FINITE ALGEBRAIC SPECIFICATIONS OF SEMICOMPUTABLE DATA TYPES(*)

G. Marongiu

Dpt.of Mathematics, University of Bologna, Italy

S. Tulipani

Dpt.of Mathematics and Physics, University of Camerino,Italy

0-Introduction

For many years computer scientists have looked at data objects in terms of axioms which govern the behaviour of data structures. Such an attempt is called *Abstract Data Type Specification*. Abstraction is involved in the fact that only properties which are independent of data representation are considered.

In the algebraic approach, initiated in the paper of LISKOV-ZiLLES [1974], data structures are thought of as algebras in the sense of general algebra (see also [Zi 79] and specifications are given in terms of equations or conditional equations. But, only particular models of the specifying axioms play a special role. Data structures are usually finite, sometimes potentially infinite. Therefore, the significant models of a data type specification E in a signature Σ are given by the class $Alg_m(\Sigma,E)$ whose members are the models of E which are finitely generated by elements named as constants in Σ. The initial and final objects in $Alg_m(\Sigma,E)$, which are given up to isomorphism, determine the initial and final algebra semantics, respectively. More precisely, assuming that two closed terms t,s of signature Σ are given, then, the equation t=s is true when the terms t,s are evaluated in the initial algebra if and only if the formal equation t=s can be proved from E. Moreover, the equation t=s is consistent with E if and only if it is true in the final algebra. Both initial and final algebra semantical approaches have been widely discussed (see [ADJ 75,78,82], [Wa 79], the book [E-M 85].

BERGSTRA and TUCKER [1983] discussed the problem of characterizing semicomputable (cosemicomputable, computable) data types by means of finite conditional specification with hidden functions and no additional sorts plus initial algebra semantics

(*) Research performed under the auspices of the Italian CNR and MPI. This is a revised version of previous (unpublished) paper which circulated under the title: Finite Specification of Data Types with extra Operations.

(plus final algebra semantics), plus both semantics, respectively. However, the problem is solved only for cosemicomputable and computable data types.

We have tried to solve the problem for semicomputable data types giving a characterization which is weaker than that conjectured in the paper [B-T 83]. More precisely, let us assume A is an infinite semicomputable algebra of signature Σ and finitely generated by constants named in its signature. Then, we can determine a finite conditional equation specification (Σ',E) where Σ' is a finite signature extending Σ with no additional sorts so that: given two closed terms t,s of signature Σ, then

(1) t=s is true in A if and only if t=s is provable from E.

Moreover, our method provides an expansion A' to signature Σ' of the structure A such that A' is a model of E. However, A' in our theorem is not the initial algebra of Alg_m (Σ',E), as BERGSTRA and TUCKER conjectured in the semicomputable case. In fact we cannot prove for closed terms, t', s' of signature Σ':

(2) t'=s' is true in A' if and only if t'=s' is provable from E.

This means that the specification (Σ',E) is a consistent extension but not an enrichment of a specification for A (see [E-M 85, Chap. 6]). Furthermore our results also extend naturally to cosemicomputable and to computable data types (see Theorems B and C).

The agility of proof via Combinatory Logic does not consent comparison of the complexity of the set of conditional equations E to the complexity of the recursive functions which define A. This problem will be explored in future work [M-T 86] where it has also been proved that one single equation is sufficient for specifying a computable algebra as a hidden enrichment under both initial and final algebra semantics (according to BERGSTRA and TUCKER Definition, see [B-T 83] and [B-T 80].

To simplify notation, we treat the case of signature Σ in a single sort. However, the results can be easily extended to many-sorted algebras. It is assumed that the reader is familiar with the papers of Bergstra and Tucker [B-T 83], [B-T 86] and with the main papers on specification theory; in particular [E-M 85]. Moreover, the basic notations of universal algebra and recursion theory are assumed (see [Gr 78], [Mz 79]). In particular, it is assumed an understanding of the notions of semicomputable, cosemicomputable and computable algebra which can be found treated in full in MAL'CEV [1961] (see also [Ra 60]).

1-Preliminaries and Notations.

