Elektronische Informationsverarbeitung und Kybernetik
EIK 19 (1983) 1/2, 17–31
(Journal of Information Processing and Cybernetics)

Initial Algebra Specifications for Parametrized Data Types

By *Jan A. Bergstra* and *Jan W. Klop*

Abstract: We consider parametrized data types $\Phi\colon \mathrm{Alg}(\Sigma) \to \mathrm{Alg}(\Delta)$ where Φ is a partial functor from the class of all Σ-algebras (the parameter algebras) to the class of Δ-algebras (the target algebras), for given signatures Σ, Δ with Δ extending Σ. Here it is required that the target algebra is generated by a homomorphic image of the parameter algebra.

For such parametrized data types a general theorem about the existence of initial algebra specifications with conditional equations is proved. The theorem involves the concept of an effectively given parametrized data type.

Introduction

We will discuss the specification theory for persistent parametrized data types according to the definitions in [9].

Our aim is to propose a general necessary and sufficient condition for the existence of an algebraic specification for a given parametrized data type.

We call a persistent parametrized data type $\Phi\colon \mathrm{Alg}(\Sigma) \to \mathrm{Alg}(\Delta)$ *effective* if there exists a uniform algorithm which transforms finite specifications for parameter algebras into finite specifications for target algebras. Especially interesting is the case that $\mathrm{Dom}(\Phi)$ contains all and only semi-computable algebras in a quasi-variety $\mathrm{Alg}(\Sigma, E)$ with E finite.

For such Φ we show that Φ is effective if and only if Φ possesses an algebraic specification (Δ, F) with F an r.e. set of conditional equations.

The following comments are in order.

(i) Of course the definitions of a parametrized data type and its specification as employed here, are by no means the only ones. For further information we refer to the papers [5, 6, 7, 8, 10].

(ii) We preferred not to use the full formalism of category theory; instead we introduce a parametrized data type $\Phi\colon \mathrm{Alg}(\Sigma) \to \mathrm{Alg}(\Delta)$ as a ternary relation containing triples $(\mathcal{A}, \alpha, \mathcal{B})$ where $\mathcal{A} \in \mathrm{Alg}(\Sigma)$, $\mathcal{B} \in \mathrm{Alg}(\Delta)$ and $\alpha\colon \mathcal{A} \to \mathcal{B}|_\Sigma$ is a homomorphism such that
   (1) the relation is closed under taking isomorphic copies of parameter and target algebras, and
   (2) if $(\mathcal{A}, \alpha_1, \mathcal{B}_1)$ and $(\mathcal{A}, \alpha_2, \mathcal{B}_2) \in \Phi$ then $\mathcal{B}_1 \cong \mathcal{B}_2$.

(iii) If one allows auxiliary sorts and functions it is possible to prove that a specification (Δ, F) with F an r.e. set can be transformed into an equivalent but finite specification (Γ, H) with $\Gamma \supseteq \Delta$ and H finite. A similar result is obtained in [1].

(iv) This paper uses a result derived in [1] about the specification of effective parametrized data types with a domain consisting of *minimal* input algebras only.

(v) An informal description of what a parametrized data type is supposed to be can be found in Section 1.4.

1. Preliminaries

1.1. Signatures and algebras

A *signature* is a triple consisting of three listings, one of *sorts*, one of *functions* and one of *constants*. These three listings of Σ are:

(i) **sorts**(Σ), a set of sort names; we will use $s_1, \dots, s_k, s$ as metavariables ranging over **sorts**(Σ),

(ii) **functions**(Σ), a set of function names with arity indication, with typical form $f\colon s_1 \times \dots \times s_k \to s$ for a function name f of arity $(s_1, \dots, s_k, s)$,

(iii) **constants**(Σ) is a set of constant names with sort: $c \in s$. With **constants**$_s(\Sigma)$ we denote Σ's constants for sort s.

Given two signatures Σ, Δ one obtains $\Sigma \cap \Delta$, $\Sigma \cup \Delta$ by taking componentwise intersection resp. union; $\Sigma \subseteqq \Delta$ is meant component-wise as well.

A Σ-algebra $\mathcal{A}$ consists of a family $\{A_s \mid s \in \mathbf{sorts}(\Sigma)\}$ of (possibly empty) sets serving as domains for each sort, equipped with an interpretation for all function and constant names. For $f \in \mathbf{functions}(\Sigma)$, $f\colon s_1 \times \dots \times s_k \to s$, an interpretation is a function $F\colon A_{s_1} \times \dots \times A_{s_k} \to A_s$; $c \in \mathbf{constants}_s(\Sigma)$ is interpreted by an element $C \in A_s$.

The set of Σ-terms is $\mathrm{Ter}(\Sigma)$; the set of *closed* Σ-terms is $\mathrm{Ter}^c(\Sigma)$. (A term is closed if it contains no variables.) The class of all Σ-algebras is $\mathrm{Alg}(\Sigma)$, and the class of all *minimal* Σ-algebras is $\mathrm{ALG}(\Sigma)$. Here an algebra $\mathcal{A}$ is a minimal if it contains no proper subalgebras, equivalently, if $\mathcal{A}$ is isomorphic ($\cong$) to a quotient of a term algebra, equivalently if every element a in $\mathcal{A} \in \mathrm{ALG}(\Sigma)$ is the denotation of a Σ-term.

The concept of a *homomorphism* α between algebras $\mathcal{A}_1, \mathcal{A}_2$ of the same signature is standard. It goes without explicit mention that every map in this paper $\alpha\colon \mathcal{A}_1 \to \mathcal{A}_2$ where $\mathcal{A}_1, \mathcal{A}_2 \in \mathrm{Alg}(\Sigma)$, is a homomorphism.

If $\Sigma \subseteqq \Sigma'$ and $\mathcal{A}' \in \mathrm{Alg}(\Sigma')$, then $\mathcal{A} = \mathcal{A}'|_\Sigma$ is the restriction of $\mathcal{A}'$ to the signature Σ. In this case $\mathcal{A}'$ is also called an expansion of $\mathcal{A}$. The following "*Joint Expansion Property*" is easily verified:

if $\mathcal{A}_i \in \mathrm{Alg}(\Sigma_i)$, $i = 0, 1, 2$, such that $\Sigma_1 \cap \Sigma_2 = \Sigma_0$ and moreover $A_{1,s} \cap A_{2,s} = \emptyset$ for all $s \in \Sigma_1 - \Sigma_0$, $s' \in \Sigma_2 - \Sigma_0$, then there is a unique expansion $\mathcal{A}_1 \sqcup \mathcal{A}_2 \in \mathrm{Alg}(\Sigma_1 \cup \Sigma_2)$ of $\mathcal{A}_1$ and $\mathcal{A}_2$ (cp. Fig. 1).

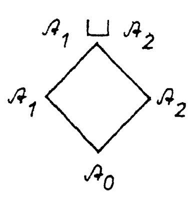

Fig. 1

Instead of $\gamma\colon \mathcal{A} \to \mathcal{B}|_\Sigma$ for $\mathcal{A} \in \mathrm{Alg}(\Sigma)$, $\mathcal{B} \in \mathrm{Alg}(\Delta)$, $\Sigma \subseteqq \Delta$, we will often use the triple notation $(\mathcal{A}, \gamma, \mathcal{B})$. Triples $(\mathcal{A}_i, \gamma_i, \mathcal{B}_i, i = 1, 2, \mathcal{A}_i \in \mathrm{Alg}(\Sigma)$, $\mathcal{B}_i \in \mathrm{Alg}(\Delta)$, $\Sigma \subseteqq \Delta$, are called *congruent* if there are isomorphisms α, β making the diagram in Fig. 2 commute. (In this diagram and in similar ones it is understood that $\gamma\colon \mathcal{A} \to \mathcal{B}|_\Sigma$

$$\begin{array}{ccc} \mathcal{A}_1 & \xrightarrow{\gamma_1} & \mathcal{B}_1 \\ {\scriptstyle\alpha}\downarrow & & \downarrow{\scriptstyle\beta} \\ \mathcal{A}_2 & \xrightarrow[\gamma_2]{} & \mathcal{B}_2 \end{array}$$

Fig. 2

(where $\mathcal{A} \in \mathrm{Alg}(\Sigma)$, $\mathcal{B} \in \mathrm{Alg}(\Delta)$, $\Sigma \subseteqq \Delta$) determines in a natural way an extension $\gamma^*\colon \mathcal{A} \to \mathcal{B}$; we will for simplicity write $\gamma\colon \mathcal{A} \to \mathcal{B}$, or $\mathcal{A} \xrightarrow{\gamma} \mathcal{B}$ in diagrams.)

