Metric Pomset Semantics for a Concurrent Language with Recursion

J.W. de Bakker
J.H.A. Warmerdam

Centre for Mathematics and Computer Science,
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

ABSTRACT: We study the semantics of a simple language with concurrency and recursion. Our semantic domain consists of (sets of) finite and infinite partially ordered multisets (pomsets) in order to model true concurrency (i.e. non-interleaved parallel execution). It will be shown that the set of pomsets can be turned into a complete ultra-metric space. With the induced notion of convergence, it is possible to provide meaning to infinite computations. Operational and denotational semantics for the considered language are provided and their equivalence is established by showing that both are fixed points of a contracting higher order operator. In a final section we give a tentative denotational semantics for an extension of the language with synchronization.

KEY WORDS AND PHRASES: denotational semantics, operational semantics, ω-proof rule, true concurrency, pomsets, metric topology.

1. Introduction

In earlier semantic investigations of the Amsterdam Concurrency Group (e.g. [BZ82, BKMOZ86, BM88, KR88, AR89, B89, BR89]), fruitful use has been made of the framework of complete metric spaces. Computations have a small distance (say 2^{-n}) if they differ only after n steps, and the induced metric turns many functions encountered in the semantic design into *contracting* mappings which have unique fixed points (by Banach's theorem). Elsewhere we have exploited these ideas to

- handle recursion and infinite processes in concurrency,
- establish equivalence of several semantics,
- define semantic operators modeling syntactic operators such as sequential and parallel composition,
- treat advanced language families such as parallel object-oriented and logic programming ([ABKR89, B88]).

In our investigations up to now, we have always adopted the so-called interleaving approach to concurrency (as suggested by the equation $\mathcal{M}(a \| b) = \{ ab, ba \}$). In the present paper, we show how the metric techniques may as well be applied to the noninterleaving (or partial order) approach to concurrency. As a case study, we provide a metric treatment of partially ordered multi sets (or *pomsets*, for short), as introduced and studied by Grabowski [Gr81], Pratt [Pr86], Gischer [Gi84], and Gaifman [Ga89] (for other references see [BRR89]). Our investigation of pomset semantics was inspired by a paper by Meyer and De Vink ([MV89]), where the semantic model is based on an order between pomsets which generalizes the usual stream order, and on the Smyth order between (certain) sets of pomsets.

The emphasis in our paper is on the development of the metric framework for pomsets, rather than on the study of some especially interesting programming language concepts. Therefore, we have chosen to illustrate our techniques firstly on a very simple parallel language, that does not even include a notion of synchronization. Later we include a CCS-style (but noninterleaving!) synchronization to this language. We show that a 'pure' noninterleaving treatment would fail in our setting and propose therefore a what might be called hybrid approach.

After introducing the metric and partial order preliminaries in Section 2, in Section 3 we present the metric framework proper to handle (sets of) pomsets. A distance is introduced which turns the collection of pomsets into a complete metric space. Next, we discuss the usual operators of sequential ('•') and parallel ('$\|$') composition. The pomset setting allows a particularly succinct definition of these. Extension of them to sets of pomsets requires some justification (in comparable situations, e.g. in [BM88], we usually handled this through the use of higher-order operators). A compactness lemma turned out to be useful here (cf. [BBKM84, theorems 2.9, 2.10 for related issues).

Section 4 contains the definition of the operational ($\mathcal{O}$) and denotational ($\mathcal{D}$) semantics. $\mathcal{O}$ is defined in terms of an (SOS-style) transition system with quite simple transitions : they are all of the form $s \xrightarrow{p}_d E$, with s a statement, p a pomset, d a declaration (mapping procedure variables to their bodies) and E the empty (or terminated) statement. On the other hand, the transition system includes some not-so-standard means to handle recursion. We mention here the introduction of a kind of ω-rule into

the system. (Further comments will follow in section 4.2.) The denotational semantics $\mathcal{D}$ is obtained as the (unique) fixed point of a higher order contracting mapping Φ. Since we also established that $\mathcal{O}$ satisfies a lemma which may, equivalently, be phrased as $\Phi(\mathcal{O}) = \mathcal{O}$, the desired equivalence $\mathcal{O} = \mathcal{D}$ is direct by Banach's theorem.

Section 5 contains a possible denotational semantics for the language extended with synchronization.

We conclude this introduction with a few words on future work :

- The operational semantics may be refined by also including transitions of the form $s \xrightarrow{p}_d s'$ (and by adapting the way the successive steps are assembled into the operational semantics $\mathcal{O}$).
- The pomset framework is (noninterleaving but) of the linear time variety : it assigns the same meanings to $a;(b_1+b_2)$ and $(a;b_1)+(a;b_2)$. In a paper in preparation, we show how four (systems of) domain equations may be defined which allow to define four pairs of equivalent semantics ($\mathcal{O}_i = \mathcal{D}_i$, $i = 1,...,4$), for each of the combinations interleaving / noninterleaving and linear time / branching time.
- We expect (or in some cases, know) that the pomset model may be replaced, without undue complications, by other models such as event structures or (sets of) directed acyclic graphs, preserving essentially the same metric approach.

Acknowledgements. We are indebted to John-Jules Meyer and Erik de Vink who showed us the way into the (erstwhile unknown to us) realm of true concurrency. We are also grateful to Erik de Vink for his detailed and constructive comments on preliminary versions of this paper. The members of the Amsterdam Concurrency Group provided useful comments on earlier presentations of the work.

2. Mathematical preliminaries

First of all we adopt the convention that a phrase like 'let $(x \in)X$ be ...' introduces a set X with variable x ranging over X.

For convenience, we introduce $\mathbb{N} = \{ 1, 2, 3, ... \}$, $\mathbb{N}_0 = \mathbb{N} \cup \{ 0 \}$, $\mathbb{N}^\infty = \mathbb{N} \cup \{ \infty \}$ and $\mathbb{N}_0^\infty = \mathbb{N} \cup \{ 0, \infty \}$.

2.1. Metric spaces

DEFINITION 2.1.1 A *metric space* is a pair (M, d) with M a non-empty set and d a mapping $d : M \times M \to [0, \infty)$, that satisfies the following properties.

(a) $\forall x,y \in M : d(x, y) = 0 \Leftrightarrow x = y$,

(b) $\forall x,y \in M : d(x, y) = d(y, x)$,

(c) $\forall x,y,z \in M : d(x, y) \leqslant d(x, y) + d(y, z)$.

A metric space is called 1-bounded if $d : M \times M \to [0, 1]$ (so the distance never

exceeds 1). In the sequel we assume that all metric spaces are 1-bounded.
A metric space is called *ultra-metric* or non-Archimedean if d satisfies $\forall x,y,z \in M : d(x, y) \leqslant \max\{ d(x, y), d(y, z) \}$.

DEFINITION 2.1.2 Let (M, d) be a metric space and let $(x_i)_i$ be a sequence in M.

1. $(x_i)_i$ is called a *Cauchy sequence* if $\forall \epsilon > 0 : \exists N \in \mathbb{N} : \forall n,m > N : d(x_n, x_m) < \epsilon$.
2. $(x_i)_i$ is called a *converging sequence* if
   $\exists x \in M : \forall \epsilon > 0 : \exists N \in \mathbb{N} : \forall n > N : d(x_n, x) < \epsilon$.
   We say $(x_i)_i$ converges to x or the limit of $(x_i)_i$ is x (which is unique) and write $\lim_i x_i = x$.
3. We call (M, d) *complete* if every Cauchy sequence is a converging sequence.

DEFINITION 2.1.3 Let (M_1, d_1) and (M_2, d_2) be metric spaces. Let $f : M_1 \to M_2$.

1. We call f *continuous* whenever
   $\forall x \in M_1 : \forall \epsilon > 0 : \exists \delta > 0 : \forall y \in M_1 : d_1(x, y) < \delta \Rightarrow d_2(f(x), f(y)) < \epsilon$ or, equivalently, for all converging sequences $(x_i)_i$ with $\lim_i x_i = x$ we have that $\lim_i f(x_i) = f(x)$.
2. Let $\gamma \geqslant 0$. With $M_1 \to^\gamma M_2$ we denote the set of all functions $f : M_1 \to M_2$ such that $\forall x,y \in M_1 : d_2(f(x), f(y)) \leqslant \gamma \cdot d_1(x, y)$. Functions $f \in M_1 \to^1 M_2$ are called *non-distance-increasing* (N.D.I.), functions $f \in M_1 \to^\epsilon M_2$ with $\epsilon < 1$ are called *contractions*.

PROPOSITION 2.1.4

1. Let (M_1, d_1) and (M_2, d_2) be metric spaces. For every $\gamma \geqslant 0$ and $f \in M_1 \to^\gamma M_2$ we have that f is continuous.
2. (Banach's fixed-point theorem)
   Let (M, d) be a complete metric space and $f : M \to M$ a contraction. Then there exists an $x \in M$ such that the following holds.
   - $f(x) = x$ (x is a fixed point of f),
   - $\forall y \in M : f(y) = y \Rightarrow y = x$ (x is unique),
   - $\forall y \in M : \lim_n f^n(y) = x$, where $f^1 = f$ and $f^{n+1} = f \circ f^n$.

DEFINITION 2.1.5 Let (M, d) be a metric space and let X be a subset of M.

1. X is called *closed*, whenever the limit of every converging sequence in X is an element of X.
2. X is called *compact*, whenever every sequence in X contains a subsequence that converges to an element in X.

3. The *closure* of X is the smallest closed subset in M containing X or, equivalently, the closure of X is the set of all limits of converging sequences in X. We denote the closure of X by $\overline{X}$.

DEFINITION 2.1.6
Let (M, d), (M_1, d_1), ..., (M_n, d_n) be metric spaces and let X be a set.

1. With $X \to M$ we denote the set of all functions from X to M.
   We define a metric d_F on $X \to M$ by $d_F(f_1, f_2) = sup \{ d(f_1(x), f_2(x)) \mid x \in X \}$.
2. We define a metric d_P on $M_1 \times ... \times M_n$ by
   $d_P((x_1, ..., x_n), (y_1, ..., y_n)) = \max \{d_i(x_i, y_i) \mid i = 1, ..., n \}$.
3. Let $\mathcal{P}_{nc}(M)$ denote $\{ X \subseteq M \mid X$ is non-empty and closed $\}$.
   We define a metric d_H on $\mathcal{P}_{nc}(M)$, called the Hausdorff distance, by $d_H(X, Y) = \max \{ sup \{ d(x, Y) \mid x \in X \}, sup \{ d(y, X) \mid y \in Y \} \}$, where $d(x, Z) = inf \{ d(x, z) \mid z \in Z \}$, for $x \in M$ and $Z \subseteq M$.

PROPOSITION 2.1.7
Let (M, d), (M_1, d_1), ..., (M_n, d_n) be metric spaces and let X be a set.

1. If (M, d), (M_1, d_1), ..., (M_n, d_n) are complete metric spaces then $(X \to M, d_F)$, $(M_1 \times ... \times M_n, d_P)$ and $(\mathcal{P}_{nc}(M), d_H)$ are complete metric spaces.
2. If (M, d), (M_1, d_1), ..., (M_n, d_n) are ultra-metric spaces then $(X \to M, d_F)$, $(M_1 \times ... \times M_n, d_P)$ and $(\mathcal{P}_{nc}(M), d_H)$ are ultra-metric spaces.

Only the proof of the completeness of $(\mathcal{P}_{nc}(M), d_H)$ is not so elementary. A proof can be found for instance in [BZ82].

If in the sequel we write $X \to M$, $M_1 \times ... \times M_n$ or $\mathcal{P}_{nc}(M)$ we mean the metric spaces with the metric defined above.

2.2. Partially ordered sets

DEFINITION 2.2.1 A *partially ordered set*, or just *partial order*, is a pair $(X, \leqslant)$ where X is a set and $\leqslant$ is a subset of $X \times X$ (notation : $x \leqslant y$ instead of $(x, y) \in \leqslant$), that satisfies the following conditions.

