Annales Societatis Mathematicae Polonae
Series IV: Fundamenta Informaticae IV.3 (1981)

IMPLICIT DEFINABILITY OF ALGEBRAIC STRUCTURES BY MEANS OF PROGRAM PROPERTIES*

Jan Bergstra

Rijksuniversiteit Leiden

Jerzy Tiuryn

Warsaw University

Received February 13, 1980

AMS Categories: 68B10, 68B15

Abstract: The following problem is investigated in the paper: what structures can be uniquely defined by algorithmic properties? The algorithmic properties are represented in this paper as open formulae of the logic of effective definitions. This approach can be viewed as an alternative way of defining abstract data types.

Key words: Logic of programs, computation in abstract structures, algorithmic properties, abstract data types.

---

*This paper has been written while the author was on leave at Lehrstuhl für Informatik II, R.W.T.H. Aachen.

0. Introduction

One possible way to define abstract data types is to impose some conditions on a structure, which are of an algorithmic character, i.e. which express some desired properties of computations over a structure, and then to prove that any two structures satisfying these conditions are isomorphic. The reader may find an example of this approach in Salwicki (1978), the data type STACK is consider.

The aim of this paper is to investigate the following problem: what structures can be uniquely defined by algorithmic conditions. For this purpose we choose the logic of effective definitions (in abbreviation LED) introduced in Tiuryn (1979). This logic is based on the notion of an effective definitional scheme due to H. Friedman [2] which seems to be more proper when dealing with abstract structures, than that of a program in algorithmic logic, mainly because of the universal expressive power of the former.

It turns out that the usual program properties like termination, equivalence, partial and total correctness can be expressed in LED by universal formulas. Thus our problem can be faithfully reformulated as follows: what structures are Π_1 implicitly definable in LED.

In this paper we are dealing exclusively with structures of finite similarity type (signature). Existing approaches to abstract data types use explicitly or implicitly the existence of constants in a signature (for example in the case of absence of constants there is no initial algebra). Therefore we have decided to assume that there is always at least one constant in a signature we are working with. This assumption in particular essentially simplifies considerations.

To compare with ADJ's initial algebra approach (ADJ (1977)) note that an initial algebra in a quasi-variety never has proper subalgebras, so it immediately follows from Theorem 2.1 that it is Π_1 implicitly definable in LED.

The paper is divided into two sections. Section 1 contains preliminary notions and definitions concerning LED, and Π_2^0 implicite definability in the standard model of arithmetic. Examples of classes of structures axiomatizable in LED are given there as well. Section 2 brings an answer to the previously stated problem. We give full characterizations of structures Π_1 implicitly definable in LED by:

(0.1) arbitrary sets of Π_1 sentences;

(0.2) r.e. sets of Π_1 sentences;

(0.3) one Π_1 sentence.

1. Preliminaries

1.1. (Syntax of LED). Let $\sigma = \langle F_1, \ldots, F_{k_1}; R_1, \ldots, R_{k_2}; C_1, \ldots, C_{k_3} \rangle$ be a fixed signature with $F_i (1 \leq i \leq k_1)$ being an $m(i)$-ary function symbol; R_j $(1 \leq j \leq k_2)$ begin an $n(j)$-ary predicate symbol; and C_l $(1 \leq l \leq k_3)$ a constant symbol. We assume that $k_1 \geq 1$ and $k_3 \geq 1$ (i.e. there are always constants and function symbols) unless we state otherwise. Denote by T_σ the set of all finite terms over σ without variables. For every n let $T_{\sigma,n}$ (tesp. $F_{\sigma,n}$) denote the set of all finite terms (resp. the set of all open first order formulas) over σ (with equality) with individual variables among $\{x_1, \ldots, x_n\}$.