An important construction is the following one: Let $\Gamma \subseteqq \Delta$ and $\mathcal{B} \in \mathrm{Alg}(\Delta)$. Furthermore, let $A \subseteqq \bigcup_{s \in \mathbf{sorts}(\Gamma)} B_s$, where B_s is the domain in $\mathcal{B}$ corresponding to sort s. Then $\langle \mathcal{B}, \Gamma, A \rangle$ is the *subalgebra generated by A in $\mathcal{B}$ by means of Γ* (i.e. by the Γ-operations and Γ-constants).

In particular, if $\mathcal{A} \in \mathrm{Alg}(\Sigma)$ with $\Sigma \subseteqq \Gamma$ and $A = \bigcup_{s \in \mathbf{sorts}(\Sigma)} A_s$, then we write also $\langle \mathcal{B}, \Gamma, \mathcal{A} \rangle$ instead of $\langle \mathcal{B}, \Gamma, A \rangle$.

1.2. Specifications of algebras

In this paper we will be interested in subclasses of $\mathrm{Alg}(\Sigma)$ of the form $\mathrm{Alg}(\Sigma, E) = \{\mathcal{A} \in \mathrm{Alg}(\Sigma), |\mathcal{A}| = E\}$, where E is a set of *conditional equations*. A conditional equation has the form

$$s_1 = t_1 \wedge \dots \wedge s_k = t_k \to s = t$$

for some $k \geqq 0$ and $s, t, s_i, t_i \in \mathrm{Ter}(\Sigma)$ $(i = 1, \dots, k)$. The conditional equation is *closed* if all terms in it are closed.

We assume the meaning of a conditional equation in an algebra $\mathcal{A}$ to be clear; but note that when a conditional equation contains a free variable ranging over a sort which is empty in $\mathcal{A}$, it is true in $\mathcal{A}$ by definition.

The unique initial term algebra of signature Σ satisfying the set E of conditional equations, is denoted by $I(\mathrm{Alg}(\Sigma, E))$. It is a representant of the isomorphism class of initial algebras in $\mathrm{Alg}(\Sigma, E)$. Isomorphism is denoted by $\cong$.

If E is a set of conditional equations, E° denotes the set of closed equations (so without conditions) derivable from E. An example of such a set of closed Σ-equations is the *congruence* $\equiv_\mathcal{A}$ corresponding to a minimal algebra $\mathcal{A} \in \mathrm{ALG}(\Sigma)$; that is, the set of all closed Σ-equations true in $\mathcal{A}$.

If $\mathcal{A} \in \mathrm{Alg}(\Sigma)$ and for some (Σ', E') with $\Sigma' \supseteqq \Sigma$ it is the case that $\mathcal{A} \cong \mathcal{A}'|_\Sigma$ where $\mathcal{A}' = I(\mathrm{Alg}(\Sigma', E'))$, then we say that $\mathcal{A}$ can be *specified* (using auxiliary sorts and functions) by (Σ', E').

Notation. $(\Sigma', E') \xrightarrow{\Sigma} \mathcal{A}$.

To give an actual specification of $\mathcal{A}$ by (Σ', E') we will insist that also the isomorphism $\alpha\colon \mathcal{A} \to \mathcal{A}'|_\Sigma$ is mentioned.

Notation. $(\Sigma', E') \xrightarrow[\alpha]{\Sigma} \mathcal{A}$. So $(\Sigma', E') \xrightarrow{\Sigma} \mathcal{A}$ is in fact short for $\exists\alpha (\Sigma', E) \xrightarrow[\alpha]{\Sigma} \mathcal{A}$.

1.3. Semi-computable algebras

Notation. If $w = s_1 \times \ldots \times s_k$, where $s_i \in \mathbf{sorts}(\Sigma)$, $i = 1, \ldots, k$, then X_w abbreviates $X_{s_1} \times \ldots \times X_{s_k}$.

The following definition is standard: $\mathcal{A} \in \mathrm{Alg}(\Sigma)$ is *effectively presented* if corresponding to the domains A_s $(s \in \mathrm{sorts}(\Sigma))$ there are mutually disjoint recursive sets Ω_s and surjective maps $\alpha_s: \Omega_s \to A_s$ $(s \in \mathbf{sorts}(\Sigma))$, such that for each function F in $\mathcal{A}$ of type $w \to s$, there is a recursive $f: \Omega_w \to \Omega_s$ which commutes the diagram of Fig. 3 where $\alpha_w(x_1, \ldots, x_k) = (\alpha_{s_1}(x_1), \ldots, a_{s_k}(x_k))$.

$$\begin{array}{ccc} A_w & \xrightarrow{F} & A_s \\ {\scriptstyle \alpha_w}\uparrow & & \uparrow{\scriptstyle \alpha_s} \\ \Omega_w & \xrightarrow{f} & \Omega_s \end{array}$$

Fig. 3

Now $\mathcal{A}$ is *semi-computable* $(\mathcal{A} \in \mathrm{Sca}(\Sigma))$ if in addition for each $s \in \mathbf{sorts}(\Sigma)$ the relation $\equiv_{\alpha_s}$ defined on Ω_k by

$$u \equiv_{\alpha_s} a' \Leftrightarrow \alpha_s(a) = \alpha_s(a')$$

is r.e.

We will need the following fact, proved in [2]:

1.3.1. Lemma. *$\mathcal{A}$ is semi-computable iff $\mathcal{A}$ has a finite specification.*

1.4. We are now able to present a precise definition of a parametrized data type. However, it is worth-while to present an intuition first.

A data type $\mathcal{D}$ is modeled as a heterogeneous Σ-algebra fixed up to isomorphism. Now one can imagine a situation where the properties of a data type $\mathcal{D}$ are only partially known, in such a way that for some subsignature Σ^* of Σ the subalgebra $\mathcal{D}|_{\Sigma^*}$ is not yet specified. As soon as $\mathcal{D}|_{\Sigma^*}$ is specified all of $\mathcal{D}$ is known. Obtaining an instance (data type) from $\mathcal{D}$ is then a matter of substituting a Σ^*-algebra $\mathcal{A}$ (perhaps satisfying some requirements) in $\mathcal{D}(.)$. $\mathcal{A}$ is the parameter algebra and $\mathcal{D}(\mathcal{A})$ is the target algebra. This clearly leads to a mapping from $\mathrm{Alg}(\Sigma^*)$ to $\mathrm{Alg}(\Sigma)$; such a mapping is called a parametrized data type.

A specification of a parametrized data type is then an (incomplete) specification which is augmented by a specification of a parameter algebra in order to obtain the specification of the target algebra.

2. Parametrized data types

For signatures Σ and Δ with $\Sigma \subseteqq \Delta$, a parametrized data type $\Phi: \mathrm{Alg}(\Sigma) \to \mathrm{Alg}(\Delta)$ is a class of triples $(\mathcal{A}, \gamma, \mathcal{B})$ where $\mathcal{A} \in \mathrm{Alg}(\Sigma)$, $\mathcal{B} \in \mathrm{Alg}(\Delta)$ and $\gamma: \mathcal{A} \to \mathcal{B}|_\Sigma$ is a surjective homomorphism such that $\mathcal{B} = \langle \mathcal{B}, \Delta, \mathcal{B}|_\Sigma \rangle$ (i.e. $\gamma(\mathcal{A})$ generates $\mathcal{B}$).