1. $\forall x \in X : x \leqslant x$,
2. $\forall x,y \in X : x \leqslant y$ and $y \leqslant x \Rightarrow x = y$,
3. $\forall x,y,z \in X : x \leqslant y$ and $y \leqslant z \Rightarrow x \leqslant z$.

We will adopt the notations $x < y$, $x \geqslant y$, $x > y$ for respectively $x \leqslant y \ \wedge \ x \neq y$, $y \leqslant x$, $y \leqslant x \ \wedge \ x \neq y$.

DEFINITION 2.2.2

1. For a partial order $(X, \leqslant)$ and $x \in X$ we define $lev(x) \in \mathbb{N}^\infty$ by $lev(x) = sup \{ n \mid \exists x_1 ... x_n \in X : x_1 < x_2 < ... < x_n = x \}$.
2. For a partial order $(X, \leqslant)$, we define $length((X, \leqslant)) \in \mathbb{N}_0^\infty$ by $length((X, \leqslant)) = sup \{ lev(x) \mid x \in X \}$, which is equal to $sup \{ n \mid \exists x_1 ... x_n \in X : x_1 < x_2 < ... < x_n \}$ (with the convention that $sup \varnothing = 0$).

DEFINITION 2.2.3 Let $(X, \leqslant)$ be a partial order and $A \subseteq X$.
We call A *downward-closed* if $\forall x \in X : [\exists a \in A : x \leqslant a] \Rightarrow x \in A$.

3. Pomsets

In the first subsection, the notion of pomset is defined, and some technical properties about pomsets are derived. In the second subsection, the set of pomsets is turned into a complete metric space and additionally a compactness property of pomsets is given. The third subsection, contains definitions of some operators on pomsets.

3.1. Definition of pomsets

Let $\mathscr{A}$ be a fixed set (finite or infinite) of atomic actions and $\mathscr{X}$ be a fixed (infinite) set of nodes, also called events.

DEFINITION 3.1.1 A labeled partial order or causality structure σ is a three-tuple $(X, \leqslant, \lambda)$, where X is a subset of $\mathscr{X}$, $\leqslant$ is a partial order on X, satisfying $\forall n \in \mathbb{N} : \{ x \mid lev(x) \leqslant n \}$ is finite and $\forall x \in X : lev(x) < \infty$, and $\lambda : X \to \mathscr{A}$ is a labeling function. We call $act(\sigma) = \{ \lambda(x) \mid x \in X \}$ the action set of σ.

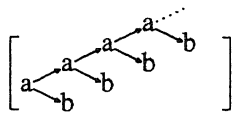
The intended meaning of a labeled partial order is the following. $\mathscr{X}$ is a set of names of events and $x_1 \leqslant x_2$ means event x_1 has to precede x_2. The meaning of λ is that $\lambda(x)$ is the action of event x or stated otherwise, x is an occurrence of $\lambda(x)$. The two restrictions on the partial order are essential for the proof of proposition 3.1.10 which, in turn, is needed to verify that the distance function, introduced in subsection 3.2 is indeed a metric. Furthermore they imply that every event has only a finite numbers of predecessors. Note that different events (even concurrent ones) may be labeled by the same action (our framework does not exclude so-called auto parallelism).
With a causality structure σ we associate X_σ, $\leqslant_\sigma$, λ_σ and also $x <_\sigma y$, $x \geqslant_\sigma y$, $x >_\sigma y$.
A pomset will be a causality structure modulo renaming of nodes, as introduced in
DEFINITION 3.1.2

1. Two structures σ and ρ are called isomorphic, if there exists a bijection

$\phi : X_\sigma \to X_\rho$ such that $\phi(x) \leqslant_\rho \phi(y) \Leftrightarrow x \leqslant_\sigma y$ and $\lambda_\rho \circ \phi = \lambda_\sigma$.

2. A pomset is an isomorphism class of causality structures. Let $(p, q \in) \mathcal{POM}$ denotes the collection of pomsets. $[\sigma]$ denotes a pomset with representative σ. $act([\sigma])$ is defined by $act(\sigma)$ (which is independent of the representative). The empty pomset $[(\varnothing, \varnothing, \varnothing)]$ is denoted by [].

We will draw pomsets by using Hasse diagrams of the partial order belonging to some representative causality structure, with the labels at the place of the nodes, as in



By the length of a structure, we mean the length of the order belonging to that structure. Note that $length(\sigma) < \infty \Leftrightarrow \#X_\sigma < \infty$. We also extend the notion of length to pomsets by taking the length of some representative. (This is independent of the choice of the representative.)

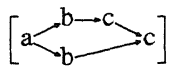
In section 4.2 we need another set of atomic actions (viz. $\mathcal{A}_e$). In that case we will denote $\mathcal{POM}$ w.r.t. $\mathcal{A}$ (resp $\mathcal{A}_e$) by $\mathcal{POM}[\mathcal{A}]$ (resp $\mathcal{POM}[\mathcal{A}_e]$).

We need the notion of truncation for defining a metric on $\mathcal{POM}$ in subsection 3.2.

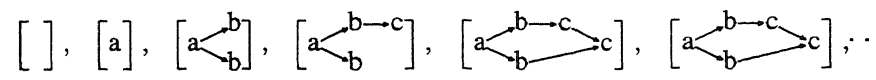
DEFINITION 3.1.3

1. For a causality structure σ and a downward-closed subset X of X_σ we define $\sigma \restriction X = (X, \leqslant \cap (X \times X), \lambda \restriction X)$. $\sigma \restriction X$ is a causality structure and $lev(x)$ w.r.t. $\sigma \restriction X$ is equal to $lev(x)$ w.r.t. σ.
2. For a causality structure σ and $n \in \mathbb{N}_0$ we define $\sigma[n] = \sigma \restriction \{ x \in X_\sigma \mid lev(x) \leqslant n \}$.
3. $p[n] = \{ \sigma[n] \mid \sigma \in p \} \in \mathcal{POM}$.

EXAMPLE 3.1.4 Let p be the following pomset.



Then $length(p) = 4$, $act(p) = \{ a, b, c \}$ and p has for instance one c at level 3 and one c at level 4. (To be more precise : every representative of p has two nodes labeled with c, one at level 3 and one at level 4.) The truncations $p[0]$, $p[1]$, $p[2]$, $p[3]$, $p[4]$, $p[5]$, ... are respectively



LEMMA 3.1.5 Let σ be a structure.

1. If $X \subseteq Y$ downward-closed and $Y \subseteq X_\sigma$ downward-closed then $(\sigma\restriction Y)\restriction X = \sigma\restriction X$.
2. If $X \subseteq X_\sigma$ downward-closed then $(\sigma\restriction X)[n] = (\sigma[n])\restriction (X_{\sigma[n]} \cap X)$.
3. $\sigma[n][m] = \sigma[\min\{ n, m \}]$.
4. For a pomset p we have that $p[n][m] = p[\min\{ n, m \}]$.

PROOF

1. Easy verification.
2. $(\sigma\restriction X)[n] = (\sigma\restriction X)\restriction \{ x \in X \mid lev(x)\_w.r.t.\_\sigma\restriction X \leqslant n \} =$
   $(\sigma\restriction X)\restriction \{ x \in X \mid lev(x)\_w.r.t.\_\sigma \leqslant n \} = \sigma\restriction \{ x \in X \mid lev(x)\_w.r.t.\_\sigma \leqslant n \} =$
   $(\sigma\restriction \{ x \in X_\sigma \mid lev(x)\_w.r.t.\_\sigma \leqslant n \})\restriction (X \cap \{ x \in X_\sigma \mid lev(x)\_w.r.t.\_\sigma \leqslant n \})$
   $= (\sigma[n])\restriction (X_{\sigma[n]} \cap X)$.
3. $(\sigma[n])[m] =$
   $(\sigma\restriction \{ x \in X_\sigma \mid lev(x)\_w.r.t.\_\sigma \leqslant n \})\restriction \{ x \in X_{\sigma[n]} \mid lev(x)\_w.r.t.\_\sigma[n] \leqslant m \} =$
   $\sigma\restriction \{ x \in X_{\sigma[n]} \mid lev(x)\_w.r.t.\_\sigma \leqslant m \} =$
   $\sigma\restriction \{ x \in X_\sigma \mid lev(x)\_w.r.t.\_\sigma \leqslant n \wedge lev(x)\_w.r.t.\_\sigma \leqslant m \} =$
   $\sigma\restriction \{ x \in X_\sigma \mid lev(x)\_w.r.t.\_\sigma \leqslant \min\{ n, m \} \} = \sigma\restriction [\min\{ n, m \}]$
4. Direct from 3. □

LEMMA 3.1.6

Let σ and ρ be structures.

Let $(Y_n)_n$ be a sequence of downward-closed subsets of X_ρ such that $\forall n : Y_n \subseteq Y_{n+1}$.

Let $\phi : X_\sigma \to X_\rho$ be a mapping such that

$\forall n : \phi\restriction X_{\sigma[n]} : \sigma[n] \to \rho\restriction Y_n$ is an isomorphism.

Then $\phi : \sigma \to \rho\restriction (\bigcup_n Y_n)$ is an isomorphism.

PROOF Easy verification. Uses the fact that $\bigcup_n X_{\sigma[n]} = X_\sigma$. □

LEMMA 3.1.7

Let σ and ρ be structures and let $\phi : X_\sigma \to X_\rho$.

Then $\phi : \sigma \to \rho$ is an isomorphism $\Leftrightarrow \forall n : \phi\restriction X_{\sigma[n]} : \sigma[n] \to \rho[n]$ is an isomorphism.

PROOF

"$\Rightarrow$" The only thing to check is $\phi[X_{\sigma[n]}] = X_{\rho[n]}$. This holds since $lev(\phi(x))\_w.r.t.\_\rho = lev(x)\_w.r.t.\_\sigma$.

"$\Leftarrow$" previous lemma : take $Y_n = X_{\rho[n]}$, then $\bigcup_n Y_n = X_\rho$. □

Next we are going to define a partial order $\leqslant$ on $\mathscr{POM}$. We use this partial order to prove Corollary 3.1.11 which, in turn, is needed to verify that the distance function, defined in subsection 3.2 is indeed a metric. Moreover, the partial order makes it possible to express that $\{ p \mid p \leqslant q \}$ is compact (proposition 3.2.5), which is used to prove that the operators, introduced is subsection 3.3, are well-defined.

DEFINITION 3.1.8 We define a relation $\leqslant$ on $\mathscr{POM}$ by putting $p \leqslant q$ iff $\exists \sigma$, $X : X \subseteq X_\sigma$ downward-closed, $q = [\sigma]$, $p = [\sigma \restriction X]$. In this case we say that p is *initial* to q.

PROPOSITION 3.1.9 "$\leqslant$" is a partial order.

PROOF

(1) If $p = [\sigma]$ then $[\sigma \restriction X_\sigma] = [\sigma] = p$ so $p \leqslant p$.

(2) Assume $p \leqslant q$ and $q \leqslant p$. Let ρ, X_2, σ, X_1 be such that $q = [\rho]$, $p = [\rho \restriction X_2]$, $p = [\sigma]$ and $q = [\sigma \restriction X_1]$. So $\sigma \sim \rho \restriction X_2$ and $\rho \sim \sigma \restriction X_1$, say $\phi : \sigma \to \rho \restriction X_2$ is an isomorphism and $\psi : \rho \to \sigma \restriction X_1$ is an isomorphism. Then also $\phi \restriction X_{\sigma[n]} : \sigma[n] \to (\rho \restriction X_2)[n] = \rho[n] \restriction (X_{\rho[n]} \cap X_2)$ and $\psi \restriction X_{\rho[n]} : \rho[n] \to (\sigma \restriction X_1)[n] = \sigma[n] \restriction (X_{\sigma[n]} \cap X_1)$ are isomorphisms. Since $X_{\sigma[n]}$ and $X_{\rho[n]}$ are finite sets, we can conclude that $X_{\rho[n]} \cap X_2 = X_{\rho[n]}$ and thus $\forall n : \phi \restriction X_{\sigma[n]} : \sigma[n] \to \rho[n]$ is an isomorphism so $\phi : \sigma \to \rho$ is an isomorphism so $p = [\sigma] = [\rho] = q$.