We shall use a fixed effective coding of each of the sets: $T_\sigma, T_{\sigma,n}, F_{\sigma,n}$ so that the standard notions of recursion theory carry over to these (for details the reader is referred to Shoen-

field (1967) Section 6.6). Later on we will explicitely refer to a coding of T_σ, denoted by h. Hence $h : \omega \to T$ is a bijective enumeration of terms in T_σ.

First we define functional effective definitional schemes. They are going to play a role analogous to that of ordinary terms in first order logic.

Let $n \in \omega$. By an n-ary functional effective definitional scheme (feds) we mean a recursive function $S : \omega \to F_{\sigma,n} \times T_{\sigma,n}$.

For every $m \in \omega$, and for every feds S denote by S_m the first component in the pair S(m).

Formulas of LED are defined inductively. They form the least set LED_σ satisfying the following conditions.

(1.1.1) If S^1, S^2 are fed's then $S^1 \doteq S^2 \in LED_\sigma$;

(1.1.2) If $\alpha, \beta \in LED_\sigma$, then $\alpha \vee \beta$, $\neg\alpha, \alpha \wedge \beta \in LED_\sigma$;

(1.1.3) If x is an individual variable and $\alpha \in LED_\sigma$, then $(\exists x)\alpha$, $(\forall x)\alpha \in LED_\sigma$. The formulas obtained by (1.1.1) and (1.1.2) are called open formulas of LED.

1.3. (Semantics of LED). Denote by K_σ the class of all σ-structures. Let $\mathfrak{A} \in K_\sigma$, $n \in \omega$, and let S be an n-ary feds. The meaning of S in $\mathfrak{A}$ is a partial function $S_{\mathfrak{A}} : A^n \dashrightarrow A$ which is defined as follows: for a given $a \in A^n$ the first formula S_m is chosen (i.e. with least m) such that $\mathfrak{A} \models S_m[a]$; denote by τ_m the second component of the pair S(m), then $\tau_{m\mathfrak{A}}[a]$ is the value of $S_{\mathfrak{A}}$ at a. If for every $m \in \omega$, $\mathfrak{A} \models \neg S_m[a]$ then $S_{\mathfrak{A}}$ is not defined at a. (We say then that $S_{\mathfrak{A}}(a)$ diverges).

Let S^1 and S^2 be fed's. Without loss of generality we may assume that both are n-ary for a certain $n < \omega$. Let $a \in A^n$. We define $\mathfrak{A} \models (S^1 \doteq S^2)[a]$ iff both $S^1_{\mathfrak{A}}(a)$ and $S^2_{\mathfrak{A}}(a)$ are defined and equal.

The meaning of $\mathfrak{A} \models \alpha[a]$ for other LED formulas is defined in the usual way.

A closed formula $\alpha \in LED_\sigma$ is said to be true in a structure $\mathfrak{A}$ if $\mathfrak{A} \models \alpha$. If $T \subseteq LED_\sigma$ is a set of closed formulas and $\mathfrak{A} \in K_\sigma$, then $\mathfrak{A} \models T$ if for every $\alpha \in T$, $\mathfrak{A} \models \alpha$.

R e m a r k s. Every feds S can be viewed as a recursively enumerable infinite definition of a function "by cases". If S is n-ary, then the formula $(\forall x_1,\ldots,x_n) S \doteq x$ expresses the property: "the function defined by S is total". Informally, the same can be expressed by the formula:

$$(\forall x_1,\ldots,x_n)\ (\exists m \in \omega)\ S_m\ .$$

This notation will be used often in the sequel when no confusion will occur.

Formulas with universal quantifiers in the prenex normal from are called Π_1-formulas. It is shown in Tiuryn [6] (section 2.5) that the properties of programs like equivalence, partial and total correctness u.r.t. open conditions can be expressed as Π_1-formulas in LED. Moreover it has been shown there that every Π_1-formula is semantically equivalent (in the sense it determines the same models) to a r.e. set of the formulas of the form $(\forall x_1,\ldots,x_n)\ S \doteq S$. (cf. [6] thm. 3.5).