Furthermore, the class Φ must satisfy the following global conditions:

(i) if $(\mathcal{A}, \gamma, \mathcal{B}) \in \Phi$ and $(\mathcal{A}', \gamma', \mathcal{B}')$ is congruent with $(\mathcal{A}, \gamma, \mathcal{B})$, then $(\mathcal{A}', \gamma', \mathcal{B}') \in \Phi$;

(ii) if $(\mathcal{A}, \gamma, \mathcal{B}) \in \Phi$, $(\mathcal{A}', \gamma', \mathcal{B}') \in \Phi$ and $\alpha: \mathcal{A} \to \mathcal{A}'$ is an (injective) homomorphism, then there is an (injective) homomorphism $\beta: \mathcal{B} \to \mathcal{B}'$ such that the diagram in Fig. 4 commutes.

$$\begin{array}{ccc} \mathcal{A} & \xrightarrow[\gamma]{} & \mathcal{B} \\ {\scriptstyle\alpha}\downarrow & & \downarrow{\scriptstyle\beta} \\ \mathcal{A}' & \xrightarrow[\gamma']{} & \mathcal{B}' \end{array}$$

Fig. 4

Furthermore, Φ is called *persistent* if for all $(\mathcal{A}, \gamma, \mathcal{B}) \in \Phi$ the homomorphism γ is injective as well as surjective.

Remark. Note that the class Φ of triples $(\mathcal{A}, \gamma, \mathcal{B})$ determines a partial functor Φ': $\mathrm{Alg}(\Sigma) \to \mathrm{Alg}(\Delta)$, given by the class of pairs $(\mathcal{A}, \mathcal{B})$ such that Φ contains a triple $(\mathcal{A}, \gamma, \mathcal{B})$ for some γ. Since we will need the extra information given by Φ as compared to Φ', we will stick to Φ.

2.1. Effectively given parametrized data types

Let (σ, ε) be a monotonic partial recursive transformation of finite specifications, transforming (Σ', E') into $(\sigma(\Sigma', E'), \varepsilon(\Sigma', E')) = (\Sigma'', E'')$. Here the monotonicity requirement is that $\Sigma'' \supseteq \Sigma'$ and $E'' \supseteq E'$.

Now we say that a parametrized data type Φ: $\mathrm{Alg}(\Sigma) \to \mathrm{Alg}(\Delta)$ is *effectively given* by (σ, ε) if for each triple $(\mathcal{A}, \gamma, \mathcal{B}) \in \Phi$ and for each finite specification $(\Sigma', E') \xrightarrow{\Sigma} \mathcal{A}$ the following triple $(\mathcal{A}', \gamma', \mathcal{B}')$ is congruent to $(\mathcal{A}, \gamma, \mathcal{B})$:

- $\mathcal{A}' = I(\mathrm{Alg}(\Sigma', E'))|_{\Sigma}$,
- $\mathcal{B}' = I(\mathrm{Alg}(\Sigma'', E''))|_{\Delta}$,
- γ': $\mathcal{A}' \to \mathcal{B}'|_{\Sigma}$ is the homomorphism induced by the unique homomorphism

$$\iota: I(\mathrm{Alg}(\Sigma', E')) \to I(\mathrm{Alg}(\Sigma'', E''))|_{\Sigma'}.$$

The diagram of Fig. 5 illustrates this definition.

$$\begin{array}{ccc} & I(\mathrm{Alg}(\Sigma', E')) & \xrightarrow{\iota} & I(\mathrm{Alg}(\Sigma'', E''))|_{\Sigma'} \\ & {\scriptstyle\Sigma}\downarrow & & \downarrow{\scriptstyle\Sigma} \\ \mathcal{A}' = & I(\mathrm{Alg}(\Sigma' E')) & \xrightarrow[\gamma']{} & I(\mathrm{Alg}(\Sigma'', E''))|_{\Sigma} = \mathcal{B}'|_{\Sigma} \end{array}$$

Fig. 5

2.2. Algebraically specified parametrized data types

Φ: $\mathrm{Alg}(\Sigma) \to \mathrm{Alg}(\Delta)$ has an *algebraic specification* if there is a specification (Γ, H) such that for each $(\mathcal{A}, \gamma, \mathcal{B}) \in \Phi$ and for each specification $(\Sigma', E') \xrightarrow{\Sigma} \mathcal{A}$ (with $\Sigma' \cap (\Gamma \cup \Delta) = \Sigma$) the following triple $(\mathcal{A}', \gamma', \mathcal{B}')$ is congruent to $(\mathcal{A}, \gamma, \mathcal{B})$:

- $\mathcal{A}' = I(\mathrm{Alg}(\Sigma', E'))$
- $\mathcal{B}' = I(\mathrm{Alg}(\Sigma' \cup \Gamma, E' \cup \Gamma, E' \cup H))$
- γ' again induced by the unique homomorphism

$$\iota: I(\mathrm{Alg}(\Sigma', E')) \to I(\mathrm{Alg}(\Sigma'', E''|))_{\Sigma'}.$$

The following lemma will play a key role in the sequel.

2.3. Lemma. *Suppose that* $\Phi\colon \mathrm{Alg}(\Sigma) \to \mathrm{Alg}(\Delta)$ *is persistent and effectively given by* (σ, ε) *with* $\mathrm{Dom}(\Phi) = \mathrm{ALG}(\Sigma, E) \cap \mathrm{Sca}(\Sigma)$ *for some finite* E.

Then Φ *has an algebraic specification* (Δ, H) *where* H *is a (possible infinite) set of closed conditional equations.*

Moreover H *is r.e., uniformly in recursive indices for* (σ, ε).

Proof. The proof is given in [1]: Theorem 3.1 (iii) $\Rightarrow$ (i) followed by an application of the Countable Specification Lemma 5.1. (Note that the domain of Φ contains only minimal algebras.) □

3. The specification theorem

In this section we will state our theorem and give an informal outline of the formal proof which occupies Section 4.

3.1. Theorem. *Let* $\Phi\colon \mathrm{Alg}(\Sigma) \to \mathrm{Alg}(\Delta)$ *be a persistent parametrized data type with* $\mathrm{Dom}(\Phi) = \mathrm{Alg}(\Sigma, E) \cap \mathrm{Sca}(\Sigma)$ *for some finite* E. *Then the following are equivalent*:

(i) Φ *is effectively given,*

(ii) Φ *has an algebraic specification* (Δ, H) *where* H *is r.e.*

First we will deal with the easy half (ii) $\Rightarrow$ (i) of the theorem.

Proof of (ii) $\Rightarrow$ (i). Let $(\Sigma', E') \overset{\Sigma}{\to} \mathcal{A}$ be a finite specification of a parameter algebra (with $\Sigma' \cap \Delta = \Sigma$). Then $(\Sigma' \cup \Delta, E' \cup H) \overset{\Delta}{\to} \mathcal{B}$ with $(\mathcal{A}, \gamma, \mathcal{B}) \in \Phi$. Because $\mathcal{B}$ has an r.e. specification, it is semi-computable. Using results from [2] one uniformly computes from a specification $(\Sigma' \cup \Delta, E')$ and an r.e.-index of H a finite specification $(\Sigma^*, E^*) \overset{\Delta}{\to} \mathcal{B}$ (which extends (Σ', E')). —

3.1.1. As to the proof of (i) $\Rightarrow$ (ii), we start with the following observation whose routine proof is omitted. First the

Notation. If $\mathcal{A} \in \mathrm{Alg}(\Sigma)$, then $\langle \mathcal{A} \rangle$ denotes $\langle \mathcal{A}, \Sigma, \emptyset \rangle$, the subalgebra generated by the Σ-operations and constants. Note that $\langle \mathcal{A} \rangle$ is a minimal algebra.

3.1.1.1. Proposition. *Let* $\mathcal{A} \in \mathrm{Alg}(\Sigma)$ *and let* e *be a closed conditional equation. Then* $\mathcal{A} \models e \Leftrightarrow \langle \mathcal{A} \rangle \models e$.

Hence we can reduce satisfaction of an arbitrary conditional equation $e(\vec{x})$ in a Σ-algebra $\mathcal{A}$, to satisfaction of closed conditional equations in some minimal subalgebras of $\mathcal{A}$, as follows:

$$\begin{aligned} \mathcal{A} \models e(\vec{x}) &\Leftrightarrow \forall \vec{a} \in \mathcal{A}\colon \mathcal{A}_{\vec{a}} \models e(\vec{c}) \\ &\Leftrightarrow \forall \vec{a} \in \mathcal{A}\langle \mathcal{A}_{\vec{a}} \rangle \models e(\vec{c}) \,. \end{aligned}$$

Here $\mathcal{A}_{\vec{a}}$ is an expansion of $\mathcal{A}$ with constants $\vec{a}$ corresponding to $\vec{x}$, and $\vec{c}$ are constant symbols for $\vec{a}$.

