stichting
mathematisch
centrum

MC

AFDELING INFORMATICA IW 131/80 MAART
(DEPARTMENT OF COMPUTER SCIENCE)

J.A. BERGSTRA & J.V. TUCKER

ON BOUNDS FOR THE SPECIFICATION OF FINITE DATA TYPES BY MEANS OF EQUATIONS AND CONDITIONAL EQUATIONS

Preprint

2e boerhaavestraat 49 amsterdam

Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O).

1980 Mathematics subject classification 03D45 03D80 68B15

ACM-CR category 4.34

On bounds for the specification of finite data types by means of equations and conditional equations.*

by

J.A. Bergstra** & J.V. Tucker

ABSTRACT

Within the framework of the ADJ Group's initial algebra semantics for data types, we prove that while any finite data type can be specified by means of finitely many equations the number of equations required is sometimes necessarily a function of the size of the data type. By using hidden operators, however, the number of equations needed to specify a finite data type A can be proved to be bounded by numbers depending only on the signature Σ of A. For example, if A is any finite single-sorted data type, generated by p initial values and having q operations, then A can be specified using 1 hidden constant, 6 hidden functions and $15 + p + q$ equations. We also prove that such a data type A can be specified using 1 hidden function, 1 equation and 2 conditional equations.

KEY WORDS & PHRASES: *algebraic data types; finite data types; equational and conditional specifications with, and without, hidden operators; boundedness properties*

---

* This paper is not for review as it is meant for publication elsewhere.

** Department of Computer Science, University of Leiden, Wassenaarseveg 80, Postbus 9512, 2300 RA Leiden, The Netherlands.

INTRODUCTION

In this paper, we shall prove four theorems about the comparative complexities of equational and conditional equation specifications of *finite* data types. To do this we shall work strictly within the mathematical framework of the initial algebra semantics for data types created by the ADJ Group.

Our starting point is the observation that *each finite data type* A *of signature* Σ *possesses a finite equational specification* (Σ,E). At first sight, this seems a comforting fact: no hidden operators are required (which is not the case for rather straightforward looking infinite data type semantics, see ADJ[7]). The equations of E turn out to be elementary identifications between terms over Σ. The observation is easy to prove.

However, the number $|E|$ of equations in E is a function of the size $|A|$ of the data type A. Actually, $|E|$ is $O(\lambda|A|^{\mu})$ where $\lambda,\mu\in\omega$, the set of natural numbers, and these constants one can read off the signature Σ. The method which provides the specification E amounts to a syntactic tabulation of every operation of A on all data: we call this technique *graph enumeration*. Disappointed, one wonders whether or not theory can illuminate more subtle and interesting relationships between finite semantics and specifications. Surely there are more concise, if more sophisticated, ways of defining finite data types? In Section 3 we shall show there are no such general methods to be found which use only equations:

> *There is a family of finite data types* $\{A_n : n\in\omega\}$ *of common single-sorted signature* Σ *such that to specify* A_n *by graph enumeration requires* $O(|A_n|)$ *simple identifications over* Σ *and each* A_n *cannot be specified with less than* $O(|A_n|)$ *equations over* Σ.

The proof of this theorem explicitly shows that *the work of defining an* A_n *must fall on identifications rather than equations* and so that graph enumeration is sometimes optimal among equational techniques.

The idea that conditional equations are more powerful than equations and can lead to concise specifications can now be neatly illustrated by proving that *each* A_n *can be specified using* 8 *identifications over* Σ *and* 1 *conditional equation.*

In section 4 we consider how hidden operators can be used to make concise specifications for finite data types. Let A be any finite data type whose signature Σ names n sorts, p constants and q operations. We prove that

> A *possesses a specification involving at most* n *hidden constants,* 2n+4 *hidden operators and* 15+p+q+3(n-1) *equations*

and that

> A *possesses a specification involving at most* n *hidden operators,* n *identifications over* Σ *and* 2n *conditional equations.*

Section 1 documents notation, some algebraic definitions, and describes the graph enumeration method. Section 2 is something of a digression: there we give an example of a family of data type specifications whose syntactic size grows linearly but defines semantics which outgrows the Ackermann Function. In Section 5 we discuss the next stage in the analysis of bounded specifications for finite data types.

This paper is the fourth in our series of mathematical studies of the power of definition and adequacy of algebraic specifications for data types [1,2,3], see also [4]. In particular, it acts as a companion to our [3] where we proved essentially the same boundedness result for equational hidden functions specifications mentioned above, but for *infinite* computable data type semantics.

In what follows, it is assumed the reader is familiar with both the *raison d'être* of the ADJ Group's algebraic theory of data types as well as with its technical machinery, at least to the level of the basic paper ADJ[5] but not extending beyond the material of ADJ[6,7]. At one place we refer to a longish technical argument of [3], rather than rewrite it here; otherwise knowledge of our previous work, although desirable, is not essential.

1. DATA TYPES: SEMANTICS AND SPECIFICATIONS

After recording, or referencing, some notation and technical ideas, we prove our basic observation that finite equational specifications are adequate to define all finite data types. For most of what follows the original sources are ADJ[5,6,7].

Typically, a *many-sorted algebra* A of signature Σ_A is composed of n (non-empty) *component domains* $A_1,\ldots,A_n$ named in Σ_A by *sorts* $1,\ldots,n$. Each operation of A is of the form

$$\sigma^{\lambda,\mu} = \sigma^{(\lambda_1,\ldots,\lambda_k),\mu} : A_{\lambda_1} \times \ldots \times A_{\lambda_k} \to A_\mu$$

where $\lambda = (\lambda_1,\ldots,\lambda_k)$ and $\lambda_i,\mu \in \{1,\ldots,n\}$ for $k \in \omega$. For such an operation $\sigma^{\lambda,\mu}$ we call k the *arity* of $\sigma^{\lambda,\mu}$ and speak of $\sigma^{\lambda,\mu}$ as a *k-ary operation* of A.

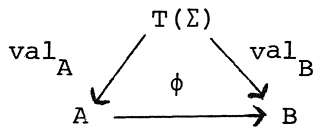
In the algebraic theory of data types, the semantical structure of a data type is modelled by a many-sorted algebra A *which is assumed to be finitely generated by elements named as constants in its signature.* Such algebras are *minimal* in the sense that they contain no proper subalgebras. The following facts are obvious:

1.1. LEMMA. *Let* A *and* B *be algebras of common signature* Σ *both finitely generated by elements named as constants in* Σ. *Then* (1) *any* Σ*-homomorphism* $\phi: A \to B$ *is surjective*; (2) *if* $\phi,\psi: A \to B$ *are* Σ*-homomorphisms then* $\phi=\psi$; *and* (3) *if there are* Σ*-homomorphisms* $\phi: A\to B$ *and* $\psi: B \to A$ *then* $A \cong B$ (*by either* ϕ *or* ψ).