<u>1.4.</u> Let $\mathfrak{A} \in K_\sigma$. By $D(\mathfrak{A})$ we denote the diagram of $\mathfrak{A}$, i.e. the set of all atomic and negated atomic sentences true in $\mathfrak{A}$.

Let $\mathfrak{A} \in K_\sigma$. Define $L_{\mathfrak{A}} = \langle \tilde{R}_1,\ldots,\tilde{R}_{k_1}, \cong \rangle$, where $\tilde{R}_i$ is a $n(i)$-ary relation in ω defined by

(1.4.1) $(m_1,\ldots,m_{n(i)}) \in \tilde{R}_i$ iff $\mathfrak{A} \models R_i\ (h(n_1),\ldots,h(m_{n(i)}))$ (here $h : \omega \to T_\sigma$ is the coding fixed in 1,1) and $\cong$ is a binary relation in ω defined by

(1.4.2) $m_1 \cong m_2$ iff $\mathfrak{A} \models h(m_1) = h(m_2)$.

Let $PC = P(\omega^{n(1)}) \times \ldots \times P(\omega^{n(k_1)}) \times P(\omega^2)$, and let C be the set of all those L's in PC such that L_{k_1+1} (the k_1+1-th component of L) is a coding, in (1.4.2) sense, of a congruence relation on T_σ (the algebraic structure of T_σ is determined by function and constant symbols from σ in a usual way), and for every $1 \leqslant i \leqslant k_1$ the following holds: for arbitrary $\vec{m}, \vec{p}, \in \omega^{n(i)}$, if $(m_1, p_1) \in \in L_{k_1+1} \oplus \ldots \oplus (m_{n(i)}, p_{n(i)}) \in L_{k_1+1}$ then $\vec{m} \in L_i$ iff $\vec{p} \in L_i$.

Let H denote the class of all σ-structures without proper substructures. It is easy to observe that to each $V \subseteq H$ there corresponds in a natural way $\tilde{V} = \{L_{\mathfrak{A}} : \mathfrak{A} \in V\} \subseteq C$. Moreover, for every $W \subseteq C$ there exists $V \subseteq H$ such that $\tilde{V} = W$.

Observe that if $\mathfrak{A}$ and $\mathfrak{B}$ are isomorphic then $L_{\mathfrak{A}} = L_{\mathfrak{B}}$.

A subset $W \subseteq PC$ is Π^0_2 if there exists a recursive (Δ^0_0) formula $\varphi(R_1, \ldots, R_{k_1}, E, x, y)$ such that $\langle R_1, \ldots, R_{k_1}, E \rangle \in W$ iff

$\forall x \; \exists y \; \varphi(R_1/R_1, \ldots, R_{k_1}/R_{k_1}, E/E, x, y)$
is true in $\langle \omega, 0, 1, +, \cdot \rangle$.

A subset $V \subseteq H$ is called Π^0_2-definable if $\tilde{V}$ is a Π^0_2 set. Observe that the C defined above is a Π^0_2 set as the conditions defining it are Π^0_2. Therefore H is Π^0_2-definable.

Let $\delta : \omega \to F_{\sigma,0}$ be an effective enumeration. The following result can be proved by using standard techniques.

<u>1.5. Proposition.</u> For every recursive (Δ^0_0) formula $\varphi(R_1, \ldots, \ldots, R_{k_1}, E\, x_1, \ldots, x)$ there exists a recursive function $g : \omega P \to \omega$ such that for every $\langle R_1, \ldots, R_{k_1}, E \rangle \in C$ and for every $\mathfrak{A} \in H$ with $L_{\mathfrak{A}} = \langle R_1, \ldots, R_{k_1}, E \rangle$:

$\varphi(R_1/R_1, \ldots, R_{k_1}/R_{k_1}, E/E, x_1/m_1, \ldots, x_p/m_p)$ iff
$\mathfrak{A} \models \delta(g(m_1, \ldots, m_p))$ for all $\langle m_1, \ldots, m_p \rangle \in \omega^p$.

