Notre Dame Journal of Formal Logic
Volume XVI, Number 4, October 1975
NDJFAM

NON-DEFINABILITY OF CERTAIN SEMANTIC PROPERTIES OF PROGRAMS

RICHARD A. DeMILLO

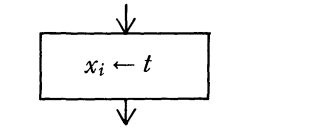
1 *Introduction* This paper has a two-fold purpose. First, we address ourselves to the problem of describing programs with languages which are, in some sense, "related" to the programming language in question. In particular, we examine generalizations of the halting and equivalence problems and show that no uniform first order methods of description can exist. Second, we illustrate the very natural sort of correspondence which may be established between results in mathematical logic and a class of foundational problems in computer science. What we will refer to as *programs*, are variously called *program schemata* and *abstract programs* in the literature. The question of description or *definability* has been examined by a number of authors. Positive results relating to definability of termination appear in Engeler [2] and Manna [4]. Equivalence and decision problems are given thorough treatment in Luckham, Park, and Paterson [3]. The import of definability results is indicated in Manna and Waldinger [5]. Details of the logical results are available in Bell and Slomson [1] and vanFraassen [6].

2 *The Languages* $\mathcal{L}$ *and* $\mathcal{P}(\mathcal{L})$ By the language $\mathcal{L}$ we mean a first order predicate calculus with identity and the following primitive alphabet:

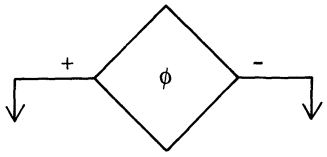
1. denumerably many individual variables: $x_0, x_1, \ldots$ (in some cases a variable may be denoted by y_j);
2. function symbols of specified addicity and unspecified number: $f_0, f_1, \ldots$;
3. predicate symbols of specified addicity and unspecified number: $q_0, q_1, \ldots$.

Function symbols taking 0—arguments are also called individual constants (denoted $a_0, a_1, \ldots$). When f is either a function or a function symbol, we let $f^n(z)$ be z when $n = 0$ and $f(f^{n-1}(z))$ when $n > 0$. $\mathcal{L}$ is completely specified when we set down its rules of formation, logical axioms, and rules of inference, all in the usual fashion. $\mathcal{P}(\mathcal{L})$ is a programming language derived from $\mathcal{L}$. About $\mathcal{P}(\mathcal{L})$'s exact structure we can be quite informal. Programs in $\mathcal{P}(\mathcal{L})$ are taken to be constructed from the following components:

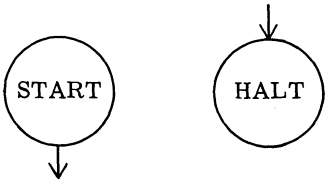
Received May 7, 1973



for x_i a variable, t a term



for ϕ a formula



The intended interpretations of these components are obvious and well-known. The geometric representation also allows us to be informal about "tracing" through programs. In particular, if a trace through a program has allowed us to reach a component C, then:

1. if C is an assignment or a start component, C's successor in the trace is given by C's exit arrow;
2. if C is a decision component, C's possible successors in the trace are given by the respective (+)-labelled and (-)-labelled exit arrows;
3. if C is a halting component, C has no successor.

It will be convenient to distinguish *input* variables (y_i) in programs from *program* variables (x_i) in the style of Manna [4]. We will assume that no confusion can arise, in writing programs, from this distinction. Hence, all programs are idealized with respect to "free" and "bound" variables.