(3) Assume $p \leqslant q$ and $q \leqslant r$. Say $q = [\rho]$ and $p = [\rho \restriction X]$ with $X \subseteq X_\rho$ downward-closed and $r = [\zeta]$ and $q = [\zeta \restriction Y]$ with $Y \subseteq X_\zeta$ downward-closed. Say $\phi : \rho \to \zeta \restriction Y$ is an isomorphism then $\phi \restriction X : \rho \restriction X \to (\zeta \restriction Y) \restriction \phi[X]$ is an isomorphism. So $p = [\rho \restriction X] = [(\zeta \restriction Y) \restriction \phi[X]] = [\zeta \restriction \phi[X]]$, so $p \leqslant r$. □

PROPOSITION 3.1.10 $[\forall n : p[n] \leqslant q] \Rightarrow p \leqslant q$

PROOF

Let $p = [\sigma]$ and $q = [\rho]$.

We will show that there exist a downward-closed subset X of X_ρ and an isomorphism $\phi : \sigma \to \rho \restriction X$, which proves that $p = [\sigma] = [\rho \restriction X]$ or equivalently $p \leqslant q$.

We will make a tree of isomorphisms in the following way.

As nodes we take triples (ϕ, X, n) where (1) $n \in \mathbb{N}_0$, (2) $X \subseteq X_\rho$ is downward-closed, and (3) $\phi : \sigma[n] \to \rho \restriction X$ is an isomorphism.

We put an arc between (ϕ, X, n) and (ϕ', X', n') if (1) $n' = n + 1$, (2) $X' \supseteq X$, and (3) $\phi' \restriction X = \phi$.

First we show that this indeed defines a rooted tree with $(\emptyset, \emptyset, 0)$ as root, as follows. If $(\phi, X, n+1)$ is a node then $\phi : \sigma[n+1] \to \rho\restriction X$ is an isomorphism, so $\phi\restriction X_{\sigma}[n] : \sigma[n] \to \rho\restriction\phi[X_{\sigma[n]}]$ is an isomorphism thus $(\phi\restriction X_{\sigma[n]}, \phi[X_{\sigma[n]}], n)$ is a node and there is an arc from this node to $(\phi, X, n+1)$.

Next we show that this tree contains infinitely many nodes. In fact we show that $\forall n : \exists \phi, X : (\phi, X, n)$ is a node, as follows. Let us fix some n. Since $p[n] \leqslant q$, by definition there exist ρ' and X' such that $p[n] = [\rho'\restriction X']$ and $[\rho'] = q$. Let $\phi' : \rho' \to \rho$ be an isomorphism. Also $\phi'\restriction X' : \rho'\restriction X' \to \rho\restriction\phi'[X']$ is an isomorphism, so take $X = \phi'[X']$. Then $X \subseteq X_I$ is downward-closed and $\sigma[n] \sim \rho'\restriction X' \sim \rho\restriction X$. So there exists an isomorphism $\phi : \sigma[n] \to \rho\restriction X$ so (ϕ, X, n) is a node.

Since the number of events in $\sigma[n]$ is finite, say m, and there exist only a finite number of downward-closed subsets of X_ρ with m number of elements, we know that the tree is finitely branching. König's Lemma guarantees the existence of an infinite path : $(\phi_n, X_n, n)_{n=0}^{\infty}$ with $X_n \subseteq X_{n+1}$ and $\phi_{n+1}\restriction X_n = \phi_n$. Now take $\phi = \bigcup_{n=0}^{\infty} \phi_n$ and $X = \bigcup_{n=0}^{\infty} X_n$. Then, by lemma 3.1.6 $\phi : \sigma \to \rho\restriction X$ is an isomorphism. □

COROLLARY 3.1.11 $[\forall n : p[n] = q[n]] \Rightarrow p = q$

PROOF

$\forall n : p[n] = q[n] \leqslant q$ so $p \leqslant q$.

Analogous $q \leqslant p$.

So $p = q$. □

3.2. Metric for pomsets

We define a metric on $\mathcal{POM}$ as follows.

DEFINITION 3.2.1 $d : \mathcal{POM} \times \mathcal{POM} \to [0, 1]$ is defined by

$$d(p_1, p_2) = inf\{\, 2^{-n} \mid p_1[n] = p_2[n] \,\}$$

PROPOSITION 3.2.2 $(\mathcal{POM}, d)$ is a complete ultra-metric space.

PROOF Proposition 3.1.5.4 and corollary 3.1.11 imply that $(\mathcal{POM}, d)$ is an ultra-metric space. What remains is the verification of the completeness. Let $(p_n)_{n=1}^{\infty}$ be a Cauchy sequence. Take a nondescending chain $(n_m)_{m=1}^{\infty}$ such that $\forall m \in \mathbb{N} : \forall k > n_m : p_k[m] = p_{n_m}[m]$.

Define σ_m, $m \in \mathbb{N}$, recursively such that $\sigma_m \in p_{n_m}[m]$ and $\sigma_{m+1}[m] = \sigma_m$.

(1) Take $\sigma_1 \in p_{n_1}[1]$.

(2) If σ_m has been defined then $\sigma_m \in p_{n_m}[m] = p_{n_{m+1}}[m] = p_{n_{m+1}}[m+1][m]$ so there exists a $\sigma_{m+1} \in p_{n_{m+1}}[m+1]$ with $\sigma_{m+1}[m] = \sigma_m$.

Now define $\sigma = (\bigcup_{i=1}^{\infty} X_{\sigma_i}, \bigcup_{i=1}^{\infty} \leqslant_{\sigma_i}, \bigcup_{i=1}^{\infty} \lambda_{\sigma_i})$ and $p = [\sigma]$.

Then $p_n \to p \quad (n \to \infty)$ because $\sigma_m = \sigma[m]$ so $p_{n_m}[m] = p[m]$ so $\forall k > n_m : p_k[m] = p[m]$ so $\forall m \in \mathbb{N} : \forall k > n_m : d(p_k, p) \leqslant 2^{-m}$. □

See example 4.2.5 for a converging sequence in $\mathcal{POM}$.

PROPOSITION 3.2.3 For $p \in \mathcal{POM}$: $\lim_n p[n] = p$.

Finally, the semantic domain will be a collection of subsets of $\mathcal{POM}$. The need for *sets of* pomsets in our semantic domain, arises from the presence, in the language to be considered, of the concept of nondeterministic choice.

DEFINITION 3.2.4

Let $(P, Q \in) \mathcal{POM}^*$ is the set of all closed and non-empty subsets of $\mathcal{POM}$ (i.e. $\mathcal{P}_{nc}(\mathcal{POM})$).

$\mathcal{POM}^*$ is a complete (ultra-)metric space if it is endowed with the Hausdorff distance (see proposition 2.1.7).

Next we are going to define a useful compactness property.

PROPOSITION 3.2.5 $\forall q : \{ p \mid p \leqslant q \}$ is compact.

PROOF Let $(p_i)_i$ be a sequence with $p_i \leqslant q$. We are going to define $(n_i)_i$ (an increasing sequence of natural numbers) inductively such that if $n_0, \ldots, n_k$ are defined, it holds that

(1) $\forall i,j : i < j \leqslant k : p_{n_j}[i] = p_{n_i}[i]$,

(2) $\#\{ i \mid p_i[k] = p_{n_k}[k] \} = \infty$.

$k = 0$: choose n_0 arbitrary. (1) and (2) are trivially satisfied.

$k \to k+1$: denote $I = \{ i \mid p_i[k] = p_{n_k}[k] \}$. Since $\forall i : p_i[k+1] \leqslant q[k+1]$ and $q[k+1]$ is finite, there exist only finitely many distinct $p_i[k+1]$, so there exists an $n_{k+1} \in I$ such that $n_{k+1} > n_k$ and $\#\{ i \mid p_i[k+1] = p_{n_{k+1}}[k+1] \} = \infty$. Moreover, $\forall i \leqslant k : p_{n_{k+1}}[i] = p_{n_{k+1}}[k][i] = p_{n_k}[k][i] = p_{n_k}[i] = p_{n_i}[i]$. So (1) and (2) are satisfied.

From (1) we can conclude that $(p_{n_j})_j$ is a Cauchy subsequence, and by proposition

3.1.10 we know that the limit is in $\{ p \mid p \leqslant q \}$, so $\{ p \mid p \leqslant q \}$ is compact. □

3.3. Operators on pomsets

In this subsection we are going to define two operators on pomsets, namely sequential and parallel composition. This is done in the following way. First we define the operators on structures (with disjoint sets of nodes only). Since the isomorphism relation will be a congruence relation with respect to these operators, the operators can be defined on pomsets. Finally, we will define the two operators on pomset-sets. As we go along, we derive some properties of these operators.

DEFINITION 3.3.1 Let σ and ρ be causality structures such that $X_\sigma \cap X_\rho = \varnothing$.

1. $\sigma \bullet \rho = \begin{cases} \sigma, \text{ if } \#X_\sigma = \infty \text{ (or equivalently } \mathit{length}(\sigma) = \infty \text{)}, \\ (X_\sigma \cup X_\rho,\ \leqslant_\sigma \cup \leqslant_\rho \cup (X_\sigma \times X_\rho),\ \lambda_\sigma \cup \lambda_\rho), \text{ otherwise.} \end{cases}$
2. $\sigma \parallel \rho = (X_\sigma \cup X_\rho,\ \leqslant_\sigma \cup \leqslant_\rho,\ \lambda_\sigma \cup \lambda_\rho)$.

LEMMA 3.3.2

1. if σ is finite then $\mathit{lev}(x)$ w.r.t. $\sigma\bullet\rho = \begin{cases} \mathit{lev}(x) \text{ w.r.t. } \sigma, \text{ if } x \in X_\sigma, \\ \mathit{lev}(x) \text{ w.r.t. } \rho + \mathit{length}(\sigma), \text{ if } x \in X_\rho. \end{cases}$
2. $\mathit{lev}(x)$ w.r.t. $\sigma \parallel \rho = \begin{cases} \mathit{lev}(x) \text{ w.r.t. } \sigma, \text{ if } x \in X_\sigma, \\ \mathit{lev}(x) \text{ w.r.t. } \rho, \text{ if } x \in X_\rho. \end{cases}$
3. $\sigma\bullet\rho$ and $\sigma \parallel \rho$ are causality structures.
4. if $\mathit{length}(\sigma) \geqslant n$ then $(\sigma\bullet\rho)[n] = \sigma[n]$;
   if $\mathit{length}(\sigma) \leqslant n$ then $(\sigma\bullet\rho)[n] = \sigma \bullet \rho[n - \mathit{length}(\sigma)]$.
5. $(\sigma \parallel \rho)[n] = \sigma[n] \parallel \rho[n]$.
6. if σ is finite then $\mathit{act}(\sigma\bullet\rho) = \mathit{act}(\sigma) \cup \mathit{act}(\rho)$;
   if σ is infinite then $\mathit{act}(\sigma\bullet\rho) = \mathit{act}(\sigma)$.
7. $\mathit{act}(\sigma \parallel \rho) = \mathit{act}(\sigma) \cup \mathit{act}(\rho)$.