3.1.2. Secondly, we observe (in Lemma 4.1) that a parametrized data type Φ: $\mathrm{Alg}(\Sigma) \to \mathrm{Alg}(\Delta)$ with $\mathrm{Dom}(\Phi) = \mathrm{Alg}(\Sigma, E) \cap \mathrm{Sca}(\Sigma)$ for some finite E, behaves well w.r.t. substructures of a parameter algebra $\mathcal{A}$, as suggested by Fig. 6.

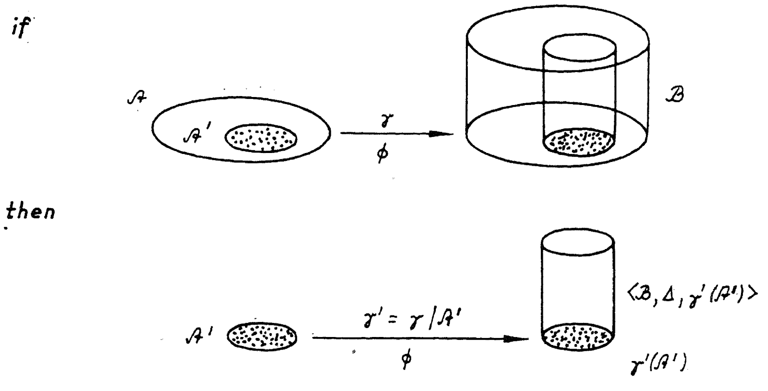

Fig. 6

Here it should be remarked that we must restrict attention to those $\mathcal{A}' \subseteqq \mathcal{A}$ which are still in Dom(Φ). To ensure that, we need only require $\mathcal{A}' \in$ Sca(Σ); for, $\mathcal{A}' \in$ Alg(Σ, E) is trivially satisfied: conditional equations stay valid in a subalgebra. Note that if moreover $\mathcal{A}$ is *finitely generated*, i.e. $\mathcal{A}' = \langle \mathcal{A}, \Sigma, \vec{a} \rangle$ for a finite string $\vec{a}$ of elements in $\mathcal{A}$, then:

$$\mathcal{A} \in \mathrm{Sca}(\Sigma) \Rightarrow \mathcal{A}' \in \mathrm{Sca}(\Sigma) .$$

Indeed, this will be the case we will encounter.

3.1.3. Thirdly, from a parametrized data type Φ: Alg(Σ) $\to$ Alg(Δ) and a given string $\vec{c}$ of new constant symbols for the signature Σ, we define in the obvious way (see Fig. 7) a parametrized data type $\Phi_{\vec{c}}$: Alg($\Sigma_{\vec{c}}$) $\to$ Alg($\Delta_{\vec{c}}$), where $\Sigma_{\vec{c}}$, $\Delta_{\vec{c}}$ is Σ, Δ plus the new constant symbols $\vec{c}$. (Here the $\vec{a}$ are the interpretations of $\vec{c}$ in $\mathcal{A}$, and $\vec{b}$ in $\mathcal{B}$. Furthermore $\alpha(\vec{a}) = \vec{b}$.)

Not surprisingly, if Φ is effectively given by (σ, ε), then the same holds for $\Phi_{\vec{c}}$. (This is proved in 4.3.)

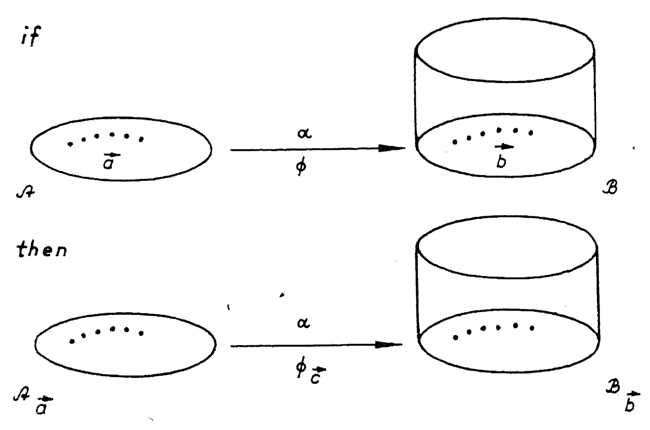

Fig. 7

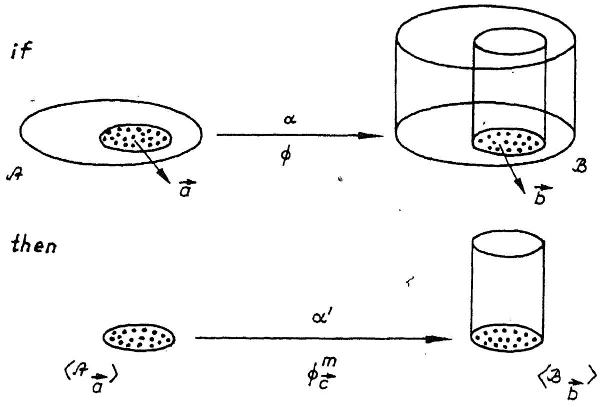

Fig. 8

Now $\mathrm{Dom}(\Phi_{\vec{c}}) = \mathrm{Alg}(\Sigma_{\vec{c}}, E) \cap \mathrm{Sca}(\Sigma_{\vec{c}})$. However, we will be only interested in the restriction of $\Phi_{\vec{c}}$ to the class of minimal algebras of $\mathrm{Alg}(\Sigma_{\vec{c}}, E)$, i.e., the algebras $\langle \mathcal{A}_{\vec{a}} \rangle$ from 3.1.1. Let $\Phi^m_{\vec{c}}$ be this restriction. We already noted in 3.1.2:

$$\mathcal{A} \in \mathrm{Dom}(\Phi) \Rightarrow \mathcal{A}_{\vec{a}} \in \mathrm{Dom}(\Phi_{\vec{c}})$$
$$\Rightarrow \langle \mathcal{A}_{\vec{a}} \rangle \in \mathrm{Dom}(\Phi^m_{\vec{c}}) = \mathrm{ALG}(\Sigma_{\vec{c}}, E) \cap \mathrm{Sca}(\Sigma_{\vec{c}}) .$$

So $\Phi^m_{\vec{c}}$ is as in Fig. 8.

3.1.4. In order to deal with all conditional equations $e(\vec{x})$ (where $\vec{x}$ might be arbitrarily long), we will use a countable set C of fresh constant symbols for Σ from which the $\vec{c}$ are taken.

It may be clear at this stage that the family of all $\Phi^m_{\vec{c}}$ ($\vec{c} \subseteqq C$) determines the original Φ. (Section 4 proves this rigorously.) Moreover, $\Phi^m_{\vec{c}}$ satisfies precisely the assumptions of Lemma 2.3; it is also effectively given by (σ, ε) and the domain has the required form. The persistency is obvious.

Hence $\Phi^m_{\vec{c}}$ has an algebraic specification $(\Delta_{\vec{c}}, H_{\vec{c}})$ where $H_{\vec{c}}$ is an r.e. set of closed conditional equations.

