In: Bulletin of the EATCS, no 13 (1981)

An equational class of I/O-computable datastructures that contains no recursive datastructure.

J.A. Bergstra* and J.-J.Ch. Meyer**

Datastructures are many sorted minimal algebras having sorts of two kinds:

i) at least one I/O-sort; the I/O-sorts contain the visible (observable) part of the data structure.

ii) perhaps some internal sorts; these contain the invisible (protected) part of the data structure.

(If τ_1,τ_2 are terms of an internal sort then the predicate $\tau_1 = \tau_2$ is not given. For I/O-sorts, however, equality is assumed to be a primitive predicate that a user of the datatype can use).

We will consider in detail two situations where there is one I/O-sort and one internal sort. This will show two interesting phenomena that arise in the algebraic theory of data structures.

CONVENTION. If A is a data structure then we will always assume that the word problem on its I/O-sorts is decidable (for each I/O-sort).

To motivate this convention one assumes that equality between closed terms of I/O-sorts is the only kind of information that a user can expect from the data structure, and that we are interested only in data structures that possess a 'computable' implementation.

Let A be a data structure of signature Σ. A has a (finite) equational specification if there are (finitely many) equations E over Σ such that $T(\Sigma,E) \cong A$.

* Department of computer Science, University of Leiden.

** Department of Computer Science, Free University, Amsterdam.

$T(\Sigma,E)$ is the initial E-algebra of signature Σ.

In [1] we proved the following.

PROPOSITION I. Let A be a data structure (in our convention), put:

$G_{I/O}(A)$ = the set of all true equations between closed terms of I/O sorts.

Then i) $G_{I/O}(A)$ is decidable (obvious from our convention),

ii) $T(\Sigma , G_{I/O}(A))$ is computable in the sense of [2].

Thus: if one admits infinitely many equations it is always possible given a data structure A to find a specification (Σ,E) using equations for I/O-sorts only that specifies an algebra $B = T(\Sigma,E)$ which is I/O-equivalent to A. (This means that B has the same signature as A and has the same word problem on the I/O-sorts. In other words A and B have the same observable behaviour.)

A question is: can one be forced for the purpose of finitary specification to use equations that run over internal sorts as well. The answer to this question is: YES. An easy example is as follows: A embodies a single counter:

I/O-sort : $B = \{T,F\}$.

internal sort : $N = \{n \mid n \in \omega\} = \omega$.

constants : $T,F \in B$ and $0 \in N$.

operations : $S : N \to N$ defined by $S(x) = x+1$

$P : N \to N$ defined by $P(x) = x-1$

$NULL : N \to B$ defined by $NULL(x) = \begin{cases} T & \text{if } x = 0 \\ F & \text{otherwise} \end{cases}$

A finite equational specification of A is for instance given by the equations:

$NULL\ (0) = T$

$NULL\ (S(x)) = F$ (x internal)

$P(0) = 0$

$PS(x) = x$ (x internal)

However, suppose that (Σ_A,E) specifies an algebra B that is I/O-equivalent to A and that E contains equations of I/O-sort only. All equations are of the forms

1) NULL (τ(x)) = T or

2) NULL (τ(x)) = F or

3) NULL (τ(0)) = T or

4) NULL (τ(0)) = F or

(equations between brackets can be reduced to equations of the above forms.)

(5) NULL (τ(x)) = NULL (ρ(y)) or

6) NULL (τ(x)) = NULL (ρ(x) or

(7) NULL (τ(x)) = NULL (ρ(0)) or

(8) NULL (τ(0)) = NULL (ρ(0))

with $\tau,\rho \in \{S,P\}^*$. Let K be the maximum length of the τ's and ρ's occurring in these equations. Consider a proof of the (true) fact NULL($P^{k+1}\ S^{k+1}$ (0)) = T. It is possible to read this proof in such a way that one sees $P^{k+1}\ S^{k+1}$ (0) being rewritten using 'rules' 6) to a form in which equations of forms 1) or 3) can be applied. As eq. 1) can never be true 3) must be used. But then it is obvious that using equations of the form 6) a sequence

NULL($P^{k+1}\ S^{k+1}$ (0)) = NULL (τ_1(0)) = ... = NULL (τ_k(0)) = T is found where at some first point i, P S^{k+1} (0) is not a subterm of τ_i(0) (because it cannot be a subterm of τ_k(0)).

Consider this last step:

NULL (τ_{i-1} (0)) = NULL (τP S^{k+1}(0)) = NULL (τ_i(0)).

In more detail:

NULL (τ P S^{k+1} (0)) = NULL ($\tau^1\ \tau^2$(0)) = NULL ($\rho^1\ \tau^2$(0))

where NULL (τ^1(x)) = NULL (ρ^1(x)) is an axiom of the form 6). The length of τ is at least K. The length of τ^1 is at most K thus τ^2(0) contains P S^{k+1}(0) as a subterm, contradicting the minimality of i.