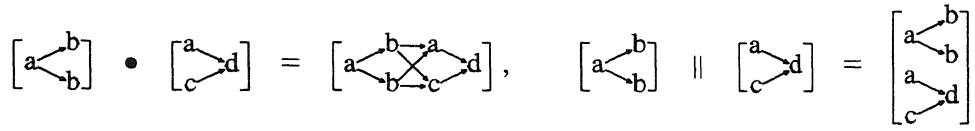
Now let us define the operators $\bullet$ and $\parallel$ on pomsets.

DEFINITION 3.3.3

$\bullet : \mathcal{POM} \times \mathcal{POM} \to \mathcal{POM}$ and $\parallel : \mathcal{POM} \times \mathcal{POM} \to \mathcal{POM}$ are defined as follows.
If $p = [\sigma]$ and $q = [\rho]$, with $X_\sigma \cap X_\rho = \varnothing$, then $p\bullet q = [\sigma\bullet\rho]$ and $p \parallel q = [\sigma \parallel \rho]$.

REMARK 3.3.4 It is always possible to find representatives with disjoint set of nodes and furthermore the definition is not dependent on the choice of the representatives.

EXAMPLES 3.3.5



LEMMA 3.3.6

1. if $length(p) \geqslant n$ then $(p \bullet q)[n] = p[n]$;
   if $length(p) \leqslant n$ then $(p \bullet q)[n] = p \bullet q[n - length(p)]$.
2. $(p \,||\, q)[n] = p[n] \,||\, q[n]$.
3. if p is finite then $act(p \bullet q) = act(p) \cup act(q)$;
   if p is infinite then $act(p \bullet q) = act(p)$.
4. $act(p \,||\, q) = act(p) \cup act(q)$.

Now we will define $\bullet$ and $||$ on $\mathcal{POM}^*$. Also an operator $+$ is defined on $\mathcal{POM}^*$, which is just the set-theoretic union.

DEFINITION 3.3.7

1. $\bullet : \mathcal{POM}^* \times \mathcal{POM}^* \to \mathcal{POM}^*$ is defined by
   $P \bullet Q = \{ p \bullet q \mid p \in P \text{ and } q \in Q \}$.
2. $|| : \mathcal{POM}^* \times \mathcal{POM}^* \to \mathcal{POM}^*$ is defined by
   $P \,||\, Q = \{ p \,||\, q \mid p \in P \text{ and } q \in Q \}$.
3. $+ : \mathcal{POM}^* \times \mathcal{POM}^* \to \mathcal{POM}^*$ is defined by
   $P + Q = P \cup Q$.

We need to show that $P \bullet Q$ and $P \,||\, Q$ are closed. (The fact that $P + Q$ is closed is immediate.) For this purpose, a lemma is given first.

LEMMA 3.3.8

1. $\forall p,q,q' \in \mathcal{POM} : d(p \bullet q, p \bullet q') = 2^{-length(p)} \cdot d(q, q')$.
2. If $\lim_n (p \bullet q_n) = r$ with $length(p) < \infty$, then $\exists q : \lim_n q_n = q$ and $r = p \bullet q$.

PROOF

1. If $length(p) = \infty$ then both sides give 0. Now suppose $length(p) = l < \infty$. $(p \bullet q)[n+l] = (p \bullet q')[n+l] \Leftrightarrow p \bullet q[n] = p \bullet q'[n] \Leftrightarrow q[n] = q'[n]$. From this 1. follows immediately.
2. Let $length(p) = l$. Since $d(p \bullet q_n, p \bullet q_m) = 2^{-l} \cdot d(q_n, q_m)$ and $(p \bullet q_n)_n$ is a Cauchy sequence, we have $(q_n)_n$ is a Cauchy sequence, say $q_n \to q$. Then $r = \lim_n (p \bullet q_n) = p \bullet \lim_n q_n = p \bullet q$. □

PROPOSITION 3.3.9

1. $P \bullet Q$ is closed.
2. $P \parallel Q$ is closed.

PROOF

1. Let $r = \lim_i r_i$ with $r_i \in P \bullet Q$, say $r_i = p_i \bullet q_i$, with $p_i \in P$ and $q_i \in Q$. Since $r_i \to r$, we have that $\forall l : \exists k_l : r_{k_l}[l] = r[l]$. So $r[l] = (p_{k_l} \bullet q_{k_l})[l]$, so $p_{k_l}[l] \leqslant r[l] \leqslant r$. By the compactness property(3.2.5), there exists an increasing sequence l_m such that $(p_{k_{l_m}}[l_m])_m$ converges, say to $p \in P$. (Note $p \in P$, since also $p_{k_{l_m}} \to p$ and P is closed.)

   If $\mathit{length}(p) = \infty$ then $\forall n : r[n] = \lim_i ((p_i \bullet q_i)[n]) = \lim_m ((p_{k_{l_m}} \bullet q_{k_{l_m}})[n]) = \lim_m p_{k_{l_m}}[n] = p[n]$. So $r = p =$ (for instance) $p \bullet q_0 \in P \bullet Q$.

   If $\mathit{length}(p) < \infty$ then $\exists M : \forall m \geqslant M : p_{k_{l_m}}[l_m] = p$. Moreover, since l_m is increasing, $\exists M' : \forall m \geqslant M' : p_{k_{l_m}} = p$. According to lemma 3.3.8.2, we have that $r = p \bullet q$ with $\lim_m q_{k_{l_m}} = q \in Q$.
2. Let $r = \lim_i r_i$ with $r_i \in P \parallel Q$, say $r_i = p_i \parallel q_i$, with $p_i \in P$ and $q_i \in Q$. Since $r_i \to r$, we have that $\forall l : \exists k_l : r_{k_l}[l] = r[l]$. So $r[l] = r_{k_l}[l] = p_{k_l}[l] \parallel q_{k_l}[l]$. So $p_{k_l}[l] \leqslant r[l] \leqslant r$ and $q_{k_l}[l] \leqslant r[l] \leqslant r$. By the compactness property, there exists an increasing sequence l_m such that $(p_{k_{l_m}}[l_m])_m$ converges, say to $p \in P$. Again by the compactness property, there exists an increasing sequence m_n such that $(q_{k_{l_{m_n}}}[l_{m_n}])_n$ converges, say to $q \in Q$. Now $r = \lim_n r_{k_{l_{m_n}}} = \lim_n (p_{k_{l_{m_n}}} \parallel q_{k_{l_{m_n}}}) = (\lim_n p_{k_{l_{m_n}}}) \parallel (\lim_n q_{k_{l_{m_n}}}) = p \parallel q \in P \parallel Q$. □

4. Semantics

In this section a simple language without synchronization $\mathscr{L}$ is introduced and an operational semantics $\mathscr{O}$ and a denotational semantics $\mathscr{D}$ are given and are proved to be equal.

4.1. The language

First we introduce the language. For this we need two basic sets. Let $(a, b, c, \ldots \in) \mathscr{A}$ be a (finite or infinite) set of atomic actions and let $(x \in) \mathscr{V}\mathit{ar}$ be a set of procedure variables.

DEFINITION 4.1.1

a. The class $(s \in) \mathscr{L}$ of *statements* is given by

$$s ::= a \mid x \mid s_1;s_2 \mid s_1+s_2 \mid s_1 \| s_2.$$

b The class ($g \in$) $\mathcal{L}^g$ of *guarded statements* is given by

$$g ::= a \mid g;s \mid g_1 + g_2 \mid g_1 \| g_2.$$

c. The class ($d \in$) $\mathcal{D}ecl$ of *declarations* consists of mappings from $\mathcal{P}var$ to $\mathcal{L}^g$.

d. The class ($t \in$) $\mathcal{P}rog$ of *programs* consists of pairs $t \equiv < d \mid s >$ with $d \in \mathcal{D}ecl$ and $s \in \mathcal{L}$.

A statement is made up from atomic actions and procedure variables, by means of sequential composition, nondeterministic choice and (non-interleaved) parallel composition. A guarded statement is a statement in which every procedure variable is preceded by an atomic action. A declaration is a mapping from procedure variables to guarded statements and finally a program is a declaration plus a statement.

4.2. Operational semantics

The operational semantics is given with the aid of a labeled transition system (l.t.s.). As labels we use pomsets (cf. [BoCa88, Ga89]). In an l.t.s. we encounter, besides statements $s \in \mathcal{L}$, also the special symbol E that we use to indicate the empty (or terminated) statement. In addition, we introduce a special atomic action $e (\notin \mathcal{A})$, used -in a way to be explained below- to handle recursion, and we put $\mathcal{A}_e = \mathcal{A} \cup \{ e \}$. Let $\longrightarrow \subseteq \mathcal{L} \times \mathcal{POM}[\mathcal{A}_e] \times \mathcal{D}ecl \times \{ E \}$ to be defined in a moment. Thus, we only employ transitions of a particular simple form, which we shall write as $s \xrightarrow{p}_d E$ (instead of $(s, p, d, E) \in \longrightarrow$). Some explanations follow after definitions 4.2.1 and 4.2.2.

DEFINITION 4.2.1 $\longrightarrow \subseteq \mathcal{L} \times \mathcal{POM}[\mathcal{A}_e] \times \mathcal{D}ecl \times \{ E \}$ is the smallest relation satisfying

(1) $a \xrightarrow{[a]}_d E$,

(2) if $g \xrightarrow{p}_d E$ and $d(x) = g$ then $x \xrightarrow{p}_d E$,

(3) if $s_1 \xrightarrow{p_1}_d E$ and $s_2 \xrightarrow{p_2}_d E$ then $s_1;s_2 \xrightarrow{p_1 \bullet p_2}_d E$ and $s_1 \| s_2 \xrightarrow{p_1 \| p_2}_d E$,

(4) if $s_1 \xrightarrow{p}_d E$ then $s_1 + s_2 \xrightarrow{p}_d E$ and $s_2 + s_1 \xrightarrow{p}_d E$,

(5) if $s \xrightarrow{p_i}_d E$ ($i = 1\ 2\ ...$) and $\lim_i p_i = p$ then $s \xrightarrow{p}_d E$,

(6) $x \xrightarrow{[e]}_d E$.

DEFINITION 4.2.2

1. $\mathcal{I}_d : \mathcal{L} \to \mathcal{POM}^*[\mathcal{A}_e]$ is given by $\mathcal{I}_d(s) = \{ p \mid s \xrightarrow{p}_d E \}$.
2. $\mathcal{O}_d : \mathcal{L} \to \mathcal{POM}^*[\mathcal{A}]$ is given by $\mathcal{O}_d(s) = \mathcal{I}_d(s) \cap \mathcal{POM}[\mathcal{A}]$.
3. $\mathcal{O} : \mathcal{P}rog \to \mathcal{POM}^*[\mathcal{A}]$ is given by $\mathcal{O}(< d \mid s >) = \mathcal{O}_d(s)$.

First we discuss the system for '$\longrightarrow$'. Clauses (1), ..., (4) of definition 4.2.1 should be clear. Clauses (5) and (6) are included in order to enable us to handle possibly infinite computations of recursive procedures. Since we only work with transitions of the form $s \xrightarrow{p}_d E$ (which terminate in one step), we have no means to build up an infinite computation without additional measures. These are provided by (5) and (6) : Axiom (6) provides an arbitrary (cf. Banach's theorem) starting point for the execution of a recursive process. Rule (5) allows us to build up possibly infinite p in a $s \xrightarrow{p}_d E$ step. This set-up would allow e to remain in the final outcome of a computation. Therefore, we obtain the desired operational semantics $\mathcal{O}_d(s)$ by restricting (def. 4.2.2, part 2.) the intermediate semantics $\mathcal{F}_d(s)$ to those outcomes which contain only pomsets involving actions from $\mathcal{A}$. Example 4.2.5 should be helpful to understand our handling of recursion.

LEMMA 4.2.3

1. $\mathcal{F}_d$ is well-defined, i.e. $\mathcal{F}_d(s)$ is non-empty and closed.
2. $\mathcal{O}_d(s)$ is non-empty and closed.