Let $\equiv$ be an equivalence relation on the many-sorted algebra A. A *traversal* for $\equiv$ is a family of sets $J_\lambda \subset A_\lambda$ indexed by the sorts $1 \le \lambda \le n$ of Σ_A, such that for each $b \in A_\lambda$ there is one, and only one, $a^\lambda \in J_\lambda$ for which $b \equiv a^\lambda$.

Let Σ be a signature and let $T(\Sigma)$ denote the Σ-algebra of all terms over Σ and $T_\Sigma[X_1,\ldots,X_n]$ denote the algebra of polynomials in the *many-sorted* list of indeterminates $X_1,\ldots,X_n$. If A is any Σ-algebra then by *term evaluation in* A we mean a map $val_A: T(\Sigma) \to A$ which evaluates each term $t \in T(\Sigma)$ on substituting the constants of A for their names in t and the operations of A for their names in t; val_A is uniquely definable as an epimorphism $T(\Sigma) \to A$.

<u>1.2. LEMMA</u>. *Let* $\phi: A \to B$ *be a homomorphism between* Σ*-algebras. Then the following diagram commutes*



Lemma 1.2 follows immediately from Lemma 1.1

By *polynomial evaluation* in A we mean the process of substituting some $a=(a_1,\ldots,a_n) \in A^n$ for indeterminates $X = (X_1,\ldots,X_n)$, where a_i is an element of the same sort as X_i, into polynomial $t(X) \in T_\Sigma[X_1,\ldots,X_n]$, along with the constants and operations of A for their names in t(X), and evaluating t(a) in A.

A *simple equation* or *simple identification* over Σ is a pair (t,t') of terms from $T(\Sigma)$ invariably written t = t' while an *equation* is a pair (t(X), t'(X)) of polynomials from some $T_\Sigma[X_1,\ldots,X_n]$ invariably written t(X) = t'(X), although this does *not* mean that t(X) and t'(X) must contain any indeterminate in common. *Conditional equations* are formulae of the form

$$t_1(X) = t_1'(X) \wedge \ldots \wedge t_k(X) = t_k'(X) \to t(X) = t'(X).$$

The *length* of any of these types of equation e we write $\|e\|$ and by this we mean the length of the equation thought of as a string over signature Σ and the alphabet

$$(\quad) \quad , \quad = \quad \wedge \quad \to \quad 0 \quad 1$$

where {0,1} is used to represent indeterminates by means of the binary representations of their natural number indices.

If E is a set of formulae over Σ and A is a Σ-algebra satisfying the laws of E we say A is an E-*algebra* and occasionally write $A \models E$. The class of all E-algebras we denote $ALG(\Sigma,E)$. We assume the reader is familiar with the construction of the initial algebra for $ALG(\Sigma,E)$ from $T(\Sigma)$ and we write this $T(\Sigma,E) = T(\Sigma)/\equiv_E$ where $\equiv_E$ denotes the smallest congruence on $T(\Sigma)$ containing those identifications between terms determined by the laws of E. See ADJ[5,6] for details.

A many-sorted algebra A has a *finite equational specification* (Σ,E) if $\Sigma_A = \Sigma$, E is a finite set of equations over Σ, and $T(\Sigma,E) \cong A$. The definition of a *simple equational specification* and of a *conditional equational specification* follows *mutato nomine*.

We now formally define the nature of our hidden function specifications (see [1]).

Let A be a many-sorted algebra of signature Σ_A. Let Σ be a signature $\Sigma \subset \Sigma_A$ and having the same sorts as Σ_A. Then we mean by

$A|_\Sigma$ the Σ-algebra whose domains are those of A and whose operations and constants are those of A named in Σ: the Σ-*reduct* of A; and by

$\langle A\rangle_\Sigma$ the Σ-subalgebra of A generated by the operations and constants of A named in Σ *viz* the smallest Σ-subalgebra of $A|_\Sigma$.

A many-sorted algebra A has a *finite, equational hidden enrichment specification* (Σ,E) if $\Sigma_A \subset \Sigma$, and Σ contains exactly the sorts of Σ_A, and E is a finite set of equations over Σ such that

$$T(\Sigma,E)|_{\Sigma_A} = \langle T(\Sigma,E)\rangle_{\Sigma_A} \cong A.$$

Again one also defines hidden enrichment specifications involving simple equations and conditional equations in the obvious way.

The following fact was noted, casually, in our [1].

<u>1.3. BASIC OBSERVATION</u>. *Let* A *be a finite many-sorted algebra finitely generated by elements named as constants in its signature* Σ. *Then* A *has a specification* (Σ,S) *involving a finite number of equations which are*

simple identifications between terms over Σ.

More specifically: let A *have component domains* $A_1,\ldots,A_n$ *and* q *operations. Let* $M = \max\{|A_i| : 1 \le i \le n\}$ *and let* m *be the maximum arity of the operations of* A. *Then the set of equations* S *can be chosen with* $|S| \le q.M^m$.

PROOF. Let A be finitely generated by $a_1,\ldots,a_\ell$ lying in various domains. Let A_i consist of m_i elements, say, $b^i_1,\ldots,b^i_{m_i}$. For each sort i, choose m_i polynomials $t^i_j(X)$ in an appropriately sorted list of indeterminates X such that

$$t^i_j(a_1,\ldots,a_\ell) = b^i_j.$$

Now for each operation $\sigma^{\lambda,\mu}:A_{\lambda_1}\times\ldots\times A_{\lambda_k} \to A_\mu$ write out the graph of $\sigma^{\lambda,\mu}$

$$\text{graph}(\sigma^{\lambda,\mu}) = \{(b^{\lambda_1}_{j_1},\ldots,b^{\lambda_k}_{j_k},b^\mu_j) : \sigma^{\lambda,\mu}(b^{\lambda_1}_{j_1},\ldots,b^{\lambda_k}_{j_k}) = b^\mu_j\}$$

in terms of the polynomials

$$\sigma^{\lambda,\mu}(t^{\lambda_1}_{j_1}(a),\ldots,t^{\lambda_k}_{j_k}(a)) = t^\mu_j(a)$$

where $a = (a_1,\ldots,a_n)$. Collect (the syntactic versions of) these identities to make the set S of simple identifications over Σ. It is routine to check $T(\Sigma,S) \cong A$ and to verify the bound on $|S|$ claimed. Q.E.D.

The method used in this last proof we refer to as the *graph enumeration technique*.