Let Σ be a finite algebraic signature having at least one constant symbol. We shall denote Σ by $\{f_1,\ldots,f_k, c_1,\ldots,c_r\}$ where $f_1,\ldots,f_k$ are function symbols and $c_1, \ldots, c_r$ constant symbols. If A is an algebra of carrier A and signature Σ we denote the interpretations of the symbols of Σ in the algebra A by $f_1^A,\ldots,f_k^A, c_1^A,\ldots,c_r^A$. As pointed out in the introduction, we shall consider only algebras of finite signature and in addition they are infinite and generated by elements named as constants in their signature. Hence, if an algebra is of signature Σ it is generated by the elements $c_1^A,\ldots,c_r^A$.

For such algebras the definitions of semicomputable, cosemicomputable and computable can be given more simply than in the general case along the following lines (see [Ma 61], [B-T 86]).

Definition 1.1 An algebra N is a recursive number algebra if and only if the carrier of N is the set N of natural numbers and the operations of N are recursive functions.

Definition 1.2 Let A be an algebra and N be a recursive number algebra of the same signature and $\pi : N \to A$ be a morphism. Then, π is

(i) a recursively enumerable (r.e) morphism,

(ii) a corecursively enumerable (co-r.e.) morphism,

(iii) a recursive morphism,

if and only if ker $\pi = \rho$ is

(i) a recursively enumerable relation,

(ii) a relation whose complement in N^2 is recursively enumerable,

(iii) a recursive relation,

respectively.

Definition 1.3 Let A be an algebra. Then, A is

(i) semicomputable,

(ii) cosemicomputable,

(iii) computable,

if and only if there exists a recursive number algebra N (of the same signature) and a surjective $\pi : N \to A$ such that π is

(i) a r.e epimorphism,

(ii) a co-r.e. epimorphism,

(iii) a recursive epimorphism, respectively.

For the following discussion we need some facts about combinatory algebras and combinatory logic that we recall in order to fix terminology and notation (for a comprehensive treatment see [Ba 81], [H-S 80]). Let cl = {•,K,S} be a signature such that • is a symbol of binary operation and K,S are constant symbols. A combinatory algebra is a structure $M = (M, \bullet, K^M, S^M)$ of signature cl which satisfies the axioms:

E1. $K \bullet x \bullet y = x$

E2. $S \bullet x \bullet y \bullet z = x \bullet z \bullet (y \bullet z)$,

where the association for the operation symbol • is done as usual from the left so that $M_1 \bullet M_2 \bullet \ldots \bullet M_k$ means $(\ldots((M_1 \bullet M_2) \bullet M_3) \bullet \ldots) \bullet M_k$. A combinator is a term in T(cl), i.e. a term of signature cl with no variables. Some combinators are called numerals. More specifically, if n is a natural number we denote by $\ulcorner n \urcorner$ a combinator which is called the numeral representing n and is defined inductively as follows:

$$\ulcorner 0 \urcorner = K \bullet I \qquad \ulcorner n+1 \urcorner = S \bullet B \bullet \ulcorner n \urcorner \tag{1.4}$$

where as usual I is the combinator $S \bullet K \bullet K$ and B is the combinator $S \bullet (K \bullet S) \bullet K$. We denote the equational theory, whose non logical axioms are E1 and E2 and which is named Combinatory Logic, by CL.

What we need is the following (see [Ba 81])

Fact 1.5. For every recursive function f there exists a combinator F_f which represents f in CL, viz. for all natural numbers $n_1, \ldots, n_m$, $CL \vdash F_f \bullet \ulcorner n_1 \urcorner \bullet \ldots \bullet \ulcorner n_m \urcorner = \ulcorner f(n_1, \ldots, n_m) \urcorner$.

2-Semicomputable Data Types. Main result.

Theorem A. Let A be a semicomputable algebra of signature Σ. Then, there exists a finite set E of conditional equations in a signature Σ' extneding Σ such that for all $t_1, t_2 \in T(\Sigma)$: $E \vdash t_1 = t_2$ if and only if $A \models t_1 = t_2$.

In order to prove Theorem A we must carry out some constructions and prove two Lemmas.

Let A be a semicomputable infinite algebra of signature $\Sigma = \{f_1, \ldots, f_k, c_1, \ldots, c_r\}$, N be a recursive number algebra of signature Σ and $\pi : N \to A$ be a r.e. epimorphism. Let us suppose that $m_1, \ldots, m_r$ are natural numbers such that

$$\pi(m_1) = c_1^A , \ldots, \pi(m_r) = c_r^A \tag{2.1}$$

and $m_i = m_j$ iff $c_i^A = c_j^A$ for i,j=1,...,r.