Now we remember that the $\vec{c}$'s play in fact the role of variables (see 3.1.1); so replacing the $\vec{c}$ again by corresponding variables x, we get $(\Delta, H_{\vec{x}/\vec{c}})$. As one may expect, taking together all these pieces of specifications to

$$(\Delta, \bigcup_{\vec{c} \subseteq C} H_{\vec{x}/\vec{c}}) = (\Delta, H)$$

yields the desired specification of Φ. The proof that (Δ, H) specifies Φ correctly, requires some more work however:

3.1.5. Consider the diagram in Fig. 9 where $(\mathcal{A}, \gamma, \mathcal{B}) \in \Phi$. We have to prove that $\mathcal{B}'|_\Delta \cong \mathcal{B}$. Now without loss of generality, we may take $\mathcal{A}'$ and $\mathcal{B}$ such that we can

$$\begin{array}{ccccc} I(\mathrm{Alg}(\Sigma', E')) = & \mathcal{A}' & \longrightarrow & \mathcal{B}' & = I(\mathrm{Alg}(\Sigma' \cup \Delta, E' \cup H)) \\ & \downarrow \Sigma & & ? \downarrow \Delta & \\ & \mathcal{A} & \xrightarrow[\Phi]{\gamma} & \mathcal{B} & \end{array}$$

Fig. 9

appeal to the "Joint Expansion Property" in Section 1.1 and the joint expansion $\mathcal{A}' \sqcup \mathcal{B}$ can be formed. So, trivially, $(\mathcal{A}' \sqcup \mathcal{B})|_\Delta = \mathcal{B}$, and we must only prove that

$$\mathcal{A}' \sqcup \mathcal{B} \cong \mathcal{B}' = I(\mathrm{Alg}(\Sigma' \cup \Delta, E' \cup H)) .$$

In other words, we must prove the correctness of the specification

$$(\Sigma' \sqcup \Delta, E' \cup H) \xrightarrow{\Sigma' \cup \Delta} \mathcal{A}' \sqcup \mathcal{B} .$$

This amounts to prove

(i) soundness: $\mathcal{A}' \sqcup \mathcal{B} \models E' \cup H$;

(ii) completeness: $\mathcal{A}' \sqcup \mathcal{B} \models s = t \Rightarrow E' \cup H \models s = t$, for all s, $t \in \mathrm{Ter}^c(\Sigma' \cup \Delta)$.

We prove (i) in Section 4.3; it follows straightforwardly from the definition of H.

To prove (ii), consider Fig. 10.

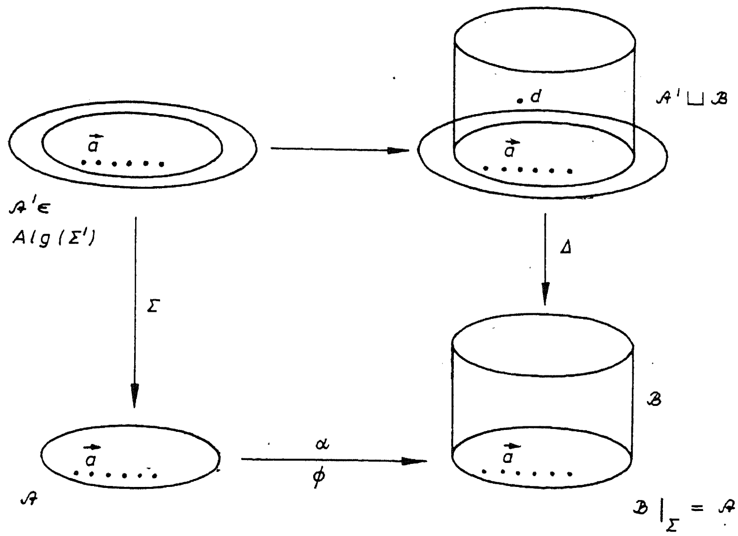

Fig. 10

Since $\mathcal{A}'$ is minimal, and $\mathcal{B}$ is generated from $\mathcal{B}|_\Sigma = \alpha(\mathcal{A}) = \mathcal{A}$, also $\mathcal{A}' \sqcup \mathcal{B}$ is minimal. I.e. every element in $\mathcal{A}' \sqcup \mathcal{B}$ is the denotation of a $(\Sigma' \cup \Delta)$-term. Something more can be said: since the $\vec{a}$ are denotated by Σ'-terms $\vec{s}$, the element d (generated from $\vec{a}$ by Δ-operations and constants) is denotated by a "$\Delta(\Sigma')$-term" $t(\vec{s})$, that is a Δ-term $t(\vec{x})$ in which the Σ'-terms $\vec{s}$ are substituted for $\vec{x}$.

Now if we can prove

(1) the completeness for the restricted class of $\Delta(\Sigma')$-terms and moreover,

(2) that each $(\Sigma' \cup \Delta)$-term is provably (from $E' \cup H$) equal to a $\Delta(\Sigma')$-term,

we are through. The proof of (1) is in Section 4.5, and of (2) in 4.7.

4. Proof of the specification theorem

In this section we will give the formal details of the proof of Theorem 3.1 (ii) $\Rightarrow$ (i) which we have already outlined in Section 3.

Let Φ be an effective parametrized data type with $\mathrm{Dom}(\Phi) = \mathrm{Alg}(\Sigma, E) \cap \mathrm{Sca}(\Sigma)$ for some finite E, effectively given by (σ, ε).

We start with a lemma that explains the effect of Φ on structures embedded in one another.

4.1. Lemma. *Let* $(\mathcal{A}, \gamma, \mathcal{B}) \in \Phi$, $\mathcal{A}' \in \mathrm{Sca}(\Sigma)$ *and* $\mathcal{A}' \subseteqq \mathcal{A}$. *Then* $(\mathcal{A}', \gamma', \mathcal{B}') \in \Phi$ *with* $\gamma' = \gamma \restriction \mathcal{A}'$ *and* $\mathcal{B}' = \langle \mathcal{B}, \Delta, \gamma'(\mathcal{A}') \rangle$.

Proof. Because $\mathcal{A}' \subseteqq \mathcal{A}$ and $\mathcal{A}' \models E$, together with $\mathcal{A}' \in \mathrm{Sca}(\Sigma)$ one finds $\mathcal{A}' \in \mathrm{Dom}(\Phi)$. So there exist $\gamma^*, \mathcal{B}^*$ with $(\mathcal{A}', \gamma^*, \mathcal{B}^*) \in \Phi$.

By (ii) of the definition of parametrized data type (Section 2) and the existence of an injective i embedding $\mathcal{A}'$ in $\mathcal{A}$ one derives the existence of λ such that the diagram in Fig. 11 commutes with $\gamma' = \gamma \circ i$ and λ injective.

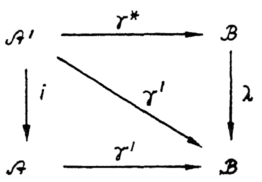

Fig. 11

Observe that $\mathcal{B}^* = \langle \mathcal{B}^*, \Delta, \gamma^*(\mathcal{A}') \rangle$ by definition of parametrized data type, and that $\lambda(\mathcal{B}^*) = \langle \mathcal{B}, \Delta, \lambda\gamma^*(\mathcal{A}') \rangle = \langle \mathcal{B}, \Delta, \gamma \circ i(\mathcal{A}') \rangle = \mathcal{B}'$. It follows that the diagram of Fig. 12 displays a congruence, whence $(\mathcal{A}', \gamma', \mathcal{B}') \in \Phi$. □

$$\begin{array}{ccc} \mathcal{A}' & \xrightarrow{\gamma^*} & \mathcal{B}^* \\ \downarrow{\scriptstyle i} & & \downarrow \\ \mathcal{A}' & \xrightarrow{\gamma'} & \mathcal{B}' \end{array}$$

Fig. 12

4.2. Let C be a set of new constants for sorts of Σ, not occurring in Δ, in such a way that for each sort countably many new constants are introduced.

