have high Kolmogorov complexity.

The notion of randomness satisfied by both the Mises-Wald-Church collectives and the Martin-Löf random infinite sequences is roughly that *effective tests* cannot detect regularity. This does not mean that a sequence may not exhibit regularities which cannot be effectively tested. Collectives generated by Nature, as postulated by von Mises, may very well always satisfy stricter criteria of randomness. Why should collectives generated by quantum mechanic phenomena care about mathematical notions of computability? Again, satisfaction of all effectively testable prerequisites for randomness is some form of regularity. Maybe nature is



more lawless than adhering strictly to regularities imposed by the statistics of randomness.

Until now the discussion has centered on infinite random sequences where the randomness is defined in terms of limits of relative frequencies. However,

“The frequency concept based on the notion of *limiting frequency* as the number of trials increases to infinity, does not contribute anything to substantiate the application of the results of probability theory to real practical problems where we always have to deal with a finite number of trials.” [Kolmogorov]

The practical objection against both the relevance of considering infinite sequences of trials and the existence of a relative frequency limit is concisely put in J.M. Keynes’ famous phrase “in the long run we shall all be dead.” It seems more appealing to try to define randomness for finite strings first, and only then define random infinite strings in terms of randomness of initial segments.

The approach of von Mises to define randomness of infinite sequences in terms of *unpredictability* of continuations of finite initial sequences under certain laws (like recursive functions) did not lead to satisfying results. The Martin-Löf approach does lead to satisfying results, and is to a great extent equivalent with the Kolmogorov complexity approach. Although certainly inspired by the random sequence debate, the introduction of Kolmogorov complexity marks a definite shift of point of departure. Namely, to define randomness of sequences by the fact that no program from which an initial segment of the sequence can be computed is significantly shorter than the initial segment itself, rather than that no program can predict the next elements of the sequence. Thus, we change the focus from the ‘unpredictability’ criterion to the ‘incompressibility’ criterion, and since this will turn out to be equivalent with Martin-Löf’s approach, the ‘incompressibility’ criterion is both necessary and sufficient.

Finite sequences which cannot be effectively described in a significant shorter description than their literal representation are called random. Our aim is to characterize random infinite sequences as sequences of which all initial finite segments are random in this sense. Martin-Löf’s related approach characterizes random infinite sequences as sequences of which all initial finite segments pass all effective randomness tests.

Initially, before the idea of complexity, Kolmogorov proposed a close analogy to von Mises’ notions in the finite domain. Consider a generalization of place-selection rules insofar as the selection of  $a_i$  can depend on  $a_j$  with  $j > i$  [A.N. Kolmogorov, *Sankhyā*, Series A, 25(1963), 369-376]. Let  $\Phi$  be a finite set of such generalized place-selection rules. Kolmogorov suggested that an arbitrary finite binary sequence  $a$  of length  $n \geq m$  can be called  $(m, \epsilon)$ -random with respect to  $\Phi$ , if there exists some  $p$  such that the relative frequency of the 1’s in the subsequences  $a_{i_1} \dots a_{i_r}$  with  $r \geq m$ , selected by applying some  $\phi$  in  $\Phi$  to  $a$ , all lie within  $\epsilon$  of  $p$ . (We discard  $\phi$  that yield subsequences shorter than  $m$ .) Stated differently, the relative frequency in this finite subsequence is approximately (to within  $\epsilon$ ) invariant under any of the methods of subsequence selection that yield



subsequences of length at least  $m$ . Kolmogorov has shown that if the cardinality of  $\Phi$  satisfies:

$$d(\Phi) \leq \frac{1}{2} e^{2m\epsilon^2(1-\epsilon)},$$

then, for any  $p$  and any  $n \geq m$  there is some sequence  $a$  of length  $n$  which is  $(m, \epsilon)$ -random with respect to  $\Phi$ .

#### 4 RANDOMNESS AS INCOMPRESSIBILITY

We are to admit no more causes of natural things (as we are told by *Newton*) than such as are both true and sufficient to explain their appearances. This central theme is basic to the pursuit of science, and goes back to the principle known as Occam's razor: "if presented with a choice between indifferent alternatives, then one ought to select the simplest one". Unconsciously or explicitly, informal applications of this principle in science and mathematics abound. The conglomerate of different research threads drawing on an objective and absolute form of this approach appears to be part of an emergent applied science ranking with information theory and probability theory.