Let ρ be ker π and g be a recursive function of three variables such that for

$m,n \in N$, $(m,n) \in \rho$ iff there is a natural number p such that $g(m,n,p) = 0$. When f is a function symbol which is not constant of signature Σ, the combinator which represents in CL the function f^N which interpretes the symbol f in the recursive number algebra N is denoted by F_f. Finally, the combinator which represents the recursive function g which enumerates $\ker \pi$ is denoted by G.

Let now Σ' be the signature

$$\{f_1,\ldots,f_k,c_1,\ldots,c_r,\bullet ,K,S, \text{Nat}, \text{Hom}\}$$

i.e., Σ' is $\Sigma \cup$ cl plus two unary operation symbols Nat,Hom. Consider the following list of conditional equations in Σ', where $x,y,x_1,x_2,\ldots$ denote variables. To simplify notation we mention only one of the function symbols $f_1,\ldots,f_k$, say it f of m arguments

E1-E2 axioms of CL.

E3 is the conjunction for all f in Σ of

$$(\bigwedge_{1 \leq i \leq m} \text{Nat}(x_i)=K) \to \text{Hom}(F_f\bullet x_1\bullet \ldots\bullet x_m) = f(\text{Hom}(x_1),\ldots,\text{Hom}(x_m))$$

E4 $\bigwedge_{1 \leq j \leq r} \text{Hom}(\ulcorner m_j \urcorner) = c_j$

E5 $\text{Nat}(\ulcorner 0 \urcorner) = K \wedge (\text{Nat}(x) = K \longrightarrow \text{Nat}((S\bullet B)\bullet x) = K)$

E6 $(\text{Nat}(x) = K \wedge \text{Nat}(y) = K \wedge \text{Nat}(z) = K \wedge G\bullet x\bullet y\bullet z = \ulcorner 0 \urcorner) \longrightarrow \text{Hom}(x)=\text{Hom}(y)$.

The set of axioms E1-E6 is denoted by E. The idea we have exploided for writing down the set of axioms E can be roughly described as follows. The algebra A may be expanded to a structure A' which also becomes a combinatory algebra. Therefore, the structure N can be codified in the expansion A' of A. The unary operation symbol Nat is intended to be interpreted in the "characteristic function" of the subset of natural numbers of A'. In this way the structures N and A are glued up in the structure A' and the unary operation symbol Hom is intended to be interpreted in the codification of π in A'. All this is formalized in the following Lemma.

Lemma 2.2 Suppose A is the algebra of signature Σ previously described. Then, there exists an algebra A' which is an expansion of A to Σ' and is a model of E.

Proof. Since A is infinite, we may take a non trivial combinatory algebra $M = (M,\bullet,K^M,S^M)$ bijective to A. Using one bijection from A to M we can translate the operations of M in the operations $\bullet ,K^A,S^A$ on A so that the structure $(A,\bullet,K^A,S^A)$ is isomorphic to M.

Let now Nat^A, Hom^A be unary operations on A such that for $a \in A$

$$\mathrm{Nat}^A(a) = \begin{cases} K^A & \text{if there is } n \in N \text{ such that } a = \ulcorner n \urcorner^A \\ S^A & \text{otherwise} \end{cases}$$

$$\mathrm{Hom}^A(a) = \begin{cases} \pi(n) & \text{if there is } n \in N \text{ such that } a = \ulcorner n \urcorner^A \\ S^A & \text{otherwise} \end{cases}$$

Here $\ulcorner n \urcorner^A$ is the interpretation in the structure $(A,\bullet,K^A,S^A)$ of the numeral $\ulcorner n \urcorner$. This interpretation will be called the "codification" in the structure $(A,\bullet,K^A,S^A)$ of the natural number n.

The expansion A' is now defined as follows

$$A' = (A,\bullet,K^A,S^A,\ \mathrm{Nat}^A,\ \mathrm{Hom}^A) \quad .$$

We will now show that A' is a model of E. Axioms E1 and E2 are true in A' because the reduct of A' to cl is a combinatory algebra being isomorphic to M. To prove E3, assume $a_1,\ldots,a_m$ are elements of A which satisfy $\bigwedge_{1 \leq i \leq m} (\mathrm{Nat}(x_i)=K)$. By the definition of Nat^A there are natural numbers $n_1,\ldots,n_m$ such that $a_1=\ulcorner n_1 \urcorner^A,\ldots,a_m=\ulcorner n_m \urcorner^A$.