2. APÉRITIF

In this section we wish to make the point that a family of small equational specifications can give rise to a family of very large data type semantics.

Let Σ be any signature. By an *effective family of equational specifications over* Σ we mean a family $E = \{E_n : n\in\omega\}$ of sets of equations over Σ such that the relation $e \in E_n$ is decidable uniformly in n.

2.1. THEOREM. *Let Σ consist of a constant symbol* 0, *a unary function* S *and a binary function symbol* ACK. *Then there exists an effective family* $E = \{E_n : n \in \omega\}$ *of finite equational specifications over Σ such that, for each* $n \in \omega$, (i) $|E_n| = 4$ (ii) *for each* $e \in E_n$ $\|e\|$ *is* $O(n)$ *and* (iii) $|T(\Sigma,E_n)| > a(n) = ack(n,n)$ *where* $ack(n,m)$ *is Ackermann's Function.*

PROOF. Consider the sequence of finite numerical algebras

$$A_n = (\{0,\ldots,a(n)\};\ 0,\ s_n,\ ack_n)$$

where $s_n(x) = \min(x+1,a(n))$ and $ack_n(x,y) = \min(ack(x,y),a(n))$ for $n \in \omega$.

The required family $E = \{E_n : n \in \omega\}$ is meant to specify the family $A = \{A_n : n \in \omega\}$. As E_* we simply take the 3 equations over Σ

$$ACK(0,X) = S(X)$$
$$ACK(S(X),0) = ACK(X,S(0))$$
$$ACK(S(X),S(Y)) = ACK(X,ACK(S(X),Y)).$$

And, on defining equation e_n to be

$$S(ACK(S^n(0),S^n(0))) = S^2(ACK(S^n(0),S^n(0)))$$

we set $E_n = E_* \cup \{e_n\}$ for each $n \in \omega$. Obviously, our family E is effective and satisfies conditions (i) and (ii). It remains to prove $T(\Sigma,E_n) \cong A_n$.

2.2. LEMMA. $\{S^i(0) : 0 \le i \le a(n)\}$ *is a traversal for* $\equiv_{E_n}$.

PROOF. It is easy to see that $S^i(0) \equiv_{E_n} S^j(0)$ if, and only if, $i = j$. So we have only to consider the completeness property for the sets: that for each $t \in T(\Sigma)$, $t \equiv_{E_n} S^z(0)$. This is done by induction on the complexity of terms in $T(\Sigma)$. The basis is obvious so assume $t = \underline{\lambda}(s_1,\ldots,s_k)$ and $s_i \equiv_{E_n} S^z(0)$. There are two cases: either $\underline{\lambda} = S$ or $\underline{\lambda} = ACK$. The first of these cases is trivial and the second follows from the identity

$$ACK(S^i(0),S^j(0)) \equiv_{E_n} S^{ack_n(i,j)}(0) \qquad (*)$$

which we leave to the reader to verify (using the equations of E_n and induction on i). Q.E.D.

From Lemma 2.2 we know that the map $\phi_n : A_n \to T(\Sigma, E_n)$, defined by

$$\phi_n(i) = [S^i(0)]$$

is a bijection. That ϕ_n is a homomorphism is an easy calculation:

$$\begin{aligned} ACK(\phi_n(i),\phi_n(j)) &= ACK([S^i(0)],[S^j(0)]) \\ &= [ACK(S^i(0),S^j(0))] \\ &= [S^{ack_n(i,j)}(0)] \qquad \text{by identity } (*); \\ &= \phi_n(ack_n(i,j)) \end{aligned}$$

Q.E.D.

3. A FAMILY OF FINITE DATA TYPE SEMANTICS WHOSE EQUATIONAL SPECIFICATIONS CANNOT BE BOUNDED

Let $[0,n]$ denote the interval $\{0,\ldots n\} \subset \omega$ and let $B = \{T,F\}$. A *finite arithmetic of order* n *with (embedded) booleans* is a single-sorted algebra FAB(n) defined on $[0,n] \cup B$ by taking 0,n,T,F as constants and by defining two unary functions S,P on $[0,n]$ by

$$\begin{array}{lll} S(x) = x+1 \quad (x < n) & & P(x) = x-1 \quad (x > 0) \\ S(n) = n & & P(0) = 0 \end{array}$$

and trivially extending these successor and predecessor functions to B by defining

$$\begin{array}{lll} S(T) = F & & P(T) = T \\ S(F) = F & & P(F) = T \end{array}$$

FAB(n) can be conveniently visualised by means of Figure 3.1. Let Σ be the signature of such algebras.

Define $A_n = FAB(2(n+1))$. Now $|A_n| = 2(n+1)+1+2 = (2n+5)$ and, according to Basic Observation 1.3, each algebra A_n can be specified using

2. $|A_n| = 2(2n+5)$ simple identifications over Σ. We will now prove that the general technique of graph enumeration is, in all essential respects, optimal for the family $\{A_n : n \in \omega\}$.

3.1. THEOREM. *Each algebra* A_n *fails to possess a finite equational specification with less than* n *equations.*

PROOF. We begin by constructing a new family of algebras K_n of signature Σ by adding n special points $a_1,\ldots,a_n$ to A_n which disturb the definition of the predecessor function P on the odd numbers in A_n. K_n is best defined, pictorially, by Figure 3.2.

Formally, S is unchanged on $[0,2(n+1)] \cup B$ and $S(a_i) = 2i+1$ for $1 \le i \le n$. And P is redefined only on $\{2i+1 : i \in [1,n]\}$ where $P(2i+1) = a_i$ for $1 \le i \le n$; while for the new points $P(a_i) = 2i-1$, $1 \le i \le n$.