PROOF

1. By induction on the complexity of s, one can easily show that $\mathcal{F}_d(s) \neq \emptyset$ (use rule (6) in case $s = x$). Because of rule (5), $\mathcal{F}_d(s)$ is closed.
2. $\mathcal{O}_d(s)$ is closed since $\mathcal{F}_d(s)$ is closed. Proving $\mathcal{O}_d(s) \neq \emptyset$ is more involved. We construct a sequence $p_i \in \mathcal{F}_d(s)$ $(i \in \mathbb{N}_0)$ such that $e \notin act(p_i[i])$ and $p_{i+1}[i] = p_i[i]$. From this it follows that $(p_i)_i$ is a Cauchy sequence, say with limit p. $p \in \mathcal{F}_d(s)$ and $\forall n \in \mathbb{N}_0 : e \notin act(p[n])$, so $e \notin act(p)$. We can conclude that $p \in \mathcal{O}_d(s)$.

   The sequence is constructed in the following way. $\mathcal{F}_d(s) \neq \emptyset$, so take a $p_0 \in \mathcal{F}_d(s)$. If $p_k \in \mathcal{F}_d(s)$ with $e \notin act(p_k[k])$ then we can find a $p_{k+1} \in \mathcal{F}_d(s)$ with $p_{k+1}[k] = p_k[k]$ and $e \notin act(p_{k+1}[k+1])$, which is guaranteed by the following lemma. □

LEMMA 4.2.4 If $s \xrightarrow{p}_d E$ and $e \notin act(p[n])$ then $\exists p' : s \xrightarrow{p'}_d E$ and $p'[n] = p[n]$ and $e \notin act(p'[n+1])$.

PROOF First we remark that $\forall g \in \mathcal{G}^g : \exists p : g \xrightarrow{p}_d E$ and $e \notin act(p[1])$, which can easily be proved by induction on the structure of g and using lemma 3.3.6.

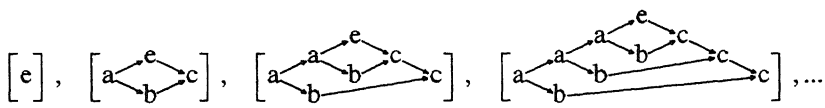
The lemma is proved by transfinite induction on the depth of the proof tree for $s \xrightarrow{p}_d E$, defined in the usual way.

- If $a \xrightarrow{[a]}_d E$ by (1) then we can take $p' = [a]$.
- If $x \xrightarrow{p}_d E$ by rule (2) then $g \xrightarrow{p}_d E$ with $g = d(x)$. By induction $\exists p' : g \xrightarrow{p'}_d E$

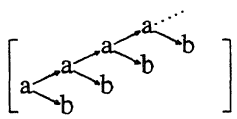
with $p'[n] = p[n]$ and $e \notin act(p'[n+1])$. Now also $x \xrightarrow{p'}_d E$.

- If $s_1;s_2 \xrightarrow{p_1 \bullet p_2}_d E$ (resp. $s_1 \| s_2 \xrightarrow{p_1 \| p_2}_d E$) by (3) then $\exists p'_1, p'_2 : s_1 \xrightarrow{p'_1}_d E$ and $s_2 \xrightarrow{p'_2}_d E$ with $e \notin act(p'_1[n+1])$, $e \notin act(p'_2[n+1])$, $p'_1[n] = p_1[n]$ and $p'_2[n] = p_2[n]$. Now $s_1;s_2 \xrightarrow{p'_1 \bullet p'_2}_d E$ (resp $s_1 \| s_2 \xrightarrow{p'_1 \| p'_2}_d E$) and $e \notin act((p'_1 \bullet p'_2)[n+1])$ and $(p'_1 \bullet p'_2)[n] = (p_1 \bullet p_2)[n]$ (resp $e \notin act((p'_1 \| p'_2)[n+1])$ and $(p'_1 \| p'_2)[n] = (p_1 \| p_2)[n]$).
- If $s_1 + s_2 \xrightarrow{p}_d E$ by (4) then $s_i \xrightarrow{p}_d E$ ($i = 1$ or 2). By induction $\exists p' : s_i \xrightarrow{p'}_d E$ with $p'[n] = p[n]$ and $e \notin act(p'[n+1])$. Now also $s_1 + s_2 \xrightarrow{p'}_d E$.
- If $s \xrightarrow{p}_d E$ by applying rule (5) then $s \xrightarrow{p_i}_d E$ ($i = 1,2...$) and $\lim_i p_i = p$. Now $\exists i_0 : p_{i_0}[n] = p[n]$. By induction $\exists p' : s \xrightarrow{p'}_d E$ and $e \notin act(p'[n+1])$ and $p'[n] = p_{i_0}[n] = p[n]$.
- If $x \xrightarrow{[e]}_d E$ by axiom (6) and $d(x)=g$ then $g \xrightarrow{p}_d E$ with $e \notin act(p[1])$ and so $x \xrightarrow{p}_d E$. □

EXAMPLE 4.2.5 Let $d(x) = a;(x \| b);c$ and $s \equiv x$. By applying rules (1), (2), (3), (6), one can derive $s \xrightarrow{p_i}_d E$, for p_1, p_2, p_3, p_4, ... equal to



Applying rule (5) gives $s \xrightarrow{p}_d E$ with $p = \lim_i p_i =$



So $\mathcal{F}_d(s)$ is the set of all pomsets listed above and $\mathcal{O}_d(s)$ is only the singleton set with the last pomset, as (only) member.

LEMMA 4.2.6

1. a. $\{ [a] \} = \mathcal{F}_d(a)$,
   b. $\mathcal{F}_d(g) \cup \{ [e] \} = \mathcal{F}_d(x)$, if $d(x) = g$,
   c. $\mathcal{F}_d(s_1) \bullet \mathcal{F}_d(s_2) = \mathcal{F}_d(s_1;s_2)$,
   d. $\mathcal{F}_d(s_1) \| \mathcal{F}_d(s_2) = \mathcal{F}_d(s_1 \| s_2)$,
   e. $\mathcal{F}_d(s_1) \cup \mathcal{F}_d(s_2) = \mathcal{F}_d(s_1 + s_2)$.
2. a. $\mathcal{O}_d(a) = \{ [a] \}$,
   b. $\mathcal{O}_d(x) = \mathcal{O}_d(g)$, when $d(x) = g$,
   c. $\mathcal{O}_d(s_1;s_2) = \mathcal{O}_d(s_1) \bullet \mathcal{O}_d(s_2)$,

d. $\mathcal{O}_d(s_1 \| s_2) = \mathcal{O}_d(s_1) \parallel \mathcal{O}_d(s_2)$,

e. $\mathcal{O}_d(s_1 + s_2) = \mathcal{O}_d(s_1) \cup \mathcal{O}_d(s_2)$.

PROOF

1. First we prove that a ... e hold with "$\subseteq$" instead of "$=$". Only case c. is proved because a. immediately follows from axiom (1) and b. from axiom (6) and rule (2), d. is like c. and e. follows from rule (4).

   Let $p \in \mathcal{F}_d(s_1) \bullet \mathcal{F}_d(s_2)$. Then $p = p_1 \bullet p_2$ with $p_1 \in \mathcal{F}_d(s_1)$ and $p_2 \in \mathcal{F}_d(s_2)$. So $s_1 \xrightarrow{p_1}_d E$ and $s_2 \xrightarrow{p_2}_d E$ so (rule(3)) $s_1;s_2 \xrightarrow{p_1 \bullet p_2}_d E$ or equivalently $p \in \mathcal{F}_d(s_1;s_2)$.

   To prove "$=$", define $\mathcal{F}_d'$ as follows.

   $\mathcal{F}_d'(a) = \{ [a] \}$, $\mathcal{F}_d'(x) = \mathcal{F}_d(g) \cup \{ [e] \}$, when $d(x) = g$,

   $\mathcal{F}_d'(s_1;s_2) = \mathcal{F}_d(s_1) \bullet \mathcal{F}_d(s_2)$, $\mathcal{F}_d'(s_1 \| s_2) = \mathcal{F}_d(s_1) \parallel \mathcal{F}_d(s_2)$ and

   $\mathcal{F}_d'(s_1 + s_2) = \mathcal{F}_d(s_1) \cup \mathcal{F}_d(s_2)$.

   It follows immediately that $\forall s : \mathcal{F}_d'(s) \subseteq \mathcal{F}_d(s)$ and $\mathcal{F}_d'(s)$ is closed.

   Define $\longrightarrow'$ by $s \xrightarrow{p}{}'_d E \Leftrightarrow p \in \mathcal{F}_d'(s)$

   $\longrightarrow'$ satisfies rules (1)...(6) :

   (1) trivial.

   (2) if $g \xrightarrow{p}{}'_d E$ and $d(x) = g$ then $p \in \mathcal{F}_d'(g) \subseteq \mathcal{F}_d(g) \subseteq \mathcal{F}_d'(x)$.

   (3) if $s \xrightarrow{p_1}{}'_d E$ and $s \xrightarrow{p_2}{}'_d E$ then $p_1 \in \mathcal{F}_d'(s_1) \subseteq \mathcal{F}_d(s_1)$ and $p_2 \in \mathcal{F}_d'(s_2) \subseteq \mathcal{F}_d(s_2)$ so $p_1 \bullet p_2 \in \mathcal{F}_d'(s_1;s_2)$ and $p_1 \| p_2 \in \mathcal{F}_d'(s_1 \| s_2)$.

   (4) if $s_1 \xrightarrow{p}{}'_d E$ then $p \in \mathcal{F}_d'(s_1) \subseteq \mathcal{F}_d(s_1) \subseteq \mathcal{F}_d'(s_1 + s_2)$ so $s_1 + s_2 \xrightarrow{p}{}'_d E$ and similar $s_2 + s_1 \xrightarrow{p}{}'_d E$.

   (5) $\mathcal{F}_d'(s)$ is closed.

   (6) $[e] \in \mathcal{F}_d'(x)$ so $x \xrightarrow{[e]}{}'_d E$.

   $\longrightarrow$ is the smallest relation satisfying (1)...(6), so $\longrightarrow \subseteq \longrightarrow'$ or equivalently $\mathcal{F}_d(s) \subseteq \mathcal{F}_d'(s)$.

   This proves 1.

2. a. $\mathcal{O}_d(a) = \mathcal{F}_d(a) \cap \mathcal{POM}[\mathcal{A}] = \{ [a] \} \cap \mathcal{POM}[\mathcal{A}] = \{ [a] \}$.

   b. $\mathcal{O}_d(x) = \mathcal{F}_d(x) \cap \mathcal{POM}[\mathcal{A}] = \mathcal{F}_d(g) \cap \mathcal{POM}[\mathcal{A}] = \mathcal{O}_d(g)$.

   c. $\mathcal{O}_d(s_1;s_2) = \mathcal{F}_d(s_1;s_2) \cap \mathcal{POM}[\mathcal{A}] = (\mathcal{F}_d(s_1) \bullet \mathcal{F}_d(s_2)) \cap \mathcal{POM}[\mathcal{A}] \overset{\alpha}{=}$

   $(\mathcal{F}_d(s_1) \cap \mathcal{POM}[\mathcal{A}]) \bullet (\mathcal{F}_d(s_2) \cap \mathcal{POM}[\mathcal{A}]) = \mathcal{O}_d(s_1) \bullet \mathcal{O}_d(s_2)$.

   Maybe the equality marked with an α needs some explanation.

   "$\supseteq$" is trivial : $\mathcal{POM}[\mathcal{A}]$ is closed under $\bullet$.

   To prove "$\subseteq$" : $(\mathcal{F}_d(s_1) \bullet \mathcal{F}_d(s_2)) \cap \mathcal{POM}[\mathcal{A}] =$

   $\{ p_1 \bullet p_2 \in \mathcal{POM}[\mathcal{A}] \mid p_1 \in \mathcal{F}_d(s_1) \text{ and } p_2 \in \mathcal{F}_d(s_2) \} = (*)$.