We must now show that $a_1,\ldots,a_m$ also satisfy the second member of the implication in E3. Now using Fact 1.5 we have

$$F_f^A \bullet \ulcorner n_1 \urcorner^A \bullet \ \ldots \bullet \ulcorner n_m \urcorner^A = \ulcorner f^N(n_1,\ldots,n_m) \urcorner^A \quad , \tag{2.3}$$

By the definition of Hom^A we have that

$$\mathrm{Hom}^A(\ulcorner f^N(n_1,\ldots,n_m) \urcorner^A) = \pi(f^N(n_1,\ldots,n_m)) \quad \text{and} \tag{2.4}$$

$$f^A(\mathrm{Hom}^A(a_1),\ldots,\ \mathrm{Hom}(a_m)) = f^A(\ \pi(n_1),\ldots,\ \pi(n_m)) \ .$$

Since π is a morphism

$$f^A(\mathrm{Hom}^A(a_1),\ldots,\ \mathrm{Hom}^A(a_m)) = \pi(f^N(n_1,\ldots,n_m)). \tag{2.5}$$

Therefore, from (2.4) and (2.5) we can conclude

$$\mathrm{Hom}^A(F_f^A \bullet a_1 \bullet \ \ldots \bullet a_m)) = f^A(\mathrm{Hom}^A(a_1),\ldots,\ \mathrm{Hom}^A(a_m)) \ .$$

That E4 is true in A' follows immediately from the definition of Hom^A and (2.1). It is

easy to prove that E5 is true in $\mathcal{A}'$ from the definition of numeral and of Nat^A.
Proof of E6. Assume that $a,b,c \in A$ satisfy the antecedent of E6, when x is substituted by a, y by b and z by c. This means that there exist natural numbers m,n,q such that

$$a = \ulcorner m \urcorner^A,\ b = \ulcorner n \urcorner^A,\ c = \ulcorner q \urcorner^A \quad \text{and} \quad G^A \bullet \ulcorner m \urcorner^A \bullet \ulcorner n \urcorner^A \bullet \ulcorner q \urcorner^A = \ulcorner 0 \urcorner^A \qquad (2.6)$$

But from Fact 1.5 we have

$$G^A \bullet \ulcorner m \urcorner^A \bullet \ulcorner n \urcorner^A \bullet \ulcorner q \urcorner^A = \ulcorner g(m,n,q) \urcorner^A \quad . \qquad (2.7)$$

Since the interpretation of numerals corresponding to distinct natural numbers in a non trivial combinatory algebra are distinct elements, from (2.6) and (2.7) we have that $g(m,n,q) = 0$. Hence, $\pi(m) = \pi(n)$. Therefore, by the definition of Hom^A,

$$Hom^A(\ulcorner m \urcorner^A) = Hom^A(\ulcorner n \urcorner^A)$$

Then, by (2.6) $Hom^A(a) = Hom^A(b)$ which means that E6 is true in $\mathcal{A}'$.

Given a term $t \in T(\Sigma)$ we now define a term $\hat{t}$ in T(cl) which describes the calculation that must be performed to obtain the natural number n such that $\pi(n) = t^A$. Recall that the structure $\mathcal{N}$ is codified in $(A,\bullet,K^A,S^A)$ via $\mathcal{M}$.

<u>Definition 2.8</u> For every term t in $T(\Sigma)$ define a term $\hat{t}$ in T(cl) by induction on the complexity of t as follows:

(i) if $t=c_i$, then $\hat{t}$ is $\ulcorner m_i \urcorner$ for $i=1,\dots,r$.

(ii) if $t=f(t_1,\dots,t_m)$, then $\hat{t}$ is $F_f \bullet \hat{t}_1 \bullet \dots \bullet \hat{t}_m$.

<u>Lemma 2.9</u> For every term t of $T(\Sigma)$:

(2.9i) there exists a natural number n such that $E \vdash \hat{t} = \ulcorner n \urcorner$,

(2.9ii) $E \vdash Hom(\hat{t}) = t$.