3 *Models* A *realization* $\mathfrak{A} = \langle A, I\rangle$ of $\mathcal{L}$ is a nonempty set A together with a correspondence I between n-ary function symbols and m-ary predicate symbols in $\mathcal{L}$ and, respectively, n-ary functions (including nullary functions or individuals of $\mathfrak{A}$) on A and m-ary relations on A. The mapping I is usually understood and is omitted in referring to realizations. Since $\mathcal{L}$ is fixed throughout, we refer to $\mathfrak{A} = \langle A, F, R\rangle$, where F is the set of functions in the range of I and R is the set of relations in the range of I, as a realization of $\mathcal{L}$. A realization $\mathfrak{A}$ of $\mathcal{L}$ is of cardinal α iff $\mathrm{Card}(A) = \alpha$. Terms of $\mathcal{L}$ are interpreted in realizations as follows. Let t be a term, $\mathfrak{A}$ a realization of $\mathcal{L}$, and $\bar{a} \in A^{\omega}$ an arbitrary ω-termed sequence in A. Then

$$t(\bar{a}) = \begin{cases} \bar{a}(i), \text{ if } t = x_i \\ I(f)(t_1(\bar{a}), \ldots, t_n(\bar{a})), \text{ if } t = f(t_1, \ldots, t_n). \end{cases}$$

Classes of models of formulas of $\mathcal{L}$ are defined with respect to the relation $\models$. Let ϕ be a formula of $\mathcal{L}$, $\mathfrak{A}$ a realization of $\mathcal{L}$, and $\bar{a} \in A^{\omega}$. Then $\mathfrak{A} \models \phi[\bar{a}]$ iff:

1. ϕ is an atomic formula $t_1 = t_2$ or $q_i(t_1, \ldots, t_n)$ and $t_1(\bar{a}) = t_2(\bar{a})$, or $\langle t_1(\bar{a}), \ldots, t_n(\bar{a})\rangle \in I(q_i)$, respectively;
2. ϕ is $\sim\psi$ and $\mathfrak{A} \not\models \psi\ [\bar{a}]$;
3. ϕ is $\psi_1 \vee \psi_2$ and $\mathfrak{A} \models \psi_1[\bar{a}] \vee \mathfrak{A} \models \psi_2[\bar{a}]$;

4. ϕ is $(\forall x_i)\psi$ and for each $a \in A$, if $\bar{a}^i_a$ differs (if at all) from $\bar{a}$ only in that $\bar{a}^i_a(i) = a$, then $\mathfrak{A} \models \psi\,[\bar{a}^i_a]$.

If $\mathfrak{A} \models \phi[\bar{a}]$ for all $\bar{a} \in A^\omega$, then $\mathfrak{A} \models \phi$ and $\mathfrak{A}$ is called a *model* of ϕ. Accordingly, if X is a set of formulas of $\mathcal{L}$, then $\mathfrak{A}$ is a model of X iff

$$\mathfrak{A} \in \bigcap_{\phi \in X} \{\mathfrak{B} \mid \mathfrak{B} \models \phi\}.$$

The following model theoretic results are relevant.

Theorem 1 (Compactness) *A set X of sentences of $\mathcal{L}$ has a model iff each finite subset of X has a model.*

Theorem 2 (Upward Löwenheim-Skolem) *If a set X of sentences of $\mathcal{L}$ has a denumerable model, then X has a model of every infinite cardinal.*

4 *Execution Sequences* Let p be a program in $\mathcal{P}(\mathcal{L})$ and $\mathfrak{A}$ be a realization of $\mathcal{L}$. We define a (finite or infinite) *execution sequence*,

$$\pi = \langle \pi(i) \mid i \geq 0 \text{ and } \pi(i) \in A^\omega \rangle,$$

and for each j such that $\pi(j)$ exists, the j'th component of p as follows:

1. the 0'th component of p is the successor of the start component;
2. if the j'th component, C, of p is an assignment component $x_i \leftarrow t$, then

$$\pi(j+1)(k) = \begin{cases} t(\pi(j)), \text{ if } k = i \\ \pi(j)(k), \text{ otherwise} \end{cases}$$

and the $j+1$st component is the successor of C;
3. if the j'th component, C, of p is a decision component ϕ, then $\pi(j+1) = \pi(j)$ and the $j+1$st component of p is the (+)-successor of C if $\mathfrak{A} \models \phi[\pi(j)]$ and the (-)-successor of C if $\mathfrak{A} \not\models \phi[\pi(j)]$;
4. if the j'th component, C, of p is a halting component, then $\pi(j)$ is the last term of π and $\mathrm{Max}(\pi) = j$.