Let $\langle , \rangle : \omega^2 \to \omega$; $\ell, r : \omega \to \omega$ be computable bijective pairing functions, i.e. $\langle \ell(m), r(m) \rangle = m$, $\ell(\langle m_1, m_2 \rangle) = m_1$, $r(\langle m_1, m_2 \rangle) = m_2$. Define $(m)_n$ inductively: $(m)_o = \ell(m)$, $(m)_{n+1} = (r(m))_n$.

1.6. Examples

1.6.1. The natural numbers: $\langle \omega, S, o \rangle$.

The standard model with zero and successor function can be uniquely defined by two axioms

i) $(\forall xy)\,[S(x) \neq o \wedge S(x) = S(y) \to x = y]$,

ii) $(\forall x)\; P \doteq P$,

where $P_n(x) \equiv x = S^n(o)$.

1.6.2. Finite structures: let σ be a signature, then the finite structures in H_σ can be defined by $(\forall x)\; S \doteq S$, with $S_n(x) \equiv [x = h((n)_o) \wedge \varphi_{(n)_1}]$, where φ_m is the proposition which expresses that $\{h(1), \ldots, h(m)\}$ is closed under the operations.

1.6.3. Other examples are: algebraic numbers, computable real numbers [3]; regular algebras over a finite signature [6]. The STACK as presented in Salwicki (1978). These structures can all be uniquely defined by means of Π_1-formulas of LED.

2. Π_1-definability

Recall that K_σ denotes the class of structures of signature σ. $\mathfrak{A} \in K_\sigma$ is called uniquely definable by $T \subseteq LED_\sigma$ if it is, up to isomorphism, the only model of T (in K_σ).

2.1. Theorem. Let σ be a signature containing at least one constant symbol.

The following conditions are equivalent for $\mathfrak{A} \in K_\sigma$.

(i) $\mathfrak{A}$ is uniquely definable by a theory T with axioms of the form $(\forall x_1, \ldots, x_n) S \doteq S$.

(ii) $\mathfrak{A}$ is uniquely definable by a theory T with axioms of the form $(\forall x_1, \ldots, x_n)\ \Phi$, where Φ is an open formula of LED.

(iii) $\mathfrak{A}$ has no proper substructures.

P r o o f.

(i) $\Rightarrow$(ii) has been demonstrated in Tiuryn (1979).

(i) $\Rightarrow$(iii) Recall that σ contains at least one constant.

Let T_σ be the (nonempty) collection of closed terms of σ. The values of $\tau \in T_\sigma$ constitute a substructure $\mathbf{B}$ of $\mathfrak{A}$ which also satisfies all (universal) sentences in T. As $\mathfrak{A}$ is the only model of T up to isomorphism it is isomorphic to $\mathbf{B}$. But then $\mathfrak{A}$ can have no proper substructures since $\mathbf{B}$ has none (iii) $\Rightarrow$(i) Recall that $h : \omega \to T_\sigma$ is an effective enumeration of the closed terms. Let $S_n(x) \equiv x = h(n)$. We take $T = \{(\forall x)\ S \doteq S\} \cup D(\mathfrak{A})$. Clearly $\mathfrak{A} \models T$. Moreover if $\mathbf{B} \models T$ then it has no proper subalgebras (as follows from $\mathbf{B} \models (\forall x)\ S \doteq S$) further $D(\mathbf{B}) = D(\mathfrak{A})$, and so $\mathbf{B} \cong \mathfrak{A}$.

Let H be as in 1.1 H contains those structures which should be considered as possible data types insofar that they can be uniquely defined by means of program properties(Π_1-LED sentences) in view of the preceding result.

Let again $S_n(x) \equiv x = h(n)$. We note that $(\forall x)\ S \doteq S$ defines H within K_σ. We will now consider structures that can be defined uniquely by one Π_1-formula of LED.