Furthermore, let $\Sigma_{\vec{c}}, \Delta_{\vec{c}}$ denote the result of augmenting Σ, Δ with a finite subset $\vec{c}$ of C. For finite $\vec{c} \subseteqq C$ we define a parametrized data type $\Phi_{\vec{c}}$ with domain $\mathrm{Alg}(\Sigma_{\vec{c}}, E) \cap \mathrm{Sca}(\Sigma_{\vec{c}})$ and range in $\mathrm{Alg}(\Delta_{\vec{c}})$ as follows:

$$(\mathcal{A}, \gamma, \mathcal{B}) \in \Phi_{\vec{c}} \quad \text{iff} \quad \begin{array}{l} \text{(i) } \mathcal{A} \in \mathrm{Alg}(\Sigma_{\vec{c}}, E) \cap \mathrm{Sca}(\Sigma_{\vec{c}}), \mathcal{B} \in \mathrm{Alg}(\Delta_{\vec{c}}) \\ \text{(ii) } (\mathcal{A}|_{\Sigma}, \gamma, \mathcal{B}|_{\Delta}) \in \Phi . \end{array}$$

4.3. Restricting $\Phi_{\vec{c}}$ to $\mathrm{ALG}(\Sigma_{\vec{c}}, E) \cap \mathrm{Sca}(\Sigma_{\vec{c}})$ we obtain a parametrized data type $\Phi_{\vec{c}}^m$ with range in $\mathrm{ALG}(\Delta_{\vec{c}})$. (Here the target algebras are indeed minimal, because they are generated from minimal parameter algebras.) Now $\Phi_{\vec{c}}^m$ turns out to be effectively given by (σ, ε), just as Φ itself is. This is evident from the diagram in Fig. 13. (Here it is essential that (σ, ε) is monotonic, from which it follows that $\Sigma'' \supseteqq \Sigma_{\vec{c}} \cup \Delta = \Delta_{\vec{c}}$ because $\Sigma' \supseteqq \Sigma_{\vec{c}}$.)

$$\begin{array}{ccc} (\Sigma', E') & \xrightarrow{(\sigma,\varepsilon)} & (\Sigma'', E'') \\ \downarrow{\scriptstyle \Sigma_{\vec{c}}} & & \downarrow{\scriptstyle \Delta_{\vec{c}}} \\ \mathcal{A} & \xrightarrow{\Phi_{\vec{c}}} & \mathcal{B} \\ \downarrow{\scriptstyle \Sigma} & & \downarrow{\scriptstyle \Delta} \\ \mathcal{A}' & \xrightarrow{\Phi} & \mathcal{B}' \end{array}$$

Fig. 13

4.4. Applying Lemma 2.3 we obtain a specification $(\Delta_{\vec{c}}, H_{\vec{c}})$ for $\Phi^m_{\vec{c}}$ with $H_{\vec{c}}$ consisting of an r.e. set of closed conditional equations. $H_{\vec{c}}$ is uniformly r.e. in $(\vec{c}, \sigma, \varepsilon)$.

Let x_c be a new variable for each $c \in C$ of the same sort. Write $H_{\vec{c}} = \{e^i_{\vec{c}} \mid i \in \omega\}$, and let $e^i_{\vec{x}/\vec{c}}$ be the result of substituting x_c for each occurrence of a constant symbol c (from C) in $e^i_{\vec{c}}$.

Obtain $H_{\vec{x}/\vec{c}} = \{e^i_{\vec{x}/\vec{c}} \mid i \in \omega\} = \{e^i_{\vec{x}/\vec{c}} \mid e^i_{\vec{c}} \in H_{\vec{c}}\}$. Note that $H_{\vec{x}/\vec{c}}$ is a set of conditional equations over the signature Δ. Taking the union of all specifications thus obtained one finds (Δ, H) with

$$H = \bigcup_{\vec{c} \subseteq C} H_{\vec{x}/\vec{c}} .$$

From the uniformity of finding $H_{\vec{c}}$ from $\vec{c}$ it follows that H is r.e. Thus (Δ, H) is a specification of the required format.

4.5. Claim. (Δ, H) specifies Φ.

To show this, let (Σ', E') be a finite specification for $\mathcal{A} \in \mathrm{Dom}(\Phi)$, with $\Sigma' \cap \Delta = \Sigma$. Choose $(\mathcal{A}, \gamma, \mathcal{B}) \in \Phi$. We must establish that the triples $(\mathcal{A}, \gamma, \mathcal{B})$ and $\big(I(\mathrm{Alg}(\Sigma', E'))|_\Sigma, \iota, I(\mathrm{Alg}(\Sigma' \cup \Delta, E' \cup H))|_\Delta\big)$ are congruent.

We may assume that $\mathcal{A}$ is identical to $\mathcal{A}'|_\Sigma$ with $\mathcal{A}' = I\big(\mathrm{Alg}(\Sigma', E')\big)$ and that $\mathcal{B}|_\Sigma = \mathcal{A}$ (whence $\gamma = \mathrm{id}$) and further that the domains corresponding to sorts of $\mathcal{A}'$ and $\mathcal{B}$ not named in Σ are pairwise disjoint.

Let $\mathcal{A}' \sqcup \mathcal{B}$ be the joint expansion of $\mathcal{A}'$ and $\mathcal{B}$. Note that $\mathcal{A}' \sqcup \mathcal{B}$ is a minimal $(\Sigma' \cup \Delta)$-algebra. To prove

$$(\Sigma' \cup \Delta, E' \cup H) \xrightarrow{\Sigma' \cup \Delta} \mathcal{A}' \sqcup \mathcal{B}$$

it suffices to derive soundness and completeness of $E' \cup H$.

(i) Soundness. Let $e \in E' \cup H$. If $e \in E'$ then $\mathcal{A}' \models e$ and so $\mathcal{A}' \sqcup \mathcal{B} \models e$. If $e \in H$, choose $\vec{c} \subseteq C$ such that $e = e_{\vec{x}/\vec{c}} \in H_{\vec{x}/\vec{c}}$. Take a set of values $\vec{a}$ in $\mathcal{A}' \sqcup \mathcal{B}$ of suitable sorts corresponding to $\vec{c}$. If this is impossible because one of the constants c_i is of a sort that is empty in $\mathcal{A}' \sqcup \mathcal{B}$, then e is trivially satisfied in $\mathcal{A}' \sqcup \mathcal{B}$. Note that $\vec{a}$ must be from $\mathbf{sorts}(\Sigma)$; hence $\vec{a} \subseteq \mathcal{A} \subseteq \mathcal{A}' \sqcup \mathcal{B}$. We will show that $\mathcal{A}' \sqcup \mathcal{B}$ satisfies e in $\vec{a}$, i.e.

$$(\mathcal{A}' \sqcup \mathcal{B})_{\vec{a}} \models e(\vec{c}) .$$

Now consider $\langle \mathcal{A}_{\vec{a}} \rangle$ and $\langle \mathcal{B}_{\vec{a}} \rangle$. From Lemma 4.1 and the definition of $\Phi^m_{\vec{c}}$ we find that

$$(\langle \mathcal{A}_{\vec{a}}, \mathrm{id}, \langle \mathcal{B}_{\vec{a}} \rangle) \in \Phi^m_{\vec{c}} .$$

Because H_c specifies $\Phi^m_{\vec{c}}$, we have $\langle \mathcal{B}_{\vec{a}} \rangle \models H_{\vec{c}}$. Especially $\langle \mathcal{B}_{\vec{a}} \rangle \models e(\vec{c})$; and since $\langle \mathcal{B}_{\vec{a}} \rangle \subseteq (\mathcal{A}' \sqcup \mathcal{B})_{\vec{a}}$, also $(\mathcal{A}' \sqcup \mathcal{B})_{\vec{a}} \models e(\vec{c})$. □

(ii) Completeness for $\Delta(\Sigma')$-terms. Let $\mathcal{A}' \sqcup \mathcal{B} \models t = r$ where $t = r$ is a closed equation. If $t, r \in \mathrm{Ter}^c(\Sigma')$, there is no problem: since E' specifies $\mathcal{A}'$, we have $E' \vdash t = r$. Otherwise, we restrict our attention to closed equations $t = r$ of the form $t = t(\tau_1, \dots, \tau_k)$, $r = r(\tau_1, \dots, \tau_k)$ where $t(x_1, \dots, x_k)$, $r(x_1, \dots, x_k) \in \mathrm{Ter}(\Delta)$ and $\tau_i \in \mathrm{Ter}(\Sigma')$, $i = 1, \dots, k$. Here it is not required that all x_i $(i = 1, \dots, k)$ do occur in $t(\vec{x})$ and $r(\vec{x})$.

(Such t, r are called $\Delta(\Sigma')$-terms; see Section 4.7.) Moreover, we require the $\vec{x}$ to be variables for Σ-sorts.