If $\mathrm{Max}(\pi)$ exists and $\pi(0) = \bar{a}$, we write $\mathsf{T}(\mathfrak{A}, \bar{a}, p)$ and say that p *terminates* in $\mathfrak{A}$ at $\bar{a}$. If π and π' are respective execution sequences for p and p' in $\mathfrak{A}$, $\pi(0) = \pi'(0) = \bar{a}$, then we write $\mathsf{E}(\mathfrak{A}, \bar{a}, p, p')$ and say that p and p' are *equivalent* in $\mathfrak{A}$ at $\bar{a}$ just in case $\pi(\mathrm{Max}(\pi)) = \pi'(\mathrm{Max}(\pi'))$ whenever either side is defined. If we agree to write $\mathsf{E}(\mathfrak{A}, p, p')$ when $\mathsf{E}(\mathfrak{A}, \bar{a}, p, p')$ for all $\bar{a} \in A^\omega$ and if for each realization $\mathfrak{A}$ of $\mathcal{L}$ we have $\mathsf{E}(\mathfrak{A}, p, p')$ then p and p' are related by $\equiv$, the strong equivalence of Luckham, Park, and Paterson [3], up to nonconflicting rearrangements of variables. There is no generality lost in assuming that variables appear uniformly in p and p', so that E applies as (actual) equivalence.

5 *Definability of Properties* E and T are examples of semantic properties of programs. A property is definable in $\mathcal{L}$ when it can be totally described in $\mathcal{L}$:

1. T *is definable in* $\mathcal{L}$ iff for each p, a program, there exists a formula ϕ of

$\mathcal{L}$, whose only free variables are the input variables of p, such that $\mathfrak{A} \models \phi[\bar{a}]$ just in case $\mathsf{T}(\mathfrak{A}, \bar{a}, p)$, whenever $\mathfrak{A}$ is a realization of $\mathcal{L}$ and $\bar{a} \in A^{\omega}$;
2. E *is definable in* $\mathcal{L}$ iff for each pair of programs p and p' there exists a formula ϕ of $\mathcal{L}$, whose only free variables are the input variables of p and p', such that $\mathfrak{A} \models \phi[\bar{a}]$ just in case $\mathsf{E}(\mathfrak{A}, \bar{a}, p, p')$, whenever $\mathfrak{A}$ is a realization of $\mathcal{L}$ and $\bar{a} \in A^{\omega}$.

These definitions easily generalize to arbitrary semantic properties. A property which is not definable in $\mathcal{L}$ is said to be a *nondefinable* semantic property.

6 *Nondefinability of Termination* In this section we apply Theorem 1 to the programs in Figure 1 (see p. 587) to obtain the following:

Theorem 3 T *is a nondefinable semantic property of programs.*

Proof: We suppose that T is definable to contradict compactness. In particular, consider Figure 1. There exists a formula ϕ of $\mathcal{L}$ such that $\mathfrak{A} \models \phi[\bar{a}]$ iff $\mathsf{T}(\mathfrak{A}, \bar{a}, p)$. Since p has no input variables ϕ is a sentence. Likewise, for each $n \geq 0$, there is a sentence ϕ_n of $\mathcal{L}$ such that $\mathfrak{A} \models \phi_n[\bar{a}]$ iff $\mathsf{T}(\mathfrak{A}, \bar{a}, p_n)$. Let

$$P = \{p\} \cup \{p_n \mid n \geq 0\}$$
$$X(P) = \{\phi\} \cup \{\phi_n \mid n \geq 0\}.$$

Each finite subset of X has a model. Indeed, let $X(P')$ be such a subset. Either $P' = \{p\}$ or there is a largest $m \geq 0$ such that $p_m \in P'$. We let $\mathfrak{N}$ be such that $N = \{0, 1, 2, \ldots\}$, the natural numbers, with $I(f)$ the successor function, $I(a_0) = 0$, and $I(a_1) = m + 1$. Then $\mathsf{T}(\mathfrak{N}, \bar{n}, p')$ for any $p' \in P'$, $\bar{n} \in N^{\omega}$. Hence $\mathfrak{N}$ is a model of $X(P')$.