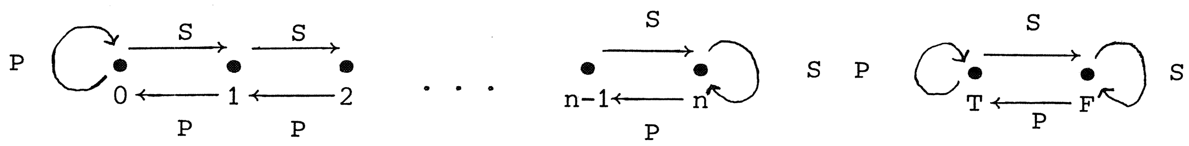


Figure 3.1

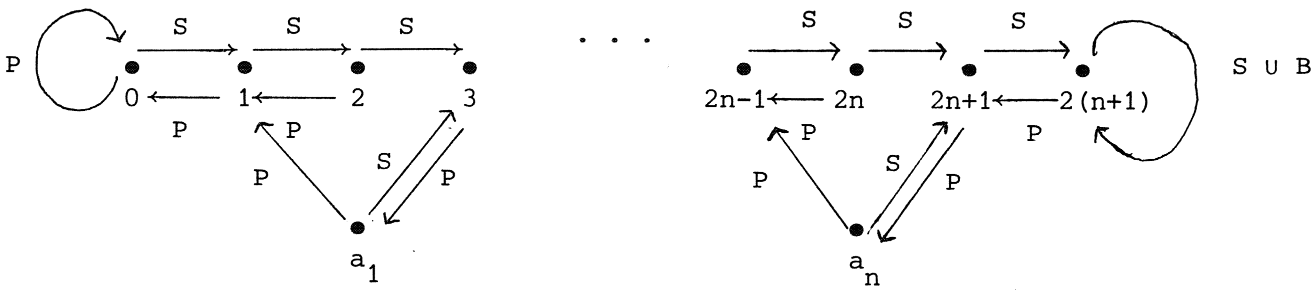


Figure 3.2

"Pushing in the triangles (of Figure 3.2)" defines a map $\Phi_n : K_n \to A_n$. Precisely, take $\Phi_n : [0,2(n+1)] \cup B \to [0,2(n+1)] \cup B$ to be the identity map and, elsewhere in K_n, take $\Phi_n(a_i) = 2i$ for $1 \le i \le n$. Φ_n is an epimorphism.

Assume n is fixed so that we may simplify our notation by writing A for A_n, K for K_n and Φ for Φ_n. We subsequently distinguish between the boolean parts of K and A by writing B_K and B_A respectively. Let $E = \{e_1, \ldots, e_m\}$ be a finite set of equations specifying A. We shall prove $m \ge n$. To do this we make use of the semantics intermediate between K and A:

Consider the following family of congruences on K. Let $\lambda = (\lambda_1, \ldots, \lambda_k)$ where $1 \le k \le n$ and $1 \le \lambda_i \le n$. Define $\equiv_\lambda$ to be the smallest congruence on K containing the identifications $\{a_{\lambda_i} = 2_{\lambda_i} : 1 \le i \le k\}$ and set $B_\lambda = K/\equiv_\lambda$ and let $\phi_\lambda : K \to B_\lambda$ be the canonical factor map determined by $\equiv_\lambda$. Clearly, the equivalence classes of $\equiv_\lambda$ are the sets

$$\{0\}, \{1\}, \ldots, \{a_{\lambda_1}, 2_{\lambda_1}\}, \ldots, \{a_{\lambda_k}, 2_{\lambda_k}\}, \ldots, \{2n+1\}, \{2(n+1)\}, \{T\}, \{F\}.$$

The map ϕ_λ simply pushes in those triangles of Figure 3.2 which are indexed by $\lambda_1, \ldots, \lambda_k$. Thus, if λ and μ are numerical sequences containing the same entries then $\phi_\lambda = \phi_\mu$ and $B_\lambda = B_\mu$. If λ contains each element of $[1,n]$ then $\phi_\lambda = \Phi$ and $B_\lambda = A$.

A *splitting* of K and A is a minimal algebra C together with homomorphisms α, β such that

$$K \xrightarrow{\alpha} C \xrightarrow{\beta} A$$

Of course, the minimality of C implies α, β are epimorphisms. We can now establish that we have a complete parameterisation of the semantics between K and A.

<u>3.2.LEMMA</u>. Let $K \xrightarrow{\alpha} C \xrightarrow{\beta} A$ *be a splitting. Then* $C \cong B_\lambda$ for some λ.

We leave the proof of Lemma 3.2 as an easy exercise for the reader.

Let I be the set of all B_λ with that $B_{\lambda_0} \cong A$ replaced by A and with K adjoined. I contains precisely K,A and *one* isomorphic copy of each algebra belonging to a proper splitting. Notice that $|I| = 2^n$.

3.3. LEMMA. A *is the only algebra in* I *which satisfies all the equations of* E.

PROOF. Let B ∈ I and assume B is an E-algebra. By the initiality of $T(\Sigma,E) \cong A$ for E-algebras there exists a homomorphism $\phi:A \to B$. On the other hand, since B ∈ I there is a splitting $K \xrightarrow{\alpha} B \xrightarrow{\beta} A$. Thus there are epimorphisms $A \to B$ and $B \to A$ and so, by Lemma 1.1, $A \cong B$ and, by the definition of I, A = B. Q.E.D.

Technically, the key to the proof of the theorem is this lemma.

3.4. LEMMA. *For each* e ∈ E, *either* e *is satisfied in* K *or there exists* $1 \le i(e) \le n$ *such that* $B_{i(e)} \in I$ *satisfies* e.

Given Lemma 3.4 the rest of the proof is as follows. Let $e_1,\ldots,e_k$ be those equations of E which are not satisfied in K. Then calculate $i(e_1),\ldots,i(e_k)$ and set $\lambda = (i(e_1),\ldots,i(e_k))$. Clearly, all of $e_1,\ldots,e_k$ are true in B_λ. Since B_λ is an epimorphic image of K, all the other equations of E are true in B_λ. Thus B_λ is an E-algebra. Now if $|\lambda| = k \le m < n$ then $B_\lambda \ne A$; a fact which contradicts Lemma 3.3. Therefore, $m \ge n$.

PROOF OF LEMMA 3.4.

First let us consider the simple identifications in E.

3.5. LEMMA. *Let* $t,t' \in T(\Sigma)$ *and suppose* $val_A(t) = val_A(t')$ *but* $val_K(t) \ne val_K(t')$. *Then there is* $1 \le i \le n$ *such that*

$$\textit{either } val_K(t) = 2i \textit{ and } val_K(t') = a_i$$
$$\textit{or } val_K(t) = a_i \textit{ and } val_K(t') = 2i$$

PROOF. Since $val_A(t) = val_A(t')$ and by Lemma 1.2, $val_A(-) = \Phi val_K(t) = \Phi val_K(t')$. We consider the values of Φ when $val_K(t)$ is a boolean, a number, and an additional point.

Case (i): $val_K(t) \in B_K$. Now $B_K = \Phi^{-1}(B_A)$ and the bijectiveness of $\Phi:B_K \to B_A$ implies $val_K(t) = val_K(t')$: a contradiction. So this case cannot arise.

Case (ii): $val_K(t) = \ell \in [0,2(n+1)]$. Here $\Phi val_K(t') = \Phi(\ell) = \ell$. If $\ell \neq 2i$ for $1 \leq i \leq n$ then, by the definition of Φ, $val_K(t') = \ell$ which contradicts $val_K(t) \neq val_K(t')$. Therefore $\ell = 2i$ for $1 \leq i \leq n$ and $val_K(t) \neq val_K(t')$ entails $val_K(t') = a_i$.