   Let $p_1 \bullet p_2 \in (*)$. We have $e \notin act(p_1)$.

   If $length(p_1) = \infty$ then

take a $p'_2 \in \mathcal{O}_d(s_2) = \mathcal{I}_d(s_2) \cap \mathcal{POM}[\mathcal{A}]$ $(\mathcal{O}_d(s_2) \neq \emptyset)$. So $p_1 \bullet p_2 = p_1 = p_1 \bullet p'_2 \in (\mathcal{I}_d(s_1) \cap \mathcal{POM}[\mathcal{A}]) \bullet (\mathcal{I}_d(s_2) \cap \mathcal{POM}[\mathcal{A}])$.

If $length(p_1) < \infty$ then also $e \notin act(p_2)$ so

$p_1 \bullet p_2 \in (\mathcal{I}_d(s_1) \cap \mathcal{POM}[\mathcal{A}]) \bullet (\mathcal{I}_d(s_2) \cap \mathcal{POM}[\mathcal{A}])$.

d. like c. but now the corresponding equation marked with the α is direct.

e. like d. □

4.3. Denotational semantics

In this section we are going to define a denotational semantics for $\mathcal{L}$. This is done with the aid of some higher-order operator, that will turn out to be a contraction. To prove this, we need the following lemma.

LEMMA 4.3.1

1. $\forall p,q,p',q' \in \mathcal{POM}$: if $p \neq []$ and $p' \neq []$ then $d(p \bullet q, p' \bullet q') \leqslant \max \{ d(p, p'), \frac{1}{2} d(q, q') \}$.
2. $\forall p,q,p',q' \in \mathcal{POM} : d(p \| q, p' \| q') \leqslant \max \{ d(p, p'), d(q, q') \}$.
3. $\forall P,Q,P',Q' \in \mathcal{POM}^*$: if $[] \notin P$ and $[] \notin P'$ then $d(P \bullet Q, P' \bullet Q') \leqslant \max \{ d(P, P'), \frac{1}{2} d(Q, Q') \}$.
4. $\forall P,Q,P',Q' \in \mathcal{POM}^* : d(P \| Q, P' \| Q') \leqslant \max \{ d(P, P'), d(Q, Q') \}$.
5. $\forall P,Q,P',Q' \in \mathcal{POM}^* : d(P + Q, P' + Q') \leqslant \max \{ d(P, P'), d(Q, Q') \}$.

PROOF

1. If $\max \{ d(p, p'), \frac{1}{2} d(q, q') \} = 1$ then 1. holds trivially.

   If $\max \{ d(p, p'), \frac{1}{2} d(q, q') \} \leqslant 2^{-n}$ $(n \geqslant 1)$ then $p[n] = p'[n]$ and $q[n-1] = q'[n-1]$. If $length(p) \geqslant n$ then also $length(p') \geqslant n$ and we have $(p \bullet q)[n] = p[n] = p'[n] = (p' \bullet q')[n]$. If $length(p) < n$ then $p = p'$ and so $(p \bullet q)[n] = p \bullet q[n - length(p)]$ (because $length(p) > 0$) $= p \bullet q'[n - length(p)]$ $= (p \bullet q')[n] = (p' \bullet q')[n]$. So $d(p \bullet q, p' \bullet q') \leqslant 2^{-n}$.
2. Similar to 1.
3. This is a consequence of 1. Details can be found in the appendix.
4. Similar to 3.
5. Straightforward verification. □

Now we will define the higher-order mapping.

DEFINITION 4.3.2 $\Phi_d : (\mathcal{L} \to \mathcal{POM}^*) \to (\mathcal{L} \to \mathcal{POM}^*)$ is defined as follows.

Let $F \in \mathcal{L} \to \mathcal{POM}^*$.

$\Phi_d(F)(a) = \{ [a] \}$

$\Phi_d(F)(s_1;s_2) = \Phi_d(F)(s_1) \bullet F(s_2)$
$\Phi_d(F)(s_1 \| s_2) = \Phi_d(F)(s_1) \| \Phi_d(F)(s_2)$
$\Phi_d(F)(s_1 + s_2) = \Phi_d(F)(s_1) + \Phi_d(F)(s_2)$
$\Phi_d(F)(x) = \Phi_d(F)(d(x))$

LEMMA 4.3.3

1. $\Phi_d(F)$ is well-defined.
2. $[] \notin \Phi_d(F)(s)$
3. Φ_d is a contraction.

PROOF 1. and 2. can easily be shown, first for guarded statements and then for general statements, with induction on the complexity of the statements. For 3. one needs to show $\forall s \in \mathscr{L} : d(\Phi_d(F_1)(s), \Phi_d(F_2)(s)) \leqslant \frac{1}{2} d(F_1, F_2)$. Again, this can be shown, first for guarded statements and then for general statements, with induction on the complexity of the statements. We only treat the case $s = s_1;s_2$ as an example.

$d(\Phi_d(F_1)(s_1;s_2), \Phi_d(F_2)(s_1;s_2)) =$
$d(\Phi_d(F_1)(s_1) \bullet F_1(s_2), \Phi_d(F_2)(s_1) \bullet F_2(s_2)) \leqslant$ by part 2. and lemma 4.3.1.3
$\max\{ d(\Phi_d(F_1)(s_1), \Phi_d(F_2)(s_1)), \frac{1}{2} d(F_1(s_2), F_2(s_2)) \} \leqslant$ by induction
$\frac{1}{2} d(F_1, F_2)$ □

DEFINITION 4.3.4

1. $\mathscr{D}_d : \mathscr{L} \to \mathscr{PCM}^*$ is defined by $\mathscr{D}_d =$ fixed-point Φ_d.
2. $\mathscr{D} : \mathscr{Prog} \to \mathscr{PCM}^*$ is defined by $\mathscr{D}\ (< d \mid s >) = \mathscr{D}_d(s)$

4.4. Operational semantics = Denotational semantics

THEOREM 4.4.1 $\mathscr{O} = \mathscr{D}$

PROOF We have to show that $\mathscr{O}_d = \mathscr{D}_d$, for all $d \in \mathscr{Decl}$. Since $\mathscr{D}_d$ is the unique fixed-point of Φ_d, it is sufficient to prove that $\Phi_d(\mathscr{O}_d) = \mathscr{O}_d$. This is a direct consequence of lemma 4.2.6.2. □

5. Synchronization

In this section we incorporate a CCS-style synchronization to our language and give a denotational semantics for this language. The most intuitive approach, where for instance we would define

$$\mathcal{D}(c \| \overline{c}) = \{ \left[\begin{matrix} c \\ \overline{c} \end{matrix}\right], \ [\tau] \},$$

leads to a parallel operator that does not satisfy the (necessary, see 4.3) requirement that it be non-distance-increasing.
Consider, for example, the following pomsets.

$$p = [a \to c], \quad p' = [a \to d], \quad q = q' = [\overline{c}].$$

If we would define

$$p \parallel q = \{ \left[\begin{matrix} a \to c \\ \overline{c} \end{matrix}\right], \ [a \to \tau] \}$$

and

$$p' \parallel q' = \{ \left[\begin{matrix} a \to d \\ \overline{c} \end{matrix}\right] \}$$

then $d(p \| q, p' \| q') = 1$, while $d(p, p') \leqslant \frac{1}{2}$ and $d(q, q') = 0 \leqslant \frac{1}{2}$, showing that the operator '$\|$' fails to be non-distance-increasing.
The solution to this problem that we present here is more or less of a mathematical nature; it doesn't have a very clear semantic intuition. Maybe this approach will be a stepping-stone for a more intuitive solution.
Instead of only delivering 'pure' non interleaved outcomes, we extend the denotational semantics with all interleaved outcomes and all intermediate results.
With p and q as given above, we will have

$$p \parallel q = \{ \left[\begin{matrix} a \to c \\ \overline{c} \end{matrix}\right], \ [a \to \tau], \ \left[\begin{matrix} a \nearrow c \\ a \searrow \overline{c} \end{matrix}\right], \ \left[\begin{matrix} a \searrow c \\ \overline{c} \nearrow c \end{matrix}\right], \ [a \to c \to \overline{c}], \ [a \to \overline{c} \to c], \ [\overline{c} \to a \to c] \}$$

and

$$p' \parallel q' = \{ \left[\begin{matrix} a \to d \\ \overline{c} \end{matrix}\right], \ \left[\begin{matrix} a \nearrow d \\ a \searrow \overline{c} \end{matrix}\right], \ \left[\begin{matrix} a \searrow d \\ \overline{c} \nearrow d \end{matrix}\right], \ [a \to d \to \overline{c}], \ [a \to \overline{c} \to d], \ [\overline{c} \to a \to d] \}$$

making $d(p \| q, p' \| q') = \frac{1}{2}$, solving the problem mentioned above.

In subsection 5.1 we define the extended language. In subsection 5.2 we make the new definition of the parallel operator precise and in subsection 5.3 we prove the fact that this operator is non-distance-increasing. We conclude with the denotational semantics and an example in subsection 5.4.

5.1. A language with synchronization

To extend the language with synchronization, assume $\mathscr{A} = \mathscr{I} \cup \mathscr{C}$: the disjoint union of a set of internal actions $(a,b,... \in)\mathscr{I}$ and a set of synchronization actions $(c \in)\mathscr{C}$. Let $^{-}: \mathscr{C} \to \mathscr{C}$ (notation : $\bar{c}$ instead of $^{-}(c)$) be a bijection, such that $\bar{\bar{c}} = c$, yielding the matching synchronization action of c. There is some special element $\tau \in \mathscr{I}$ denoting successful synchronization.

Definition 5.1.1

a. The class $(s \in)\mathscr{L}$ of *statements* is given by

$$s ::= a \mid c \mid x \mid s_1;s_2 \mid s_1+s_2 \mid s_1 \| s_2 \qquad a \neq \tau$$

b. The class $(g \in)\mathscr{L}^g$ of *guarded statements* is given by

$$g ::= a \mid c \mid g;s \mid g_1+g_2 \mid g_1 \| g_2 \qquad a \neq \tau$$

c. The class $(d \in)\mathscr{D}ecl$ of *declarations* consists of mappings from $\mathscr{P}var$ to $\mathscr{L}^g$.

d. The class $(t \in)\mathscr{P}rog$ of *programs* consists of pairs $t \equiv < d \mid s >$ with $d \in \mathscr{D}ecl$ and $s \in \mathscr{L}$.

5.2. The parallel operator

In order to give a semantics for this language, we need to change the definition of the parallel operator ($\|$). Let $\|_{OLD}$ denote the parallel composition defined in section 3.3. The new parallel composition will be defined by taking the result of the old parallel operator and adding some more results. The additional results will be obtained by transforming old results by two kinds of transformation steps : $\xrightarrow{FUSE}$ and $\xrightarrow{AUG}$. Two nodes in a structure, one labeled with c, the other labeled with $\bar{c}$, are taken together in a $\xrightarrow{FUSE}$ step and the label is replaced by a τ. This step models the real synchronization. To solve the problem mentioned in the introduction of this section, we also add structures obtained by adding more causal dependencies in the structure. For this purpose, we define the $\xrightarrow{AUG}$ steps.

Definition 5.2.1

1. For a structure σ and $x_1, x_2 \in X_\sigma$ independent (i.e. $x_1 \not\leq_\sigma x_2 \wedge x_2 \not\leq_\sigma x_1$), we define a new structure $\sigma' = (X_\sigma, \leq_{\sigma'}, \lambda_\sigma)$, where $\leq_{\sigma'} = \leq_\sigma \cup$

$\{ (x, y) \mid x \leqslant_\sigma x_1 \wedge x_2 \leqslant_\sigma y \}$. We will use the notation $\sigma \xrightarrow{AUG(x_1, x_2)} \sigma'$.