Proof of (2.9i). By induction on the complexity of t. If t is the constant symbol c_j then $\hat{t}$ is $\ulcorner m_j \urcorner$. Hence, the number m_j works. Let t be $f(t_1,\dots,t_m)$. By induction hypothesis we have natural numbers $n_1,\dots,n_m$ such that

$$E \vdash \hat{t}_1 = \ulcorner n_1 \urcorner ,\dots,\ E \vdash \hat{t}_m = \ulcorner n_m \urcorner \text{ and } \hat{t} \text{ is } F_f \bullet \hat{t}_1 \bullet \dots \bullet \hat{t}_m . \qquad (2.10)$$

But F_f represents f in CL. So we have a fortiori

$$E \vdash F_f \bullet \ulcorner n_1 \urcorner \bullet \dots \bullet \ulcorner n_m \urcorner = \ulcorner f^N(n_1,\dots,n_m) \urcorner \quad . \qquad (2.11)$$

Hence, when $n=f^N(n_1,\dots,n_m)$ we have (2.9i) from (2.10) and (2.11).
Proof of (2.9ii). By induction on the complexity of t. If t is the constant symbol c_j

then $\hat{t}$ is $\ulcorner m_j \urcorner$. Hence, from axiom E3 we get $E \vdash Hom(\hat{t})=t$.

Let now t be $f(t_1,\dots,t_m)$. By induction hypothesis we have

$$E \vdash Hom(\hat{t}_i)=t_i \ , \text{ for } i=1,\dots,m. \tag{2.12}$$

Therefore, using (2.9i) and axiom E5, we get

$$E \vdash Nat(\hat{t}_i)=K, \quad \text{for } i=1,\dots,m. \tag{2.13}$$

Then, from (2.13) and axiom E3 we have

$$E \vdash Hom(F_f \bullet \hat{t}_1 \bullet \dots \bullet \hat{t}_m)=f(Hom(\hat{t}_1),\dots,Hom(\hat{t}_m)). \tag{2.14}$$

Hence, from (2.12) and from the definition of $\hat{t}$ we get $Hom(\hat{t})=t$.

<u>Proof of Theorem A.</u> Given terms t_1,t_2 in $T(\Sigma)$ we have to prove that

$$E \vdash t_1=t_2 \quad \text{if and only if } A \models t_1=t_2 \ . \tag{2.15}$$

The "only if" direction follows immediately from Lemma 2.2. To prove the "if" direction assume

$$A \models t_1=t_2 \tag{2.16}$$

By Lemma 2.9 there exist natural numbers n_1,n_2 such that

$$E \vdash \hat{t}_i= \ulcorner n_i \urcorner \quad \text{and } E \vdash Hom(\hat{t}_i)=t_i \quad \text{for } i=1,2. \tag{2.17}$$

Then, from (2.16) and (2.17)

$$A' \models Hom(\ulcorner n_1 \urcorner)=Hom(\ulcorner n_2 \urcorner) \ . \tag{2.18}$$

Hence, by the definition of Hom^A we get

$$\pi(n_1)=\pi(n_2). \tag{2.19}$$

Therefore, there is a natural number q such that $g(n_1,n_2,q)=0$. Since G represents the function g we have

$$E \vdash G \bullet \ulcorner n_1 \urcorner \bullet \ulcorner n_2 \urcorner \bullet \ulcorner q \urcorner = \ulcorner 0 \urcorner \tag{2.20}$$

Now, using axiom E6 and (2.20) we get $E \vdash Hom(\ulcorner n_1 \urcorner)=Hom(\ulcorner n_2 \urcorner)$.
Hence, from (2.17) $E \vdash t_1 = t_2$.

3. <u>Cosemicomputable, Computable Data Types and remarks</u>

We can also apply our method to cosemicomputable and to computable data types.

Suppose that A is a cosemicomputable algebra of signature Σ, N is a recursive number algebra and $\pi : N \to A$ is a co-r.e. epimorphism. Let g' be a recursive function of three variables which enumerates the complement of $\rho=\ker\pi$. Therefore, for $m,n\in N$, we have

$\pi(m)\neq\pi(n)$ if and only if there is a natural number q such that $g'(m,n,q)=0$.

We denote the combinator which represents the recursive function g' by G'. Moreover, we let Σ' be the signature considered in the semicomputable case, i.e. $\Sigma'=\Sigma\cup cl\cup\{Hom,Nat\}$.