<u>2.2. Theorem.</u> Let σ be a signature containing at least one constant symbol.

For $\mathfrak{A} \in H$ the following conditions are equivalent:

(i) $\mathfrak{A}$ can be uniquely defined by means of a formula of the form $(\forall x)S \doteq S$.

(ii) $\mathfrak{A}$ can be uniquely defined by means of a set of formulas $\{\emptyset_i : i \in \omega\}$ where $\emptyset_i$ depends recursively on i and has the form $(\forall x_1, \ldots, x_{k(i)})\psi_i$ with ψ_i an open LED formula.

(iii) $\mathfrak{A}$ is implicitly Π^0_2-definable, i.e. $\{a\}$ is Π^0_2-definable in the sense of 1.4.

P r o o f. We distinguish between finite and infinite $\mathfrak{A}$. If $\mathfrak{A}$ is finite then all of the three conditions are evidently satisfied. Indeed $\mathfrak{A}$ being generated by its constants can be defined by means of a single universal first order formula. Now suppose that $\mathfrak{A}$ is finite, we will show

(i) $\Rightarrow$ (ii) $=$ (iii) $=$ (i).

(i) $\Rightarrow$ (ii) is immediate.

(ii) $\Rightarrow$ (iii) Let $\mathfrak{A}$ be axiomatised by the theory $T = \{(\forall x_1, \ldots, x_{k(i)})\ \psi_i : i \in \omega\}$.

To simplify notation we will assume that for each $i, k(i) = 1$. In the general case the proof is almost identical. We will now informally describe how the condition $\mathfrak{A} \models T$ can be transformed in five steps to a Π^0_2 condition on $L_{\mathfrak{A}}$, (or rather, the components of $L_{\mathfrak{A}}$).

(2.2.0) $$\mathfrak{A} \models T,$$

(2.2.1) $$(\forall i \in \omega)\ [\mathfrak{A} = (\forall x)\psi_i(x)].$$

Now, as $\mathfrak{A} \in H$, the quantification over $\mathfrak{A}$ can be replaced by a quantification over T_G. So we get:

(2.2.2) $$(\forall i \in \omega)(\forall y \in \omega)\ [\mathfrak{A} \models \psi_i\ (h(y))].$$

In [6], (Thm. 3.5) it is shown that an open formula ψ of LED is always equivalent to the infinite conjunction of a recursively

enumerable set of formulas $\{\psi_i^1 : i \in \omega\}$, where ψ_i^1 is of the form $(\forall x_1,\ldots,x_{k(i)})\, S^{1,i} \doteq S^{1,i}$. In our case let $\psi_i(x)$ be equivalent to conjunction of formulas $(S^{i,j} \doteq S^{i,j})(x)$, where $S^{i,j}$ is computed uniformly from i, j.

We obtain:

(2.2.3) $(\forall i \in \omega)(\forall y \in \omega)(\forall j \in \omega)\,[\alpha \models (S^{i,j} \doteq S^{i,j})(h(y))]$.

By definition this is equivalent with:

(2.2.4) $(\forall i \in \omega)(\forall y \in \omega)(\forall j \in \omega)(\exists n \in \omega)\,[\alpha \models S_n^{i,j}(h(y))]$

From 1.4 it follows that a recursive predicate φ of L_α, i, j, n, y can be found such that for all α, i, j, n, y

$$\alpha \models S_n^{i,j}(h(y)) \text{ iff } \varphi(L_\alpha, i, j, n, y).$$

This reduces our condition to:

(2.2.5) $(\forall i \in \omega)(\forall y \in \omega)(\forall j \in \omega)(\exists n \in \omega)\varphi(L_A, i, j, n, y)$.