2. We define $\xrightarrow{AUG} \subseteq \mathcal{PAM} \times \mathcal{PAM}$ by $p \xrightarrow{AUG} p' \Leftrightarrow \exists \sigma, \sigma' : p = [\sigma] \wedge p' = [\sigma'] \wedge \sigma \xrightarrow{AUG(x_1, x_2)} \sigma'$ for some pair of independent nodes $x_1, x_2 \in X_\sigma$.

REMARKS 5.2.2

1. It is easy to see that σ' is indeed a structure.
2. If $\sigma \xrightarrow{AUG(x_1, x_2)} \sigma'$ and $\phi : \sigma \to \rho$ is an isomorphism then $\exists \rho' : \rho \xrightarrow{AUG(\phi(x_1), \phi(x_2))} \rho'$ and $\phi : \sigma' \to \rho'$ is an isomorphism.
3. From 2. it follows that $\xrightarrow{AUG}$ is well defined.

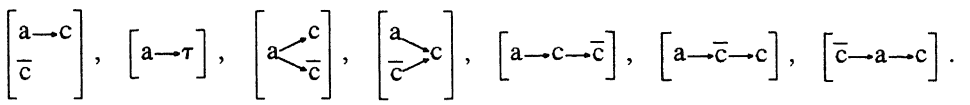
DEFINITION 5.2.3

1. Let σ be a structure. We call (x_1, x_2) a matching pair in σ if $x_1 \not\leqslant_\sigma x_2$, $x_2 \not\leqslant_\sigma x_1$ and $\lambda_\sigma(x_1) \in \mathcal{C}$, $\lambda_\sigma(x_2) \in \mathcal{C}$ and $\overline{\lambda_\sigma(x_1)} = \lambda_\sigma(x_2)$.
   We define a new structure $\sigma' = (X_{\sigma'}, \leqslant_{\sigma'}, \lambda_{\sigma'})$ associated with σ and a matching pair (x_1, x_2), where
   $X_{\sigma'} = X_\sigma \setminus \{ x_2 \}$,
   $\leqslant_{\sigma'} = (\leqslant_\sigma \cap (X_{\sigma'} \times X_{\sigma'})) \cup$
   $\{ (x, y) \mid x \leqslant_\sigma x_1 \wedge x_2 <_\sigma y \} \cup \{ (x, y) \mid x <_\sigma x_2 \wedge x_1 \leqslant_\sigma y \}$ and
   $\lambda_{\sigma'}(x) = \lambda_\sigma(x)$, if $x \neq x_1$, and τ otherwise.
   We will use the notation $\sigma \xrightarrow{FUSE(x_1, x_2)} \sigma'$.
2. We define $\xrightarrow{FUSE} \subseteq \mathcal{PAM} \times \mathcal{PAM}$ by $p \xrightarrow{FUSE} p' \Leftrightarrow \exists \sigma, \sigma' : p = [\sigma] \wedge p' = [\sigma'] \wedge \sigma \xrightarrow{FUSE(x_1, x_2)} \sigma'$ for some matching pair of nodes $x_1, x_2 \in X_\sigma$.

REMARKS 5.2.4

1. It is easy to see that σ' is indeed a structure.
2. If $\sigma \xrightarrow{FUSE(x_1, x_2)} \sigma'$ and $\phi : \sigma \to \rho$ is an isomorphism then $\exists \rho' : \rho \xrightarrow{FUSE(\phi(x_1), \phi(x_2))} \rho'$ and $\phi \restriction X_{\sigma'} : \sigma' \to \rho'$ is an isomorphism.
3. By 2. we have that $\xrightarrow{FUSE}$ is well defined.

EXAMPLES 5.2.5 Let $p_1, \ldots, p_7$ be equal to respectively



Then $p_2, \ldots, p_7$ are all obtained from p_1 by doing one or more $\xrightarrow{AUG}$ or $\xrightarrow{FUSE}$ steps. For instance p_2 is derived from p_1 by doing a $\xrightarrow{FUSE}$ step, p_3 is obtained from p_1 by doing a $\xrightarrow{AUG}$ step (with (x_1, x_2) equal to (the node belonging to a, the node

belonging to $\overline{c}$)) and p_5 can for instance be produced by doing a $\xrightarrow{AUG}$ step from p_3.

The next lemma states the following. If $n \in \mathbb{N}_0$ is fixed and σ can be transformed to ρ by some transformation step then either $\sigma[n] = \rho[n]$ or $\sigma[n]$ can be transformed to a ρ' that is equal to ρ up to level n ($\rho[n] = \rho'[n]$). This will be needed to prove that the parallel operator is non-distance-increasing.

LEMMA 5.2.6 Let $n \in \mathbb{N}_0$ be fixed.

1. Let $\sigma \xrightarrow{AUG(x_1, x_2)} \rho$.
   a. If $lev(x_2) > n$ then $\sigma[n] = \rho[n]$.
   b. If $lev(x_1) \leq n \wedge lev(x_2) \leq n$ then
      $\exists \rho' : \sigma[n] \xrightarrow{AUG(x_1, x_2)} \rho' \wedge \rho'[n] = \rho[n]$.
   c. If $lev(x_1) > n \wedge lev(x_2) \leq n$ then
      $\exists x_1' : x_1' \not\leq_\sigma x_2 \wedge x_2 \not\leq_\sigma x_1' \wedge lev(x_1') = n$:
      $\exists \rho' : \sigma \xrightarrow{AUG(x_1', x_2)} \rho' \wedge \rho'[n] = \rho[n]$.
2. Let $\sigma \xrightarrow{FUSE(x_1, x_2)} \rho$.
   a. If $lev(x_1) > n \wedge lev(x_2) > n$ then $\sigma[n] = \rho[n]$.
   b. If $lev(x_1) \leq n \wedge lev(x_2) \leq n$ then
      $\exists \rho' : \sigma[n] \xrightarrow{FUSE(x_1, x_2)} \rho' \wedge \rho'[n] = \rho[n]$.
   c. If $lev(x_1) > n \wedge lev(x_2) \leq n$ then
      $\exists x_1' : x_1' \not\leq_\sigma x_2 \wedge x_2 \not\leq_\sigma x_1' \wedge lev(x_1') = n$:
      $\exists \rho' : \sigma \xrightarrow{AUG(x_1', x_2)} \rho' \wedge \rho'[n] = \rho[n]$
      [Note that a $\xrightarrow{FUSE}$ step is replaced by a $\xrightarrow{AUG}$ step!].

We omit the proof here because it is only technical and does not give any insight. Moreover, for the most difficult case (2.c) we give an example after the next proposition.

Let $\xrightarrow{A\&F} = \xrightarrow{AUG} \cup \xrightarrow{FUSE}$ and let $\xrightarrow{A\&F}^*$ denote the reflexive transitive closure of $\xrightarrow{A\&F}$.

PROPOSITION 5.2.7 Let $n \in \mathbb{N}_0$ be fixed.
If $p[n] = q[n]$ and $p \xrightarrow{A\&F} p'$ then $p'[n] = q[n]$ or $\exists q' : q \xrightarrow{A\&F} q' \wedge p'[n] = q'[n]$.

PROOF
Case I : $p \xrightarrow{AUG} p'$. Say $\sigma \in p$, $\sigma' \in p'$ and $\sigma \xrightarrow{AUG(x_1, x_2)} \sigma'$.
If $lev(x_2) > n$ then $\sigma'[n] = \sigma[n]$ so $p'[n] = p[n] = q[n]$.
Assume now that $lev(x_2) \leq n$. If also $lev(x_1) \leq n$ then $\sigma[n] \xrightarrow{AUG(x_1, x_2)} \sigma''$ with $\sigma''[n] = \sigma'[n]$. Let $\rho \in q$ and $\phi : \sigma[n] \sim \rho[n]$. Since x_1 and x_2 are incomparable in σ we also have that $\phi(x_1)$ and $\phi(x_2)$ are incomparable in ρ so

$\rho \xrightarrow{AUG(\phi(x_1),\ \phi(x_2))} \rho'$, say, and since $lev(\phi(x_1)) \leqslant n \wedge lev(\phi(x_2)) \leqslant n$, $\rho[n] \xrightarrow{AUG(\phi(x_1),\ \phi(x_2))} \rho''$ with $\rho''[n] = \rho'[n]$. Since $\sigma[n] \sim \rho[n]$ we have $\sigma'' \sim \rho''$ so $\sigma'[n] = \sigma''[n] \sim \rho''[n] = \rho'[n]$. If now $q' = [\rho']$ then $q \xrightarrow{A\&F} q'$ and $p'[n] = [\sigma'[n]] = [\rho'[n]] = q'[n]$.

The last case is $lev(x_1) > n$ and $lev(x_2) \leqslant n$. Then $\exists x_1' : lev(x_1') = n$ and $\sigma \xrightarrow{AUG(x_1',\ x_2)} \sigma''$ and $\sigma''[n] = \sigma'[n]$. So this reduces this case to the previous one.

Case II : $p \xrightarrow{FUSE} p'$. Say $\sigma \in p$, $\sigma' \in p'$ and $\sigma \xrightarrow{FUSE(x_1,\ x_2)} \sigma'$.

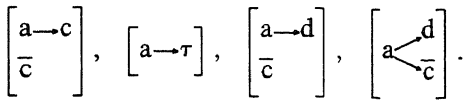
If $lev(x_1) \leqslant n$ and $lev(x_2) \leqslant n$ then $\sigma[n] \xrightarrow{FUSE(x_1,\ x_2)} \sigma''$ and $\sigma''[n] = \sigma'[n]$. Let $\rho \in q$ and $\phi : \sigma[n] \sim \rho[n]$. Since (x_1, x_2) is a matching pair in σ, we have that $(\phi(x_1), \phi(x_2))$ is a matching pair in ρ so $\rho \xrightarrow{FUSE(\phi(x_1),\ \phi(x_2))} \rho'$, say, and since $lev(\phi(x_1)) \leqslant n \wedge lev(\phi(x_2)) \leqslant n$, $\rho[n] \xrightarrow{FUSE(\phi(x_1),\ \phi(x_2))} \rho''$ with $\rho''[n] = \rho'[n]$. Since $\sigma[n] \sim \rho[n]$ we have $\sigma'' \sim \rho''$ so $\sigma'[n] = \sigma''[n] \sim \rho''[n] = \rho'[n]$. If now $q' = [\rho']$ then $q \xrightarrow{A\&F} q'$ and $p'[n] = [\sigma'[n]] = [\rho'[n]] = q'[n]$.

If $lev(x_1) > n$ and $lev(x_2) \leqslant n$ then $\exists x_1' : lev(x_1') = n$ and $\sigma \xrightarrow{AUG(x_1',\ x_2)} \sigma''$ and $\sigma''[n] = \sigma'[n]$. Case I "$\leqslant$" "$\leqslant$" gives $\exists \rho' : \rho \xrightarrow{AUG(\phi(x_1'),\ \phi(x_2))} \rho'$ and $\sigma''[n] = \rho'[n]$ so $\sigma'[n] \sim \rho'[n]$. If now $q' = [\rho']$ then $q \xrightarrow{A\&F} q'$ and $p'[n] = [\sigma'[n]] = [\rho'[n]] = q'[n]$.

Case $lev(x_1) \leqslant n$ and $lev(x_2) > n$: analogous.

If $lev(x_1) > n$ and $lev(x_2) > n$ then $\sigma'[n] = \sigma[n]$ so $p'[n] = p[n] = q[n]$. □

EXAMPLE 5.2.8 We give a little explanation about lemma 5.2.6 and proposition 5.2.7 for the most difficult case namely 5.2.6.2.c and the corresponding Case II ">" "$\leqslant$" of proposition 5.2.7. Let p, p', q, q' be equal to respectively



We have $p \xrightarrow{FUSE} p'$ and $p[1] = q[1]$ and since $p'[1] \neq q[1]$ lemma 5.2.7 guarantees the existence of a q' such that $q \xrightarrow{A\&F} q'$ and $p'[1] = q'[1]$. Indeed the q defined above satisfies $q \xrightarrow{AUG} q'$ and $p'[1] = q'[1]$.