Now, consider the set E_F of conditional equations obtained from E by replacing E6 with the new axiom

$(E6)_F$ $(Nat(x)=K\wedge Nat(y)=K\wedge Nat(z)=K\wedge G'\bullet x\bullet y\bullet z = \ulcorner 0\urcorner\wedge Hom(x)\bar{=}Hom(y))\rightarrow K=S.$

(The subscript F is for "final semantics"). Then we have

Theorem B. Let A be a cosemicomputable algebra of signature Σ and let E_F be the finite set of conditional equations in the signature Σ' as previously described. Then, for all $t_1,t_2\in T(\Sigma)$ the following holds:

$$A\models\neg(t_1=t_2) \text{ if and only if } E_F\cup\{t_2=t_2\}\vdash K=S$$

The proof of Theorem B can be given in complete analogy with the proof of Theorem A.

Now consider a computable algebra A. Assume that N is as before and $\pi:N\rightarrow A$ such that $\ker\pi$ is recursive. Then, define the set E_C of conditional equations in the signature Σ' by $E_C=E\cup E_F$. We have

Theorem C. Let A be a computable algebra of signature Σ and let E_C be the finite set of conditional equations in the signature Σ' as previously defined. Then, for all $t_1,t_2\in T(\Sigma)$ the following holds:

$$A\models t_1=t_2 \text{ if and only if } E_C\vdash t_1=t_2 \text{ and}$$

$$A\not\models t_1=t_2 \text{ if and only if } E_C\cup\{t_1=t_2\}\vdash K=S.$$

The proof follows immediately from Theorem A and from Theorem B

We conclude with the following two remarks.

Remark 1. In the introduction we said that we would treat the case of signature Σ in a single sort in order to simplify notation. We now want to biefly explain how the method extends to many sorted algebras. Let Σ be a many sorted finite signature with a finite set of sorts S. A S-sorted algebra has a carrier A_s of sort s for every s in S. We say that a S-sorted algebra N is a recursive number algebra if every carrier of sort s is the set N of natural numbers and the operations of N are recursive functions (Cf. [B-T 86]). We say that A is a semicomputable S-sorted algebra of signature Σ if there exists a recursive S-sorted algebra N of signature Σ and r.e. epimorphism $\pi : N \to A$. When $s \in S$ let π_s be the s-component of π. If $c_j^{(s)}$ is a constant symbol of sort s let $m_j^{(s)}$ be a natural number such that

$$\pi_s(m_j^{(s)}) = (c_j^{(s)})^{A_s}$$

where the right hand side member is the interpretation of the symbol $c_j^{(s)}$ in the algebra A. Let g_s be a recursive function of three variables that enumerates $\ker\pi_s$ and G_s be a combinator which represents the function g_s. We say that a function symbol f is of sort $s_1 \ldots s_m \to s$ iff the interpretation of that symbol in A is a function of domain $A_{s_1} x \ldots x A_{s_m}$ and range A_s.

If the semicomputable algebra is infinite, then there is $s_o \in S$ such that A_{s_o} is infinite. Now take

$$\Sigma' = \Sigma \cup \{\bullet, K.S\} \cup \{Nat\} \cup \{Hom_s\}_{s \in S}$$

where

- $\bullet$ is a function symbol of sort $s_o s_o \to s_o$,
- K,S are constant symbols of sort s_o,
- Nat is a function symbol of sort $s_o \to s_o$,
- Hom_s is a function symbol of sort $s_o \to s$.

Now the proof goes on by replacing, in a obvious manner, axioms E3, E4 and E6 of E. Moreover, the expansion A' of A to the signature Σ' is constructed by translating a combinatory algebra in the carrier A_{s_o} and by interpreting Nat and Hom_s in order to codify the recursive S-sorted algebra N in A'. Then, Theorem A works also for many-sorted algebras. The same argument holds for Theorem B and Theorem C. Finally, we notice that if we allow s_o to be a new sort, we get a result which is analogous to Theorem 5.3 of [B-T 86] .

Remark 2. Let E be a set of conditional equations in a signature Σ' which extends the signature Σ . Consider the quotient structure $T(\Sigma')/{\equiv_E}$, where $\equiv_E$ is the usual congruence defined by: $t_1 \equiv_E t_2$ iff $E \vdash t_1 = t_2$. Then there exists an embedding

$$j: \langle T(\Sigma')/\equiv_E \rangle_\Sigma \hookrightarrow T(\Sigma')/\equiv_E \upharpoonright \Sigma$$

where the first algebra is the subalgebra of $T(\Sigma')/\equiv_E$ generated by the operations of Σ and the second algebra is the reduct of $T(\Sigma')/\equiv_E$, to the signature Σ . The specification (Σ',E) is said to be a *hidden enrichment* specification with respect to the initial algebra semantics of the algebra A if

$$A \cong \langle T(\Sigma')/\equiv_E \rangle_\Sigma = T(\Sigma')/\equiv_E \upharpoonright \Sigma$$

where the isomorphism into A is induced by the natural evaluation of the terms of $T(\Sigma)$ in A (see [E-M 85] , [B-T 86]).