It follows that $L_\alpha \in C$ is the one and only element of C (satisfying condition) (2.2.5)). Hence $\{L_\alpha\}$ is a Π_2^0-subset of C, and as C is Π_2^0-definable in PC, we find that $\{L_\alpha\}$ is a Π_2^0-subset of PC. So we conclude that α is Π_2^0-definable. (iii) $\Rightarrow$(i) Let α be infinite and Π_2^0-definable (implicitely, uniquely). Let L_α be the only element of PC which satisfies the condition $(\forall x)(\exists y)\varphi(L_\alpha, x, y)$, for a recursive predicate φ. Let IH be the collection of infinite structures in H. We will first find a Π_1 formula of the form $(\forall x)S \doteq S$ which implicitely defines α within IH. To perform this we will rewrite the condition in 7 steps. Using proposition 1.5 one can find a computable function g(x, y) such that for each $x, y \in \omega$, $\varphi(L_\alpha, x, y)$ iff $\alpha \models \delta(g(x,y))$. Here $\delta(g(x,y))$ is a closed propositional formula of LED Starting from the condition

(2.2.6) $$(\forall x)(\exists y)\ \varphi(L_{\alpha}, x, y)$$

we find

(2.2.7) $$(\forall x \in \omega)(\exists y \in \omega)[\alpha \models \delta(g(x,y))].$$

Now this condition is easily seen to be equivalent to

(2.2.8) $$(\forall x \in \omega)(\exists y \in \omega)(\forall z \leq x)[\alpha \models \delta(g(z,(y)_z))].$$

Let $f : A \to \omega$ be a function which assume arbitrarily large values. Then (2.2.8) is equivalent to

(2.2.9) $$(\forall x \in A)(\exists y \in \omega)(\forall z \leq f(x))[\alpha \models \delta(g(z,(y)_z))].$$

This can be replaced by

(2.2.10) $$(\forall x \in A)(\exists y \in \omega)\ [(y)_0 = f(x) \wedge (\forall z \leq (y)_0)[\alpha \models \delta(g(z,((y)_1)_z))]]$$

Now let $f(x)$ be given by $f(x) = \mu i\ [\alpha \models x = h(i)]$.
We see that $u = f(x)$ is equivalent to the following LED proposition $\psi_u(x)$.

$$\psi_u(x) = (x = h(u) \wedge \neg x = h(o) \wedge \ldots \wedge \neg x = h(u-1)).$$

To be precise: $u = f(x)$ iff $\alpha \models \psi_u(x)$. As α is infinite f takes arbitrarily large values, this reduces the condition to

(2.2.11) $$(\forall x \in A)(\exists y \in \omega)([\alpha \models \psi_{(y)_0}(x)] \wedge \forall z \leq (y)_0 [\alpha \models \delta(g(z,((y)_1)_z))]$$

Now let S be the following feds : $S(y) = (S_y, x)$, where

$$S_y(x) = \psi_{(y)_0}(x) \wedge \delta(g(o,((y)_1)_o)) \wedge \ldots \wedge \delta(g((y)_o,((y)_1)_{(y)_o})) .$$

Then (2.2.11) reduces to

(2.2.12) $$(\forall x)\ S \doteq S.$$

This concludes the proof that $\mathfrak{A}$ can be defined uniquely by a formula of the form $(\forall x)\ S \doteq S$ in IH. Next we show that IH Next we show that IH is definable within H by means of a formula $(\forall x)\ S' \doteq S'$.

Let $S'_n\ (x) \equiv$ "$x = h((n)_0)$ and there are at least $(n)_0+1$ different elements in the list $h(0),\ldots,h(n)$".

It is not difficult to see that $\mathbf{B} \models (\forall x) S' \doteq S'$ iff $\mathbf{B}$ is infinite. Finally take $S^*_n(x) \equiv S_{(n)}\ (x) \wedge S'_{(n)}\ (x)$, then $\mathfrak{A}$ is the unique structure in H which satisfies $(\forall x)\ S^* \doteq S^*$.