So suppose $\mathcal{A}' \sqcup \mathcal{B} \models t = r$; we will prove that $E' \cup H \models t = r$. Let $\vec{a} = (a_1, ,,, , a_k)$ be the values of $(\tau_1, .. , \tau_k)$ in $\mathcal{A}$; they are also the values of $(\tau_1, \dots, \tau_k)$ in $\mathcal{B}$ and in $\mathcal{A}' \sqcup \mathcal{B}$. As before, $\mathcal{A}_{\vec{a}}$ and $\mathcal{B}_{\vec{a}}$ are the expansions of $\mathcal{A}, \mathcal{B}$ by adding $\vec{a}$ as constants. The corresponding signatures are $\Sigma_{\vec{c}}$ resp. $\Delta_{\vec{c}}$. Further, $\langle \mathcal{A}_{\vec{a}} \rangle$ and $\langle \mathcal{B}_{\vec{a}} \rangle$ are again the minimal substructures. From $\mathcal{A}' \sqcup \mathcal{B} \models t = r$ we have $\mathcal{B} \models t = r$, hence $\mathcal{B}_{\vec{a}} \models t(\vec{c}) = r(\vec{c})$ and $\langle \mathcal{B}_{\vec{a}} \rangle \models t(\vec{c}) = r(\vec{c})$ (Proposition 3.1.1.1).

Let $=_{\vec{c}}$ abbreviate $=^{\langle \mathcal{A}_{\vec{a}} \rangle}$. Clearly $(\Sigma_{\vec{c}}, \equiv_{\vec{c}})$ specifies $\langle \mathcal{A}_{\vec{a}} \rangle$; and because $(\Delta_{\vec{c}}, H_{\vec{c}})$ specifies $\Phi^m_{\vec{c}}$ we have the diagram of Fig. 14.

$$\begin{array}{ccc} (\Sigma_{\vec{c}}, \equiv_{\vec{c}}) & \longrightarrow & (\Delta_{\vec{c}}, \equiv_{\vec{c}} \cup H_{\vec{c}}) \\ \downarrow \Sigma_{\vec{c}} & & \downarrow \Delta_{\vec{c}} \\ \langle \mathcal{A}_{\vec{a}} \rangle & \xrightarrow[\Phi^m_{\vec{c}}]{} & \langle \mathcal{B}_{\vec{a}} \rangle \end{array}$$

Fig. 14

From $\langle \mathcal{B}_{\vec{a}} \rangle \models t(\vec{c}) = r(\vec{c})$ it follows that $=_{\vec{c}} \cup H_{\vec{c}} \vdash t(c) = r(c)$. A fortiori: $=_{\vec{c}} \cup H_{\vec{x}/\vec{c}} \vdash t(\vec{c}) = r(\vec{c})$. Now let $=_{\vec{\tau}/\vec{c}}$ be the result of substituting τ_i for c_i $(i = 1, \ldots, k)$ in the equations in $\equiv_{\vec{c}}$. Then also

$$\equiv_{\vec{\tau}/\vec{c}} \cup H_{\vec{x}/\vec{c}} \vdash t(\vec{\tau}) = r(\vec{\tau}) .$$

Now the equations in $\equiv_{\vec{\tau}/\vec{c}}$ are closed Σ'-equations, true in $\mathcal{A}'$; hence they are derivable from E', the specification of $\mathcal{A}'$. So we have

$$E' \cup H_{\vec{x}/\vec{c}} \vdash t(\vec{\tau}) = r(\vec{\tau}). \ \square$$

4.6. Intermezzo: $\Sigma_1(\Sigma_2)$-terms

Let Σ_1, Σ_2 be extension signatures of Σ_0 such that $\Sigma_1 \cap \Sigma_2 = \Sigma_0$. We will define $\mathrm{Ter}(\Sigma_1(\Sigma_2))$, a subset of $\mathrm{Ter}(\Sigma_1 \cup \Sigma_2)$; and for $t \in \mathrm{Ter}(\Sigma_1 \cup \Sigma_2)$ we will define the $\Sigma_1 \mid \Sigma_2$-*degree* of t. In a $(\Sigma_1 \cup \Sigma_2)$-term t the symbols (i.e. the names of functions and constants) from Σ_0, Σ_1, Σ_2 can occur in a complex "mixed" fashion, see Example 4.6.4; the $\Sigma_1 \mid \Sigma_2$-degree is a measure of this complexity.

Let $t \in \mathrm{Ter}(\Sigma_1 \cup \Sigma_2)$ and let Tree(t) be its formation tree, written such that the head operator of t is the top label of the tree. We will refer to the symbols from Σ_0 as 0-symbols, from $\Sigma_1 - \Sigma_0$ as I-symbols and from $\Sigma_2 - \Sigma_0$ as II-symbols. Here 0, I, II are called labels of symbols. Now to each branch α in Tree(t) we associate the tuple of labels of the symbols occurring in α, "reading" α starting at the top of Tree(t) (see Example 4.6.4). From each such tuple, e.g. (I, 0, 0, II, I, 0, I, II, 0), we compute the number of alternations from a I- to a II-label and vice versa, disregarding the 0-labels. In the example just given, this *alternation number* is 3.

4.6.1. Definition. The $\Sigma_1 \mid \Sigma_2$-degree of t is the multiset of alternation numbers of all branches in Tree(t). The degrees are ordered by the usual multiset ordering.

4.6.2. Definition. $\mathrm{Ter}(\Sigma_1(\Sigma_2))$, the set of $\Sigma_1(\Sigma_2)$-terms, is the union of $\mathrm{Ter}(\Sigma_2)$ and the set of results $t(\vec{s})$ of substitution of Σ_2-terms $\vec{s}$ into Σ_1-terms $t(\vec{x})$.

4.6.3. Remark. (i) $\mathrm{Ter}(\Sigma_1 \cup \Sigma_2) \supseteqq \mathrm{Ter}(\Sigma_1\Sigma_2)) \supseteqq \mathrm{Ter}(\Sigma_1) \cup \mathrm{Ter}(\Sigma_2)$.

(ii) t is a $\Sigma_1(\Sigma_2)$-term iff in Tree(t) no I-symbol occurs below a II-symbol. (So along each branch there is at most one alternation allowed, viz. from a I- to a II-symbol, disregarding 0-symbols.)

4.6.4. Example. Σ_0 has sorts s_0, functions F_0: $s_0 \to s_0$, constants $C_0 \in s_0$;
$\Sigma_1 - \Sigma_0$ has sorts s_1, functions F_{I}: $s_0 \times s_0 \to s_1$;
$\Sigma_2 - \Sigma_0$ has sorts s_2, functions F_{II}: $s_2 \times s_0 \times s_0 \to s_0$, $F'_{\mathrm{II}}: s_0 \to s_0$, $F''_{\mathrm{II}}: s_2 \times s_0 \to s_0$, constants $C_{\mathrm{II}} \in s_2$.

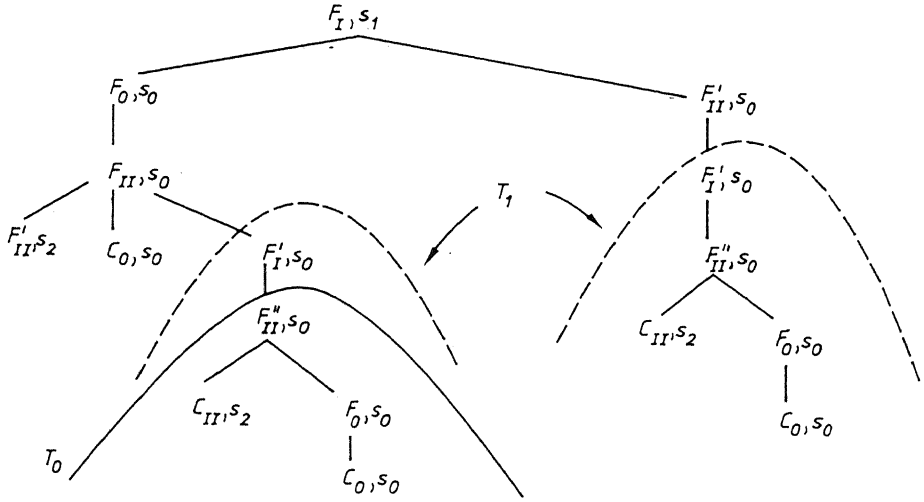

Fig. 15

Let $t \in \text{Ter}(\Sigma_1 \cup \Sigma_2)$ have the tree of Fig. 15 (where next to each function and constant symbol also its target sort is indicated).

Here the tuple corresponding to e.g. the rightmost branch is (I, II, I, II, 0, 0), with alternation number 3. Now the $\Sigma_1 \mid \Sigma_2$-degree of t is $\{1, 1, 3, 3, 3, 3\}$.