Case (iii): $val_K(t) = a_i$ for $1 \leq i \leq n$. Here $\Phi val_K(t') = \Phi(a_i) = 2i$ and since $\Phi^{-1}(2i) = \{2i,a_i\}$ we must have $val_K(t') = 2i$ if $val_K(t) \neq val_K(t')$. Q.E.D.

If $e \equiv t = t' \in E$ is a simple identification which is *not* satisfied in K then, by Lemma 3.5, we can choose $i(e)$ from $val_K(t)$, $val_K(t')$ such that $B_{i(e)}$ makes the right identification in K so as to satisfy e. Thus we know Lemma 3.4 to be true of the simple identifications in E.

Consider the equations in E. In principle, these are of essentially 3 kinds:

(1) $t(X) = t'(Y)$ (2) $t(X) = t'(X)$ (3) $t(X) = t'$

where X,Y are single indeterminates and $t' \in T(\Sigma)$. The upshot of our analysis will be that *an equation is true in* A *if, and only if, it is true in* K. First we shall show that E may contain equations of type (2) only.

Case (3). Let $e \equiv t(X) = t'$ and suppose $val_A(t') \in B_A$. Choosing $X = 0$ we observe that $val_A(t(0)) \notin B_A$. Conversely, if $val_A(t') \notin B_A$ then choosing $X = T$ we see that $val_A(t(T)) \in B_A$. Thus e cannot be true in A.

Case (1). Let $e = t(X) = t'(Y)$. Such an e cannot be true in A because setting $Y = T$ yields an equation of type (3).

So let $e \equiv t(X) = t'(X)$ and suppose this is not true in K. Then there is $z \in K$ such that $t(z) \neq t'(z)$ and we may choose some $r \in T(\Sigma)$ such that $val_K(r) = z$. We now consider the simple identification $t(r) = t'(r)$ which is true in A but not in K. Using Lemma 3.5 we can calculate i such that (say)

$$val_K(t(r)) = e_i \text{ and } val_K(t'(r)) = 2i.$$

We can now embark on a case distinction argument based upon the leading function symbols of t and t'. In each case we obtain a contradiction to the hypothesis that e is not true in K.

Let $t(X) = P(\tau(X))$ and $t'(X) = P(\tau'(X))$. Substituting $X = r$, as chosen above, we know $val_A(P\tau(r)) = val_A(P\tau'(r))$ but $val_K(P\tau(r)) \neq val_K(P\tau'(r))$. Clearly, $val_K(\tau(r)) = val_K(\tau'(r))$ - suppose not and there is an immediate contradiction. This means that $val_K(\tau(r))$, $val_K(\tau'(r))$ are a pair $i,j \in K$ such that $P(i) = P(j)$ but $i \neq j$. On inspecting the algebra K (Figure 3.2) we see that this implies $i = 0$, $j = 1$ or $i = 1$, $j = 0$. In both cases $val_K(P\tau(r)) = 0 = val_K(P\tau'(r))$ which is the required contradiction.

The other case distinctions, such as $t(X) = P(\tau(X))$, $t'(X) = S(\tau'(X))$ and so on, we leave to the reader.

This completes the proofs of Lemma 3.4 and Theorem 3.1.

We are now in a position to formally illustrate that conditional equation specifications can *concisely* specify a data type which cannot be *concisely* specified by equations alone; see ADJ [7].

3.6. THEOREM. *Each finite arithmetic with booleans* FAB(n) (and so, in particular, each A_n) *possesses a specification involving* 8 *simple identifications and* 1 *conditional equation.*

PROOF. Let the signature of the algebras be $\Sigma = \{0,b,T,F,S,P\}$ where b names the largest number in each FAB(n). Define E_n to be the set of equations

$$\begin{aligned} S(T) &= F & \qquad P(T) &= T \\ S(F) &= F & \qquad P(F) &= T \\ S^n(0) &= b & \qquad P(0) &= 0 \\ S(b) &= b & \qquad P^n(b) &= 0 \end{aligned}$$

$$S(X) = P(X) \rightarrow PS(X) = X$$

We invite the reader to verify $T(\Sigma,E_n) \cong FAB(n)$. Q.E.D.

3.7. PROBLEM. *Does there exist a family of finite algebras* $\{A_n : n \in \omega\}$ *of common single-sorted signature* Σ *such that to specify* A_n *by graph* *enumeration requires* $O(\lambda|A_n|^{\mu})$ *simple identifications over* Σ *and each* A_n *cannot be specified with less than* $O(\lambda|A_n|^{\mu})$ *conditional equations over* Σ?

4. BOUNDS FOR THE SPECIFICATION OF FINITE DATA TYPE SEMANTICS USING HIDDEN OPERATORS

Let us begin with a resumé of the rôle of hidden operators in the theory of data type specifications. To specify certain *infinite* computable data type semantics by means of equations, or conditional equations, it is known that the use of hidden operators is necessary; see our [1], or, better, ADJ [7]. On the other hand, it is also known [1] that equations and hidden functions are sufficient to define any computable data type semantics, the finite data types requiring only simple identifications, of course. In this section we use hidden operators to reduce the number of equations, or conditional equations, needed to specify a finite data type. The theorem we prove for equations and hidden functions is meant to be contrasted with Theorem 3.1 as well as to complement our earlier study of bounds for equational specifications of infinite computable data types [3]. Since the theorem about bounds for conditional specifications is simpler than the theorem for equational specifications we shall prove it first.

4.1. THEOREM. *Let* A *be a finite many-sorted algebra containing* n *sorts and assumed finitely generated by elements named as constants in its signature* Σ. *Then* A *possesses a finite conditional hidden enrichment specification involving* n *hidden functions,* n *identifications and* 2n *conditional equations. In particular, any such single-sorted algebra may be specified by means of* 1 *hidden operator,* 1 *identification and* 2 *conditional equations.*

PROOF. We shall prove this theorem for the case of single-sorted finite algebras since the technical ideas involved have obvious modifications which cover the many-sorted case.

Given single-sorted finite algebra A, choose some named constant

$a \in A$ and any function $h:A^3 \to A$ such that

$$h(x,y,z) = a \text{ if, and only if, } x = y \text{ and } z = a.$$

Add this function h to A to make a new algebra A_0 of signature Σ_0. Obviously, $A_0|_{\bar{\Sigma}} = \langle A_0\rangle_\Sigma = A$. We shall give an appropriately bounded specification for A_0.