Intuitively, the amount of information in a finite string is the size (number of binary digits or *bits*) of the shortest program that, without additional data, computes the string and terminates. A similar definition can be given for infinite strings, but in this case the program produces element after element forever. Thus, a long sequence of 1's such as

$$\underbrace{11111 \dots 1}_{10,000 \text{ times}}$$

contains little information because a program of size about  $\log 10,000$  bits outputs it:

```
for i := 1 to 10,000
  print 1
```

Likewise, the transcendental number  $\pi = 3.1415\dots$ , an infinite sequence of seemingly 'random' decimal digits, contains but a few bits of information. (There is a short program that produces the consecutive digits of  $\pi$  forever.) Such a definition would appear to make the amount of information in a string (or other object) depend on the particular programming language used.

Fortunately, it can be shown that all reasonable choices of programming languages lead to quantification of the amount of 'absolute' information in individual objects that is invariant up to an additive constant. We call this quantity the 'Kolmogorov complexity' of the object. If an object is regular, then it has a shorter description than itself. We call such an object 'compressible'.

More precisely, suppose we want to describe a given object by a finite binary string. We do not care whether the object has many descriptions; however, each description should describe but one object. From among all descriptions

of an object we can take the length of the shortest description as a measure of the object's complexity. It is natural to call an object 'simple' if it has at least one short description, and to call it 'complex' if all of its descriptions are long.

But now we are in danger of falling in the trap so eloquently described in the Richard-Berry paradox, where we define a natural number as "the least natural number that cannot be described in less than twenty words". If this number does exist, we have just described it in thirteen words, contradicting its definitional statement. If such a number does not exist, then all natural numbers can be described in less than twenty words. We need to look very carefully at the notion of 'description'.

Assume that each description describes at most one object. That is, there is a specification method  $D$  which associates at most one object  $x$  with a description  $y$ . This means that  $D$  is a function from the set of descriptions, say  $Y$ , into the set of objects, say  $X$ . It seems also reasonable to require that, for each object  $x$  in  $X$ , there is a description  $y$  in  $Y$  such that  $D(y) = x$ . (Each object has a description.) To make descriptions useful we like them to be finite. This means that there are only countably many descriptions. Since there is a description for each object, there are also only countably many describable objects. How do we measure the complexity of descriptions?

Taking our cue from the theory of computation, we express descriptions as finite sequences of 0's and 1's. In communication technology, if the specification method  $D$  is known to both a sender and a receiver, then a message  $x$  can be transmitted from sender to receiver by transmitting the sequence of 0's and 1's of a description  $y$  with  $D(y) = x$ . The cost of this transmission is measured by the number of occurrences of 0's and 1's in  $y$ , that is, by the length of  $y$ . The least cost of transmission of  $x$  is given by the length of a shortest  $y$  such that  $D(y) = x$ . We choose this least cost of transmission as the 'descriptonal' complexity of  $x$  under specification method  $D$ .

Obviously, this descriptonal complexity of  $x$  depends crucially on  $D$ . The general principle involved is that the syntactic framework of the description language determines the succinctness of description.

In order to objectively compare descriptonal complexities of objects, to be able to say " $x$  is more complex than  $z$ ", the descriptonal complexity of  $x$  should depend on  $x$  alone. This complexity can be viewed as related to a universal description method which is *a priori* assumed by all senders and receivers. This complexity is optimal if no other description method assigns a lower complexity to any object.

We are not really interested in optimality with respect to all description methods. For specifications to be useful at all it is necessary that the mapping from  $y$  to  $D(y)$  can be executed in an effective manner. That is, it can at least in principle be performed by humans or machines. This notion has been formalized as 'partial recursive functions'. According to generally accepted mathematical viewpoints it coincides with the intuitive notion of effective computation.

The set of partial recursive functions contains an optimal function which

minimizes description length of every other such function. We denote this function by  $D_0$ . Namely, for any other recursive function  $D$ , for all objects  $x$ , there is a description  $y$  of  $x$  under  $D_0$  which is shorter than any description  $z$  of  $x$  under  $D$ . (That is, shorter up to an additive constant which is independent of  $x$ .) Complexity with respect to  $D_0$  minorizes the complexities with respect to all partial recursive functions.

We identify the length of the description of  $x$  with respect to a fixed specification function  $D_0$  with the ‘algorithmic (descriptive or Kolmogorov) complexity’ of  $x$ . The optimality of  $D_0$  in the sense above means that the complexity of an object  $x$  is invariant (up to an additive constant independent of  $x$ ) under transition from one optimal specification function to another. Its complexity is an objective attribute of the described object alone: it is an intrinsic property of that object, and it does not depend on the description formalism. This complexity can be viewed as ‘absolute information content’: the amount of information which needs to be transmitted between all senders and receivers when they communicate the message in absence of any other *a priori* knowledge which restricts the domain of the message.