Our Theorem A proves that there is such an isomorphism and that the embedding j is a retraction, i.e. there exists a morphism h such that $h \circ j$ = identity. In this case we could say that the specification (Σ',E) is a *consistent extension for* A. The term consistent extension is used in the literature (for example in [E-M 85] Chap. 6) in quite similar situations but involves two specifications.

Then, according to Definition 3.4 of [B-T 86] we can restate Theorem A as follows.

Proposition. The specification method (for abstract data types) by means of a finite conditional equation consistent extension with no additional sorts, is complete for the class of semicomputable data types.

References

[ADJ 75] J.A. GOGUEN, J.W. THATCHER, E.G. WAGNER and J.B. WRIGHT, *Abstract data types as initial algebras and correctness of data representations*, in Proceedings ACM Conference on Computer Graphics, Pattern Recognition and Data Structure, NY 1975, 89-93.

[ADJ 78] J.A. GOGUEN, J.W. THATCHER and E.G. WAGNER, *An initial algebra approach to the specification, correctness and implementation of abstract data types*, Current trends in Programming Methodology, Vol. 4: Data Structuring (R.T. Yeh, ed.), Prentice-Hall (1978), 80-149.

[ADJ 82] J.W. THATCHER, E.G. WAGNER and J.B. WRIGHT, *Data type specification: parametrization and the power of specification technique*, TOPLAS 4(4), 1982, 711-732.

[Ba 81] H.P. BARENDREGT, *The lambda calculus, its syntax and semantics*, North-Holland, Amsterdam 1981.

[B-T 80] J.A. BERGSTRA and J.V. TUCKER, *On bounds for the specification of finite data types by means of equations and conditional equations*, preprint IW 131/80 Amsterdam 1980.

[B-T 83] J.A. BERGSTRA and J.V. TUCKER, *Initial and final algebra semantics for data type specifications: two characterization theorems*", SIAM J. of Computing 12 (1983), 366-387.

[B-T 86] J.A. BERGSTRA and J.V. TUCKER, *Algebraic specifications of Computable and Semicomputable Data Types*, Department Computer Science Research Report CS-R8619, Amsterdam (May 1986).

[E-M 85] H. EHRIG and B. MAHR, *Fundamentals of Algebraic Specification 1*, Monograph EATCS, Springer-Verlag 1985.

[Gr 78] G. GRATZER, *Universal Algebra*, 2nd ed. Springer-Verlag, 1979.

[H-S 80] J.R. HINDLEY and J.P. SELDIN (Eds.), *To H.B. Curry: Essays on combinatory logic, lambda-calculus and formalism*, Academic Press, NY 1980.

[L-Z 74] B.H. LISKOV and S.M. ZILLES, *Programming with Abstract Data Types*, Proc. ACM Symp. on very high level languages, SIGPLAN Notices 9, (1974), 50-59.

[L-Z 75] B. LISKOV and S. ZILLES, *Specification techniques for data abstractions*, IEEE Transactions on Software Engineering 1 (1975), 7-19.

[Ma 71] A.I. MAL'CEV, *Constructive algebras I*, The Metamathemathics of Algebraic Systems, Collected papers: 1936-1967, North-Holland.

[M-T 86] G. MARONGIU and S. TULIPANI, *Remarks on complexity in specifying computable types*, in preparation.

[Mz 79] J. MALITZ, *Introduction to Mathematical Logic*, Springer- Verlag, 1979.

[Ra 60] M.O. RABIN, *Colputable algebra, general theory and the theory of computable fields*, Trans. Amer. Math. Soc. 95 (1960), 341-360.

[Wa 79] M. WAND, *Final algebra semantics and data type extensions*, J. Computer and Ststems Sciences 19 (1979), 27-44.

[Zi 79] S. ZILLES, *An introduction to data algebras* Lect. Notes in Comp. Sci. 86, (1979), 248-270.