4.6.4.1. Remark. Note that if a subterm having the tree T_0 (as indicated in Example 4.6.4), denoting an s_0-element, is replaced by a Σ_0-term denoting the same element (if such a term exists), then this elimination of the "foreign" II-symbols F'_{II}, C_{II} results in a decreased $\Sigma_1 \mid \Sigma_2$-degree, viz. $\{1, 1, 2, 2, 3, 3\}$. Furthermore, if the twice occurring subtree T_1 is replaced by a Σ_0-term, the result would be a $\Sigma_1(\Sigma_2)$-term.

It is important to note the following obvious fact:

4.6.5. Proposition. *If in a branch α of* Tree(t), $t \in \text{Ter}(\Sigma_1 \cup \Sigma_2)$, *a* II-*symbol F_{II} is followed immediately by a* I-*symbol G_I (disregarding* 0-*symbols), i.e. the tuple of α is*

$$(\text{———}, \text{II}, 0, 0, \dots, 0, \text{I}, \text{———}) \qquad (k \geq 0 \text{ times } 0)$$

where the displayed II, I *are the labels of F_{II}, G_I, then the target sort of G_I must be a Σ_0-sort.* □

4.7. It remains to be shown that each $(\Sigma' \cup \Delta)$-term is provably (from $E' \cup H$) equal to some $\Delta(\Sigma')$-term.

Let $t \in \text{Ter}(\Sigma' \cup \Delta)$. Consider Tree($t$). If $t \notin \text{Ter}(\Delta(\Sigma'))$, then there is a $(\Delta - \Sigma)$-function or constant symbol, say D, occurring below a $(\Sigma' - \Sigma)$-function or constant symbol, say S.

Now we can find in Tree(t) a pair S, D such that (cp. Fig. 16)

(i) D is below S,
(ii) S is immediately followed by D (disregarding Σ-symbols),
(iii) the pair S, D is a lowest pair with these properties.

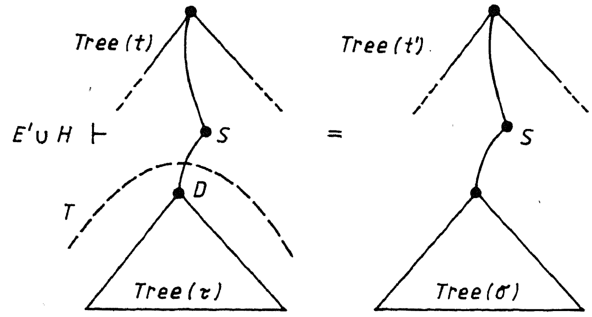

Fig. 16

Then, as we observed in Proposition 4.6.5 the target sort of D must be a Σ-sort. Let T be the subtree headed by D and let τ be the corresponding term. Since τ denotes an element of a Σ-sort, $\mathcal{A}' \sqcup \mathcal{B} \models \tau = \sigma$ for some $\sigma \in \mathrm{Ter}(\Sigma')$. Noting that $\sigma, \tau \in \mathrm{Ter}(\Delta(\Sigma'))$, we have by the completeness of $E' \cup H$ for $\Delta(\Sigma')$-terms, as proved in 4.5:

$$E' \cup H \vdash \tau = \sigma .$$

Now let t' be t where τ is replaced by σ. Then also

$$E' \cup H \vdash t = t' ,$$

and the $\Delta \mid \Sigma'$-degree of t' is less than that of t. Continuing this procedure we find

$$E' \cup H \vdash t = t' = t'' = \ldots = s$$

for some $\Delta(\Sigma')$-term s. □

References

[1] *Bergstra, J. A., J. W. Klop*, Algebraic specifications for parametrized data types with minimal parameter and target algebras. In: Proc. 9th ICALP; Lecture Notes in Computer Science, 140; Springer, 1982; pp. 23–34.

[2] *Bergstra, J. A., J. V. Tucker*, Algebraic specifications of computable and semicomputable data structures. Mathematical Centre, Department of Computer Science Research Report IW 115, Amsterdam 1979.

[3] *Bergstra, J. A., J. V. Tucker*, A characterisation of computable data types by means of a finite equational specification method. In: Proc. 7th ICALP; Lecture Notes in Computer Science, 85; Springer, 1980; pp. 76–90.

[4] *Burstall, R. M., J. A. Goguen*, An informal introduction to specifications using CLEAR. Lecture notes for the International Summer School on theoretical foundations of programming methodology, Munich 1981.

[5] *Ehrich, H.-D.*, On the theory of specification, implementation and parametrization of abstract data types. Research Report, Dortmund 1978.

[6] *Ehrig, H. E., H.-J. Kreowski, J. W. Thatcher, E. G. Wagner, J. B. Wright*, Parameterized data types in algebraic specification languages. In: Proc. 7th ICALP; Lecture Notes in Computer Science, 85; Springer, 1980; pp. 157–168.

[7] *Ganzinger, H.*, Parametrized specifications: parameter passing and optimizing implementation. Report TUM-18110, Technische Universität München, August 1981.

[8] *Kaphengst, H., H. Reichel*, Algebraische Algorithmentheorie. VEB Robotron, WIB, Dresden 1971.

[9] *Thatcher, J. W., E. G. Wagner, J. B. Wright*, Data type specification: parametrization and the power of specification techniques. In: Proc. SIGACT 10th Annual Symp. on Theory of Computing, May 1978; pp. 119–132.

[10] *Wirsing, M.*, An analysis of semantic models for algebraic specifications. Lecture notes for the International Summer School on theoretical foundations of programming methodology, Munich 1981.

Kurzfassung

Es werden parametrisierte Datentypen untersucht, die mathematisch als partielle Funktoren $\Phi: \mathrm{Alg}(\Sigma) \to \mathrm{Alg}(\Delta)$ verstanden werden, wobei $\Sigma \subseteqq \Delta$ vorausgesetzt ist. Darüber hinaus wird gefordert, daß für jede Σ-Algebra $\mathcal{A} \in \mathrm{Dom}(\Phi)$ ein surjektiver Homomorphismus

$$\gamma_{\mathcal{A}}: \mathcal{A} \to \Phi(\mathcal{A})|_{\Sigma}$$

gegeben ist, so daß die Δ-Algebra $\Phi(\mathcal{A})$ erzeugt wird vom homomorphen Bild $\gamma_{\mathcal{A}}(\mathcal{A})$.

Für derartige parametrisierte Datentypen werden die Begriffe der initialen algebraischen Spezifizierbarkeit und der effektiven Darstellbarkeit eingeführt. Das Hauptergebnis der Arbeit ist der Nachweis der Äquivalenz der effektiven Darstellbarkeit und der initialen algebraischen Spezifizierbarkeit mit rekursiv aufzählbarer Menge definierender bedingter Gleichungen, falls $\mathrm{Dom}(\Phi)$ die Klasse aller semi-berechenbaren Σ-Algebren ist. Eine Σ-Algebra heißt dabei semi-berechenbar, wenn sie Faktoralgebra einer berechenbaren Σ-Algebra nach einer rekursiv aufzählbaren Kongruenzrelation ist.

Резюме

Рассматриваются параметризированные типы данных $\Phi: \mathrm{Alg}(\Sigma) \to \mathrm{Alg}(\Delta)$, причем Φ — частичный функтор от класса всех Σ-алгебр (алгебры параметров) к классу всех Δ-алгебр (целевые алгебры) для данных сигнатур Σ, Δ, где $\Sigma \subseteqq \Delta$. При этом редполагается, что целевая алгебра порождается с помощью гомоморфного образа алгебры параметров.

Для этих параметризированных типов данных доказывается общая теорема о существовании инициальных алгебраических спецификаций с условными уравнениями. Теорема включает в себя концепт эффективно данного параметризированного типа данных.

(Received: first version May 5, 1982,
revised version August 17, 1982)

Authors' addresses:

J. A. Bergstra
Department of Computer Science
University of Leiden
Wassenaarseweg 80
2300 RA Leiden
The Netherlands

J. W. Klop
Department of Computer Science
Mathematical Centre
Kruislaan 413
1098 SJ Amsterdam
The Netherlands