First, using Basic Observation 1.3, choose any finite simple equational specification (Σ_0,S_0) for A_0. Let $S_0 = \{t_i = s_i : 1 \le i \le m\}$. Now define E_0 to consist of these 3 formulae over Σ_0, where H names h of A_0 in Σ_0:

$$H(X,Y,Z) = \underline{a} \to X = Y \quad (1)$$

$$H(X,Y,Z) = \underline{a} \to Z = \underline{a} \quad (2)$$

$$H(t_n,s_n,H(t_{n-1},s_{n-1},H(\ldots \quad \ldots H(t_1,s_1,\underline{a})\ldots) = \underline{a} \quad (3)$$

We claim $T(\Sigma_0,E_0) \cong A_0$.

Clearly, A_0 is an E_0-algebra and so, by initiality, there is an epimorphism $\phi:T(\Sigma_0,E_0) \to A_0$. Thanks to Lemma 1.1, it is enough to show the existence of an epimorphism $\psi:A_0 \to T(\Sigma_0,E_0)$.

Define inductively the Σ_0 terms $\tau_1 = H(t_1,s_1,\underline{a})$

$$\tau_{k+1} = H(t_{k+1},s_{k+1},\tau_k)$$

Thus equation (3) is merely $\tau_n = \underline{a}$, and observe that this implies $t_n = s_n$ and $\tau_{n-1} = \underline{a}$ by equation (1) and (2). A trivial induction now shows that equation (3) implies $t_i = s_i$ for $1 \le i \le m$. Therefore, $T(\Sigma_0,E_0)$ is an S_0-algebra and, by initiality, there is an epimorphism $\psi:T(\Sigma_0,S_0) \to T(\Sigma_0,E_0)$. Since $A_0 \cong T(\Sigma_0,S_0)$ we are done.

The strategy for the many-sorted case is simply this. One knows each domain A_i of A is non-empty and so one chooses either a named constant of sort $1 \le i \le n$ or some other element from A_i which, by minimality, is a polynomial function of named constants of various sorts. With these elements an $h_i:A_i^3 \to A_i$ can be made for each domain and added to A to form A_0. Thus A_0 has n more functions than A. To define E_0, choose a simple specification (Σ_0,S_0) for A_0 and copy down equations (1)-(3) simply adding a sort index i to (1) and (2), and in the case of equation (3), selecting only those terms obtained from S_0 which are of the correct sort i under consideration. This way E_0 contains 2n conditional equations and n equations. The formal arguments are just as in the single-sorted case. Q.E.D.

4.2. THEOREM. *Let* A *be a finite many-sorted algebra containing* n *sorts and assumed finitely generated by elements named as constants in its signature* Σ. *Then* A *possesses a finite equational hidden enrichment specification involving* n *hidden constants,* 2n+4 *hidden functions and* 15+p+q+3(n-1) *equations. In particular, any such single-sorted algebra may be specified by means of* 1 *hidden constant,* 6 *hidden functions and* 15+p+q *equations.*

PROOF. We follow our usual method of explaining the proof for the single-sorted case in detail before discussing the proof for the many-sorted case.

Let $|A|$ = n+1 and choose a bijection $\alpha:[0,n] \to A$ so as to construct a finite numerical algebra R on [0,n] by inducing operations on [0,n] from those of A. Thus, if f is a k-ary operation of A let f_α (temporarily) denote that unique map which commutes the diagram

$$\begin{array}{ccc} A^k & \xrightarrow{f} & A \\ {\scriptstyle \alpha^k}\uparrow & & \uparrow{\scriptstyle \alpha} \\ [0,n]^k & \xrightarrow{f_\alpha} & [0,n] \end{array}$$

where $\alpha^k(x_1,\ldots,x_k) = (\alpha x_1,\ldots,\alpha x_k)$. Obviously, $\alpha:R \to A$ is an isomorphism and we can identify A with R and concentrate on providing an appropriate specification for R. We build a new algebra R_0 of signature Σ_0 by adding to R $0 \in R$, as a constant, and 6 new functions:

$$S_n(x) = \begin{cases} x+1 & \text{if } x<n \\ n & \text{if } x=n \end{cases} \qquad P_n(x) = \begin{cases} 0 & \text{if } x=0 \\ x-1 & \text{if } x>0 \end{cases}$$

$$\min(x,y) = \begin{cases} x & \text{if } x\le y \\ y & \text{if } x>y \end{cases} \qquad \mathrm{sum}_n(x,y) = \min(x+y,n)$$

$$\text{null}(x,y,z) = \begin{cases} y \text{ if } x=0 \\ z \text{ if } x\neq 0 \end{cases} \qquad h(x,y,z) = \begin{cases} 0 \text{ if } x=y \;\&\; z=0 \\ 1 \text{ otherwise} \end{cases}$$

Clearly, $R_0|_\Sigma = \langle R_0\rangle_\Sigma = R$ and so it is enough to prove that R_0 has a finite equational specification (Σ_0, E_0) with $|E_0| = 15+p+q$.

Let the above additions to R be named in Σ_0 by

0, S, P, MIN, SUM, NULL, H

Here is a prescription for the finite set of equations E_0 over Σ_0.

First, for each constant $\underline{c} \in \Sigma$ naming numerical constant $c \in R$, set

$$\underline{c} = S^c(0) \tag{0}$$

Next come equations to define the 6 functions above.

Successor $\quad S^{n+1}(0) = S^n(0) \quad (1)$

Minimum
$$\begin{aligned} MIN(X,Y) &= MIN(Y,X) \\ MIN(0,Y) &= 0 \\ MIN(S(X),S(Y)) &= S(MIN(X,Y)) \end{aligned}$$

Predecessor $\quad SP(MIN(X,S^{n-2}(0))) = MIN(X,P^{n-2}(0)) \quad (3)$
$$\begin{aligned} P(0) &= 0 \\ PS^n(0) &= S^{n-1}(0) \end{aligned}$$

Addition $\quad SUM(X,0) = X \quad (4)$
$$SUM(X,S(Y)) = S(SUM(X,Y))$$

Equality with zero $\quad NULL(0,Y,Z) = Y \quad (5)$
$$NULL(S(X),Y,Z) = Z$$

Equality $\quad H(X,Y,Z) = H(Y,X,Z) \quad (6)$
$$\begin{aligned} H(X,X,0) &= 0 \\ H(X,Y,S(Z)) &= S(0) \\ H(P(X),S(SUM(X,Y)),0) &= S(0) \end{aligned}$$

It remains for us to give the equations which are to define the operations of R. This is fairly involved as we will assign to each operation of R just one equation over Σ_0 into which the graph of that operation has been coded.

Let f be a k-ary operation of R named by $\underline{f} \in \Sigma$.