2.3. Theorem. For each class V of infinite σ-structures without proper substructures the following conditions are equivalent.

(i) V is definable by means of a formula of the form $(\forall x) S \doteq S$.

(ii) V is definable by an r.e. theory $T = \{(\forall \vec{x})\ \psi_i : i \in \omega\}$ with ψ_i open formulas of LED.

(iii) V is Π^0_2-definable.

P r o o f. Similar to the proof of the previous theorem.

R e m a r k. In the precence of constants the above results give complete answers concerning the power of Π_1-LED formulas. If there are no constants we can only prove the following partial result.

2.4. Proposition. Let $\mathfrak{A}$ be uniquely definable by a set T of Π_1-formulas. Then $\mathfrak{A}$ has no proper substructures.

P r o o f (Sketch). Each substructure $\mathbf{B} \subseteq \mathfrak{A}$ satisfies T as well. Therefore $\mathbf{B} \subseteq \mathfrak{A}$ implies $B \cong \mathfrak{A}$. Especially we may take $\mathbf{B}$ finitely generated. We conclude that $\mathfrak{A}$ is finitely generated. Now suppose that $\mathbf{B}$ is a proper substructure of $\mathfrak{A}$. We construct a sequence of structures $\mathfrak{A}_0, \mathfrak{A}_1, \ldots,$ such that:

(i) $\mathfrak{A}_0 = B$, $\mathfrak{A}_1 = \mathfrak{A}$,

(ii) for each n, $\mathfrak{A}_n \subsetneq \mathfrak{A}_{n+1}$.

(iii) for each n there is an isomorphism $f_n : \mathfrak{A}_{n+1} \to \mathfrak{A}_{n+2}$ such that $f_n(\mathfrak{A}_n) = \mathfrak{A}_{n+1}$.

(informally: $\mathfrak{A}_n : \mathfrak{A}_{n+1} = B : \mathfrak{A}$).

Now $\mathfrak{A}' = {}_n\mathfrak{A}_n$ is not finitely generated. So it is not isomorphic to $\mathfrak{A}$. On the other hand let $(\forall x_1,\dots,x_k)\varphi$ be a universal formula satisfied by $\mathfrak{A}$, and let $b_1,\dots,b_k \in |\mathfrak{A}'|$. Take n so large that $b_1,\dots,b_k \in |\mathfrak{A}_n|$. Then $\mathfrak{A}_n \models \varphi[b_1,\dots,b_k]$ (as $\mathfrak{A}_n \cong \mathfrak{A}$) thus $\mathfrak{A}' \models \varphi[b_1,\dots,b_k]$. We conclude $\mathfrak{A}' \models T$, and this contradicts the assumptions.

2.5. Problem. Does the converse of 2.4 hold? (presumably not).

References

[1] ADJ: G o g u e n, J. A., T h a t c h e r, J.W., W a g n e r, E. G., An initial algebra approach to the specification correctness and implementation of abstract data types, in: R.Yeh (ed.) Current Trends in Programming Methodology, Vol. 3, Data Structuring, Prentice-Hall, Automatic Computation Series (1977).

[2] F r i e d m a n, H., Algorithmic procedures, generalized Turing algorithms, and elementary recursion theory, in R.O. Gandy and C.M.E. Yates (eds.), Logic Colloqium '69, North-Holland Publ. Co. (1971).

[3] K r e c z m a r, A., Effectivity problems of algorithmic logic, Ph.D.Thesis, Warsaw University (1973).

[4] S a l w i c k i, A., On algorithmic theory of stacks, in: J. Winkowski (ed.), MFCS '78, Lecture Notes in Computer Sc.64, Springer Verlag (1978).

[5] S h o e n f i e l d, J. R., Mathematical Logic. Addison-Wesley Publ. Co. (1967).

[6] T i u r y n, J., Logic of Effective Definitions, R.W.T.H. Aachen, Research Report No. 55. To appear in Fundamenta Informaticae (1979).