The lesson of this result is clearly that equations over internal sorts cannot be avoided. But this introduces a problem at once. If (Σ,E) specifies B, I/O-equivalent to A then B may not be computable (i.e. the word problem

on internal sorts is not decidable); this in contrast with proposition I which ensures that B is computable if one has I/O-equations exclusively. Even worse, it is conceivable that the data abstraction $DA(G_{I/O}(A) \cup E)$ that contains all (Σ,E) algebras I/O-equivalent with A has no computable element at all. That this is indeed possible is the content of our second result.

Let F and G be two binary recursive functions on ω such that the sets $V_F = \{n|\exists m\ F(n,m) = 0\}$ and $V_G = \{n|\exists m\ G(n.m) = 1\}$ are recursively inseparable r.e. sets. Assume that F and G have values 0 and 1 only, and that for all $n,m,m' : m \geq m' \Rightarrow F(n,m) \leq F(n,m')$ and $G(n,m) \geq G(n,m')$.

We construct an algebra A with sorts N and S. A is computable. The I/O-sort N has the following shape:

$$N = (\omega, S, F, G, H_1, \ldots, H_G, \cdot, 0) .$$

The internal sort S has no functions on its own.

There are three functions connecting both sorts:

MAP : $N \to S$; MAP is a bijection.

EVAL : $S \times N \to N$ is defined by the equation

e_1 : $EVAL(MAP(x), y) = F(x,y) \cdot G(x,y)$

D : $N \times S \to S$, defined by equations e_2 and e_3

e_2 : $D(S(x)\ Y) = MAP(S^K(0)$ with $K \notin A \cup B$ fixed

e_3 : $D(0,Y) = Y$.

The functions $H_1, \ldots, H_6$ are hidden functions on A that, according to [2] can be chosen in such a way that N has a finite equational specification (Σ_N, E).

It is easy to verify that A is the initial algebra $T(\Sigma, E \cup \{e_1,e_2,e_3\})$.

Let us construct $\bar{A}$ from A by identifying $MAP(S^i(0))$ and $MAP(S^j(0))$ as soon as for each ℓ $EVAL(MAP(S^i(0), S^\ell(0)) = EVAL(MAP(S^i(0)), S^\ell(0))$.

In $\bar{A}$ (which is of course I/O-equivalent with A) the following equation e holds:

$$e : D(F(x,y), MAP(x)) = MAP(S^K(0))$$

We claim that $E_0 = E \cup \{e_1, e_2, e_3\} \cup \{e\}$ has no computable model which is I/O-equivalent with A.

This implies that the collection of all E_0 algebras that are I/O-equivalent with A is a data abstraction without computable element. Thus in this case the nice implication of PROPOSITION I is lost.

To show the claim suppose that $\equiv$ is a decidable congruence on A which turns it into an E_0-algebra. Observe:

$i \in A \Rightarrow$ for some n $F(i,n) = 0$ and thus:

$$MAP(X^i(0)) =$$
$$D(0, MAP(S^i(0))) = \quad (e_2)$$
$$D(F(S^i(0), S^n(0)), MAP(S^i(0))) =$$
$$MAP(S^k(0)) \quad (e)$$

$i \in B \Rightarrow$ for some n $G(i,n) = 1$, moreover $i \notin A$ and consequently for all m $F(i,m) = 1$. Thus

$$EVAL(MAP(S^i(0)), S^n(0)) =$$
$$F(S^i(0), S^n(0)) \cdot G(S^i(0), S^n(0)) =$$
$$S(0) \cdot S(0) = S(0)$$

but $EVAL(MAP(S^K(0), S^n(0)) =$

$$F(S^K(0), S^n(0) \cdot G(S^{K(0)}, S^n(0)) =$$
$$S(0) \cdot 0 = 0.$$

It follows that $i \in A \Rightarrow MAP(S^i(0)) \equiv MAP(S^K(0))$

and $i \in B \Rightarrow MAP(S^i(0)) \not\equiv MAP(S^K(0))$

But this implies that A and B have a recursive separation, contradicting the assumptions.

REFERENCES

[1] J.A. Bergstra and J.-J.Ch. Meyer, I/O-computable datastructures. Submitted for publication in the SIGPLAN notices.

[2] J.A. Bergstra and J.V. Tucker, Equational specifications for computable data types: six hidden functions suffice and other sufficiency bounds, Mathematical Centre, Department of Computer Science Research Report IW 131, Amsterdam 1980.

A 'HARD-CORE' THEOREM FOR RANDOMIZED ALGORITHMS

by

Shimon Even* and Yacov Yacobi**

* Computer Science Dept., Technion-IIT, Haifa, Israel

** Electrical Engineering Dept., Technion-IIT, Haifa, Israel.

ABSTRACT

Lynch proved that if a decision problem A is not solvable in polynomial time then there exists an infinite recursive subset X of its domain on which the decision is almost everywhere complex.

We prove a similar result for a common class of randomized algorithms.