First we choose some linear enumeration of $[0,n]^k$ say $a_1, a_2, \ldots, a_i, \ldots a_d$ where $1 \le i \le d = (n+1)^k$. Let $a_i(j)$ denote the j-th coordinate of $a_i \in [0,n]^k$ for $1 \le j \le k$.

Secondly, we define a family $\{e_i(X_1,\ldots,X_k): 1 \le i \le d\}$ of polynomials over Σ_0 in indeterminates $X_1,\ldots,X_k$. Each e_i is defined inductively over the list of indeterminates: for $1 \le i \le d$ we define polynomials $e_i^0,\ldots,e_i^k$ by

$$e_i^0 = 0$$
$$e_i^j(X_1,\ldots,X_j) = H(X_j, S^{a_i(j)}(0), e_i^{j-1}(X_1,\ldots,X_{j-1})) \quad 1 \le j \le k$$

so that $e_i(X_1,\ldots,X_k) = e_i^k(X_1,\ldots,X_k)$. The point is that we want polynomials with this property: let $z_1,\ldots,z_k \in [0,n]$

$$e_i(z_1,\ldots,z_k)=0 \text{ in } R_0 \text{ if, and only if, } z_1=a_i(1)\&\ldots\&s_k=a_i(k)$$

a fact which is realised in our specification later on (Lemma 4.6).

Using these e_i we may now inductively build the equation for $\underline{f}$. Define more polynomials u^i for $0 \le i \le d$.

$$u^0(X_1,\ldots,X_k) = 0$$
$$u^i(X_1,\ldots,X_k) = \mathrm{NULL}(e_i(X_1,\ldots,X_k),\ S^{f(a_i)}(0),\ u^{i-1}(X_1,\ldots,X_k)).$$

These polynomials have the property that, for each $1 \le i \le d$ and any $z_1,\ldots,z_k \in [0,n]$,

$$u^i(z_1,\ldots,z_k) = \begin{cases} f(z_1,\ldots,z_k) & \text{if } (z_1,\ldots,z_k)=a_j \in [0,n]^k \ \&\ j\le i \\ 0 & \text{otherwise} \end{cases}$$

This fact is realised by our specification in Lemma 4.5. Whence our equation for $\underline{f}$ is

$$f(X_1,\ldots,X_k) = u^d(X_1,\ldots,X_k) \tag{7}$$

and, having completed the description of E_0, we may observe that $|E_0| = 15+p+q$.

It now must be shown that $T(\Sigma_0,E_0) \cong R_0$. Let $\equiv$ abbreviate $\equiv_{E_0}$ on $T(\Sigma_0)$

4.3. LEMMA. *$\{S^i(0): i \in [0,n]\}$ is a traversal for $\equiv$.*

PROOF. We leave to the reader the task of checking that

$$S^i(0) \equiv S^j(0) \text{ if, and only if, } i = j$$

and prove that each $t \in T(\Sigma_0)$ is equivalent to some numeral $S^z(0)$. This is done by induction on term complexity. The basis is obvious and the induction step follows from this next proposition.

4.4. LEMMA. *Let $t = \lambda(s_1,\ldots,s_k)$ where $\underline{\lambda} \in \Sigma_0$ names operation λ of R_0. If $s_i \equiv S^{z_i}(0)$ for $1 \le i \le k$ and $z_i \in [0,n]$ then $t \equiv S^{\lambda(z_1,\ldots,z_k)}(0)$.*

PROOF. This is proved by considering the different cases for $\underline{\lambda}$ in the order

$$S,\ MIN,\ P,\ SUM,\ NULL,\ H,\ \underline{f}_1,\ldots,\underline{f}_q$$

All cases are routine except that of $\underline{\lambda} = \underline{f}$ naming a k-ary operation f of R and this we will now explain. (Notice the ordering is based on the equations of E_0: MIN is used to define P and so in proving Lemma 4.4 in case $\underline{\lambda} = P$ one needs to know the lemma is true in case $\underline{\lambda} = MIN$.)

It must be shown that $\underline{f}(S^{z_1}(0),\ldots,S^{z_k}(0)) \equiv S^{f(z_1,\ldots,z_k)}(0)$. By equation (7),

$$\underline{f}(S^{z_1}(0),\ldots,S^{z_k}(0)) \equiv u^d(S^{z_1}(0),\ldots,S^{z_k}(0)).$$

The result now follows from this next lemma.

4.5. LEMMA. *For each* $1 \le j \le d$,

$$u^j(S^{z_1}(0),\ldots,S^{z_k}(0)) \equiv \begin{cases} S^{f(z_1,\ldots,z_k)}(0) & \text{if } (z_1,\ldots,z_k) = a_i \in [0,n]^k \ \& \ i \le j \\ 0 & \text{otherwise} \end{cases}$$

PROOF. This is proved by induction on j. The basis is trivial so assume the lemma true of u^{j-1} and consider u^j. Let $z = (z_1,\ldots,z_k) = a_i \in [0,n]^k$ and let $S^z(0)$ abbreviate $(S^{z_1}(0),\ldots,S^{z_k}(0))$. We know

$$u^j(S^z(0)) \equiv \mathrm{NULL}(e_j(S^z(0)),\ S^{f(z)}(0),\ u^{j-1}(S^z(0)))$$

and to apply equation (5) for NULL we first need to calculate $e_j(S^z(0))$:

4.6. LEMMA. *For any* $1 \le i \le d$ *and any* $z_1,\ldots,z_k \in [0,n]$

$$e_i(S^{z_1}(0),\ldots,S^{z_k}(0)) \equiv 0 \textit{ if, and only if, }$$
$$S^{z_1}(0) \equiv S^{a_i(1)}(0) \& \ldots \& S^{z_k}(0) \equiv S^{a_i(k)}(0).$$

We shall not prove this fact as it is an easy induction on the complexity of e_i. Lemma 4.5 presents us with 3 cases:
Case (1): $i > j$. Here Lemma 4.6 and equation (5) yields $u^j(S^z(0)) \equiv u^{j-1}(S^z(0))$ which, using the induction hypothesis and the assumptions that $z = a_i$, $i > j > j-1$, allows us to conclude $u^j(S^z(0)) \equiv 0$.
Case (2): $i = j$. This is precisely the case when $e_j(S^z(0)) \equiv 0$. By equation (5) for NULL, $u^j(S^z(0)) \equiv S^{f(z)}(0)$ and we are done.
Case (3): $i < j$. Again, by Lemma 4.6, $u^j(S^z(0)) \equiv u^{j-1}(S^z(0))$ which by the induction hypothesis has the correct value.

This completes the proof of Lemmas 4.2, 4.3, and 4.4.