Broadly speaking, this means that all description syntaxes which are powerful enough to express the partial recursive functions are approximately equally succinct. The remarkable usefulness and inherent rightness of the theory of Kolmogorov complexity stems from this independence of the description method. Thus, we have outlined the program for a general theory of algorithmic complexity. The four major innovations are as follows.

1. In restricting ourselves to formally effective descriptions our definition covers every form of description that is intuitively acceptable as being effective according to general viewpoints in mathematics and logics.
2. The restriction to effective descriptions entails that there is a universal description method that minorizes the description length or complexity with respect to any other effective description method. This would not be the case if we considered, say, all noneffective description methods. Significantly, this implies Item 3.
3. The description length or complexity of an object is an intrinsic attribute of the object independent of the particular description method or formalizations thereof.
4. The disturbing Richard-Berry paradox above does not disappear, but resurfaces in the form of an alternative approach to proving Kurt Gödel’s famous result that not every true mathematical statement is provable in mathematics.

#### *Randomness of Individual Sequences Resolved*

The notion of randomness of an infinite sequence in the sense of Martin-Löf, as possessing all effectively testable properties of randomness (one of which

is unpredictability), turns out to be identical with the notion of an infinite sequence having maximal Kolmogorov complexity of all finite initial segments. This equivalence of a single notion being defined by two completely different approaches is a truly remarkable fact. (To be precise, the so-called prefix Kolmogorov complexity of each initial segment of the infinite binary sequence must not decrease more than a fixed constant, depending only on the infinite sequence, below the length of that initial segment, [3].) This property sharply distinguishes the random infinite binary sequences from the nonrandom ones. The set of random infinite binary sequences has uniform measure one. That means that as the outcome from independent flips of a fair coin they occur with probability one.

For finite binary sequences the distinction between randomness and nonrandomness cannot be abrupt, but must be a matter of degree. For example, it would not be reasonable if one string is random but becomes nonrandom if we flip the first nonzero bit. In this context too it has been shown that finite binary sequences which are random in Martin-Löf's sense correspond to those sequences which have Kolmogorov complexity at least their own length. Space limitations forbid a complete treatment of these matters here. Fortunately, it can be found elsewhere, [3].

#### REFERENCES

1. D.E. Knuth, *Seminumerical Algorithms*, Addison-Wesley, 1981.
2. A.N. Kolmogorov, Three approaches to the definition of the concept 'quantity of information', *Problems in Information Transmission*, 1:1(1965), 1-7.
3. M. Li and P.M.B. Vitányi, *An Introduction to Kolmogorov Complexity and its Applications*, Springer-Verlag, New York, 1993.
4. P. Martin-Löf, On the definition of random sequences, *Information and Control*, (1966).
5. R. von Mises, *Probability, Statistics and Truth*, MacMillan, 1939. Reprint: Dover, 1981.
6. C.E. Shannon, A mathematical theory of communication, *Bell System Tech. J.*, 27(1948), 379-423, 623-656.

J.M. Anthonisse  
K.R. Apt  
F. Arbab  
R.T. Baanders  
P.C. Baayen  
R.H. Baayen  
M. Bakker  
H.P. Barendregt  
A. Bensoussan  
J. van Benthem  
J.A. Bergstra  
E.H. Blake  
O.J. Boxma  
A.E. Brouwer  
D.C.A. Bulterman  
A.M. Cohen  
J.-A. Désidéri  
J. van Eijck  
P. van Emde Boas  
L. Fleischhacker  
C. Galindo-Legaria  
P.J.W. ten Hagen  
M. Hazewinkel  
J. Heering  
P.W. Hemker  
I. Herman  
P.J. van der Houwen  
M.S. Keane  
M. Kersten  
P. Klint  
J.W. Klop  
T.H. Koornwinder  
B. Koren  
A.A.M. Kuijk  
M.-H. Lallemand  
R. de Leeuw  
J.K. Lenstra  
W.M. Lioen  
J. van de Lune  
P.C. Marais  
J. van Mill  
H.M. Mulder  
K.S. Mullender  
G.Y. Nieuwland  
A. Overkamp  
A. Pellenkoff  
G. Reynolds  
H.J.J. te Riele  
J. Schipper  
A. Schrijver  
J.H. van Schuppen  
A. Siebes  
B.P. Sommeijer  
N.M. Temme  
F. Teusink  
W.H. Tossijn  
A.S. Troelstra  
F.W. Vaandrager  
J.G. Verwer  
P.M.B. Vitányi  
J.A.J. van Vonderen  
J.W. van der Werf