DEFINITION 5.2.9

1. $syn : \mathcal{POM} \to \mathcal{POM}^*$ is defined by $syn(p) = \overline{\{\ q \mid p \xrightarrow{A\&F}{}^{*}\ q\ \}}$.
2. $syn : \mathcal{POM}^* \to \mathcal{POM}^*$ is defined by $syn(P) = \overline{\bigcup \{\ syn(p) \mid p \in P\ \}}$.

3. $\| : \mathcal{POM}^* \times \mathcal{POM}^* \to \mathcal{POM}^*$ is defined by $\| = syn \circ \|_{OLD}$.

The closures are taken to get closed sets and thus elements of $\mathcal{POM}^*$. Moreover, pomsets that contain infinitely many synchronizations are added in this way (see example 5.4.1).

5.3. The parallel operator is non distance increasing

PROPOSITION 5.3.1 $syn : \mathcal{POM} \to \mathcal{POM}^*$ is non distance increasing.

PROOF Let $p, q \in \mathcal{POM}$ such that $p[n] = q[n]$. It suffices to show that $d(\{ p' \mid p \xrightarrow{A\&F}{}^* p' \}, \{ q' \mid q \xrightarrow{A\&F}{}^* q' \}) \leqslant 2^{-n}$ or equivalently $p \xrightarrow{A\&F}{}^* p' \Rightarrow \exists q' : q \xrightarrow{A\&F}{}^* q' \wedge p'[n] = q'[n]$ and vice versa. This is done by induction on the number of steps in which p' is obtained from p. Let us denote this by $p \xrightarrow{A\&F}{}^k p'$. If $k = 0$ then $p' = p$, so we can take $q' = q$. If $p \xrightarrow{A\&F}{}^{k+1} p'$ then $\exists p_k$ such that $p \xrightarrow{A\&F}{}^k p_k \xrightarrow{A\&F} p'$. By induction there exists a $\overline{q'}$ such that $q \xrightarrow{A\&F}{}^* \overline{q'}$ and $p_k[n] = \overline{q'}[n]$. By proposition 5.2.7 we have that either $p'[n] = \overline{q'}[n]$, in which case we can take $q' = \overline{q'}$, or there exists a q' such that $\overline{q'} \xrightarrow{A\&F} q'$ and $q'[n] = p'[n]$. The symmetric case is similar. □

PROPOSITION 5.3.2 $syn : \mathcal{POM}^* \to \mathcal{POM}^*$ is non distance increasing.

PROOF This is a consequence of proposition 5.3.1 and a small adaptation of the appendix. □

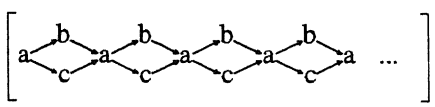
PROPOSITION 5.3.3 $\| : \mathcal{POM}^* \times \mathcal{POM}^* \to \mathcal{POM}^*$ is non distance increasing.

PROOF The composition of two N.D.I. mappings is again N.D.I. □

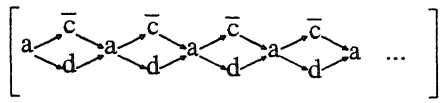
5.4. Denotational semantics

In the previous subsection we showed that $\|$ is a non-distance-increasing mapping. So lemmas 4.3.1.3, 4.3.1.4, and 4.3.1.5 hold in the new setting. We can now give the denotational semantics for the extended language in the same way as we did in subsection 4.3 by substitution of the old $\|$ by the new $\|$.

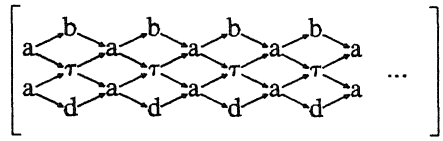
EXAMPLE 5.4.1 Let $d(x) = a;(b \| c);x$ and $d(y) = a;(\overline{c} \| d);y$. Then $\mathcal{D}(< d \mid x >)$ contains for instance :



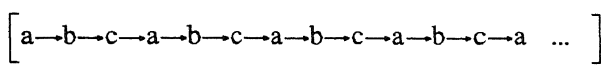
and $\mathscr{D}(< d \mid y >)$ contains for instance :



So $\mathscr{D}(< d \mid x \| y >)$ contains for instance :



but also :



6. Appendix

In this appendix, it is shown that lemma 4.3.1.3 is a consequence of lemma 4.3.1.1 by applying some metric techniques.

LEMMA 6.1 Let M_1, ..., M_n and M be metric spaces.
Let $f : M_1 \times ... \times M_n \to M$ with $\lambda x_i.f(x_1, ..., x_i, ..., x_n) : M_i \to^{\gamma_i} M$.
Then $F : \mathscr{P}_{nc}(M_1) \times ... \times \mathscr{P}_{nc}(M_n) \to \mathscr{P}_{nc}(M)$ defined by
$F(A_1, ..., A_n) = \overline{\{ f(a_1, ..., a_n) \mid a_i \in A_i , i = 1, ..., n \}}$ satisfies
$\lambda A_i.F(A_1, ..., A_i, ..., A_n) : \mathscr{P}_{nc}(M_i) \to^{\gamma_i} \mathscr{P}_{nc}(M)$.
PROOF We have to show that

$$d(F(A_1, ..., A_i, ..., A_n), F(A_1, ..., A'_i, ..., A_n) \leqslant \gamma_i \cdot d(A_i, A'_i)$$

or equivalently :

$$\forall \epsilon > 0 : d(F(A_1, ..., A_i, ..., A_n), F(A_1, ..., A'_i, ..., A_n) \leqslant \gamma_i \cdot d(A_i, A'_i) + \epsilon.$$

Let $x \in F(A_1, ..., A_i, ..., A_n)$. We will show that there exists an $y \in F(A_1, ..., A'_i, ..., A_n)$ such that $d(x, y) \leqslant \gamma_i \cdot d(A_i, A'_i) + \epsilon$ (the other part is analogous).
Since $x \in \overline{\{ f(a_1, ..., a_n) \mid a_i \in A_i , i = 1, ..., n \}}$, there exist $a_1, ..., a_n$ such that $d(x, f(a_1, ..., a_n)) \leqslant \frac{\epsilon}{2}$. By the definition of the Hausdorff distance, $\exists a'_i \in A'_i : d(a_i, a'_i) \leqslant \frac{\epsilon}{2\gamma_i + 1} + d(A_i, A'_i)$. Take $y = f(a_1, ..., a'_i, ..., a_n)$.

$$\begin{aligned}\text{Now } d(x, y) &\leqslant d(x, f(a_1, ..., a_n)) + d(f(a_1, ..., a_i, ..., a_n), f(a_1, ..., a'_i, ..., a_n)) \\ &\leqslant d(x, f(a_1, ..., a_n)) + \gamma_i \cdot d(a_i, a'_i) \\ &\leqslant \frac{\epsilon}{2} + \gamma_i \cdot (\frac{\epsilon}{2\gamma_i + 1} + d(A_i, A'_i)) \leqslant \epsilon + \gamma_i \cdot d(A_i, A'_i).\end{aligned}$$

□

To show lemma 4.3.1.3, let $M_1 = \mathcal{POM} \setminus \{[]\}$ and $M_2 = M = \mathcal{POM}$ and let $f = \bullet\restriction(M_1 \times M_2) : M_1 \times M_2 \to M$. By lemma 4.3.1.1 f satisfies the premise of the lemma with $\gamma_1 = 1$ and $\gamma_2 = \frac{1}{2}$. The derived F is equal to $\bullet$ on $\mathcal{POM}^* \times \mathcal{POM}^*$ restricted to $\mathcal{P}_{nc}(M_1) \times \mathcal{POM}^*$. That is, F is restricted in its first argument to pomset-sets that do not contain the empty pomset. The derived property of F is exactly the one formulated in lemma 4.3.1.3, since $\mathcal{POM}^*$ is an ultra-metric space.

7. References

[ABKR89] P. America, J.W. de Bakker, J.N. Kok, J.J.M.M. Rutten, *Denotational semantics of a parallel object-oriented language,* Information and Computation, Vol. 83, pp. 152-205, 1989.

[AR89] P. America, J.J.M.M. Rutten, *Solving reflexive domain equations in a category of complete metric spaces,* Journal of Computer and System Sciences, Vol 39, nr. 3, pp.343-375, 1989.

[B88] J.W. de Bakker, *Comparative semantics for flow of control in logic programming without logic,* Report CS-R8840, Centre for Mathematics and Computer Science, Amsterdam (1988), to appear in Information and Computation.

[B89] J.W. de Bakker, *Designing concurrency semantics,* in: Information Processing 89, G.X. Ritter (ed.), Elsevier, pp. 591-598, 1989.

[BBKM84] J.W. de Bakker, J.A. Bergstra, J.W. Klop, J.-J.Ch. Meyer, *Linear time and branching time semantics for recursion with merge,* Theoretical Computer Science 34 (1984) 135-156.

[BKMOZ86] J.W. de Bakker, J.N. Kok, J.-J.Ch. Meyer, E.-R. Olderog, J.I. Zucker, *Contrasting themes in the semantics of imperative concurrency,* in Current Trends in Concurrency: Overviews and Tutorials (J.W. de Bakker, W.P. de Roever, G. Rozenberg, eds.), Lecture Notes in Computer Science, Vol. 224, Springer (1986) 51-121.

[BM88] J.W. de Bakker, J.-J.Ch. Meyer, *Metric semantics for concurrency,* BIT 28, pp. 504-529, 1988.

[BRR89] J.W. de Bakker, W.P. de Roever, G. Rozenberg (eds.), *Linear Time, Branching Time and Partial Order,* Proc. REX School/Workshop,

Noordwijkerhout, June 1988, Lecture Notes in Computer Science, Vol. 354, Springer 1989.

[BR89] J.W. DE BAKKER, J.J.M.M. RUTTEN, *Concurrency semantics based on metric domain equations,* Report CS-R8954, Centre for Mathematics and Computer Science, Amsterdam (1989).

[BZ82] J.W. DE BAKKER, J.I. ZUCKER, *Processes and the denotational semantics of concurrency,* Information and Control 54 (1982) 70-120.

[BoCa88] G. BOUDOL, I. CASTELLANI, *Concurrency and atomicity,* Theoretical Computer Science 59 (1988) 25-84.

[Ga89] H. GAIFMAN, *Modeling concurrency by partial orders and nonlinear transition systems,* in Proc. REX School/Workshop, Noordwijkerhout, June 1988, (J.W. de Bakker, W.P. de Roever, G. Rozenberg, eds.), *Linear Time, Branching Time and Partial Order,* Lecture Notes in Computer Science, Vol. 354, Springer (1989), 467-488.

[Gi84] J. GISCHER, *Partial orders and the axiomatic theory of shuffle,* Ph.D. thesis, Stanford University, 1984.

[Gr81] J. GRABOWSKI, *On partial languages,* Fundamenta Informaticae IV.2 (1981) 427-498.

[KR88] J.N. KOK, J.J.M.M. RUTTEN, *Contractions in comparing concurrency semantics,* in Proc. 15th ICALP (T. Lepistö, A. Salomaa, eds.), Lecture Notes in Computer Science, Vol. 317, Springer (1988), 317-332. (To appear in Theoretical Computer Science.)

[MV89] J.-J.Ch. Meyer, E.P. de Vink, *Pomset semantics for true concurrency with synchronization and recursion (extended abstract),* in Proc. MFCS '89 (A Kreczmar & G. Mirkowska, eds.), Lecture Notes in Computer Science, Vol. 379, Springer (1989), 360-369.

[Pr86] V. PRATT, *Modelling concurrency with partial orders,* Int. Journal of Parallel Programming 15 (1986) 33-71.