Now Lemma 4.3 allows us to define $\phi: R_0 \to T(\Sigma_0, E_0)$ by $\phi(i) = [S^i(0)]$ and to conclude it is a bijection. It is easy to check ϕ is a homomorphism by a calculation based upon Lemma 4.4.

Having proved Theorem 4.2 in the single-sorted case we shall turn to the many-sorted case. The pattern established in our [2] and [3] is to be repeated: we are to choose the largest component data domain of the algebra and simulate the whole algebra on it.

First, we make a many-sorted numerical copy R of A and to each of its n domains we add 0 as a constant and a finite successor function tailored to the size of its domain just as above. Let R_1 be one of the largest domains of R. We simulate R over R_1 by adding to R_1 the other 5 functions

min, P, sum, null, h

along with projection functions ${}^{i}copy$: $R_1 \to R_i$ for $i \neq 1$. These are all the hidden functions we use in making the new algebra R_0. To give a finite equational specification (Σ_0, E_0) for R_0 we now take an equation of type (1) above for each successor function, the 5 operator equations (2)-(6) and the pair of equations for each ${}^{i}copy$ we gave in [3]. What now remains are the normalising equations for the p constants, of the type (0) above, and the q equations for the operators defined

$$\underline{f}^{\lambda,\mu}({}^{\lambda_1}COPY(x_1), \ldots, {}^{\lambda_k}COPY(x_k)) = {}^{\mu}COPY(u^d(x_1, \ldots, x_k)$$

Since a formal description of the argument is nothing more than an obvious merge of the simulation argument in [3] and the single-sorted case argument given here we omit it. Q.E.D.

The bounds obtained in the *infinite* case treated in [3] were, incidentally, n hidden constants, 3n+3 hidden functions, and $17+p+q+4(n-1)+n_F$ equations where n_F is the number of finite sorts in the infinite many-sorted algebra. (This n_F as a bound is essentially a signature invariant because $n_F \leq n-1$). Obviously, the bounds of [3] cover *all* computable data types.

5. CONCLUDING REMARKS

The bounds on the numbers of hidden functions and equations, or conditional equations, are obtained by coding up data type semantics into a few *long* equations. The extent to which this is inevitable is a subject of some theoretical interest and is probably best organised around the derivation of trade-off formulae:

<u>5.1. THEOREM</u>. *Let* Σ *be a single-sorted finite signature. If every* Σ*-algebra of cardinality* n *can be specified by a hidden enrichment specification over some signature* $\Sigma_0 \supset \Sigma$ *using at most* e *conditional equations each of which is of length at most* ℓ*, where* $e > 0$ *and* $\ell \geq 3$*, then*

$$\ell e \log(\ell e) > \frac{1}{3} \log (N(\Sigma,n))$$

wherein $\log(x) = \log_2(x)$ *and* $N(\Sigma,n)$ *is the number of distinct (up to isomorphism)* Σ*-algebras of size* n.

<u>PROOF</u>. Define Σ_u to be a signature containing ℓe k-ary function symbols for each $0 \leq k < \ell$ so that $|\Sigma_u| = \ell^2 e$. One can check that if A is any Σ-algebra possessing a hidden enrichment specification (Σ_0, E_0) in which E_0 satisfies the (e,ℓ) condition then Σ_0 can be chosen as a subset of Σ_u. So we consider all sets of conditional equations E_0 over Σ_u satisfying $|E_0| \leq e$ and with the length of formulae bounded by ℓ.

Recalling, from Section 1, that, for the purposes of measuring formula length, we consider each conditional equation as a string over Σ_u and the list

$$(\quad) \quad , \quad = \quad \wedge \quad \rightarrow \quad 0 \quad 1$$

we see that the number of sets E_0 is at most $(8+\ell^2 e)^{\ell e}$. Thus, the hypothesis of the theorem implies

$$(8+\ell^2 e)^{\ell e} \geq N(\Sigma,n).$$

Now for $e > 0$, $\ell \geq 3$ it is the case that $\ell^3 e^3 > (8+\ell^2 e)$. The rest of the proof is a routing calculation:

$$(\ell^3 e^3)^{\ell e} = 2^{\log(\ell e)^{3\ell e}} = 2^{3\ell e.\log(\ell e)} > N(\Sigma,n)$$

$$\ell e \log(\ell e) > \frac{1}{3}\log(N(\Sigma,n))$$

Q.E.D.

The restrictions $e > 0$ and $\ell \geq 3$ in Theorem 5.1 are there to rule out degenerate cases in the sense that $e = 0$ gives rise to infinite algebras and the smallest equations, such as those which identify constants in a signature, require at least 3 symbols.

The invention of more general and more exact formulae is a problem well worth pursuing, along with investigations of the structure of such expressions as $N(\Sigma,n)$ for commonly used signatures Σ. Clearly, if Σ is single-sorted and names p constants and q operations, and m bounds the arity of these operations, then on any set A of size n one can define not more than

$$n^p + n^{n^{qm}}$$

Σ-structures on A. Since isomorphism of Σ-structures is the criterion of semantical identity, one knows that

$$N(\Sigma,n) < n^p + n^{n^{qm}}$$

More refined statements of these kinds of facts is the objective of an entirely new project, of course.

REFERENCES

[1] 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.

[2] BERGSTRA, J.A. & J.V. TUCKER, *A characterisation of computable data types by means of a finite, equational specification method,* Mathematical Centre, Department of Computer Science Research Report IW 124, Amsterdam, 1979. (To be presented at ICALP '80.)

[3] ———, *Equational specifications for computable data types: six hidden functions suffice and other sufficiency bounds,* Mathematical Centre, Department of Computer Science Research Report IW 128, Amsterdam, 1980.

[4] ———, *On the adequacy of finite equational methods for data type specification,* ACM-SIGPLAN Notices 14 (11) (1979) 13-18.

[5] GOGUEN, J.A., J.W. THATCHER & E.G. WAGNER, *An initial algebra approach to the specification, correctness and implementation of abstract data types,* in R.T. YEH (ed.) *Current trends in programming methodology* IV, *Data structuring,* Prentice-Hall, Engelwood Cliffs, New Jersey, 1978, 80-149.

[6] THATCHER, J.W., E.G. WAGNER, & J.B. WRIGHT, *Specification of abstract data types using conditional axioms,* IBM-T.J. Watson Research Center Report RC-6214, Yorktown Heights, 1976.

[7] ———, *Data type specification: parameterization and the power of specification techniques,* IBM-T.J. Watson Research Center Report RC 7757, Yorktown Heights, 1979. (Revision of paper with same title in *Proceedings, SIGACT 10th Annual Symposium on Theory of Computing*, ACM 1978, 119-132.)