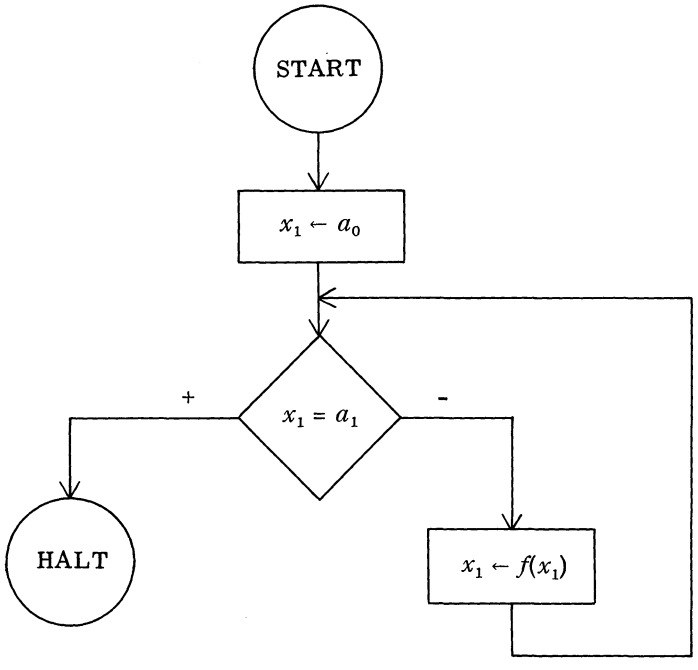
Now, consider $X(P)$. The sentences $X(P)$ have a model iff the programs in P simultaneously terminate in a realization $\mathfrak{A}$. But p terminates in $\mathfrak{A}$ just in case at least one of the statements in the infinite list (*) is true in $\mathfrak{A}$, and $\{p_n \mid n \geq 0\}$ simultaneously terminate in $\mathfrak{A}$ just in case none of the statements in (*) hold in $\mathfrak{A}$. There is clearly no such $\mathfrak{A}$. Therefore, each finite subset of $X(P)$ has a model, but $X(P)$ itself has no model.

$$\left.\begin{array}{r} I(a_0) = I(a_1) \\ I(f)(I(a_0)) = I(a_1) \\ \vdots \\ I(f)^n(I(a_0)) = I(a_1) \\ \vdots \end{array}\right\} (*)$$

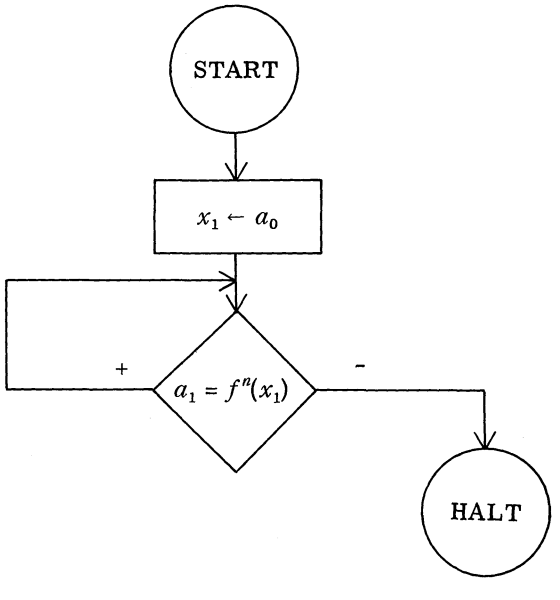
7 *Nondefinability of Equivalence* In this section we apply Theorem 2 to the programs in Figure 2 (see p. 588) to obtain the following:

Theorem 4 E *is a nondefinable semantic property of programs.*

Proof: As in Theorem 3, we suppose that E is definable to contradict the Löwenheim-Skolem property. Consider the programs in Figure 2. Identify y_1 with x_j and let $\mathfrak{N}$ be as above. Then for $n \geq 0$, $I(f)^n(I(a_0)) = n$. So, for

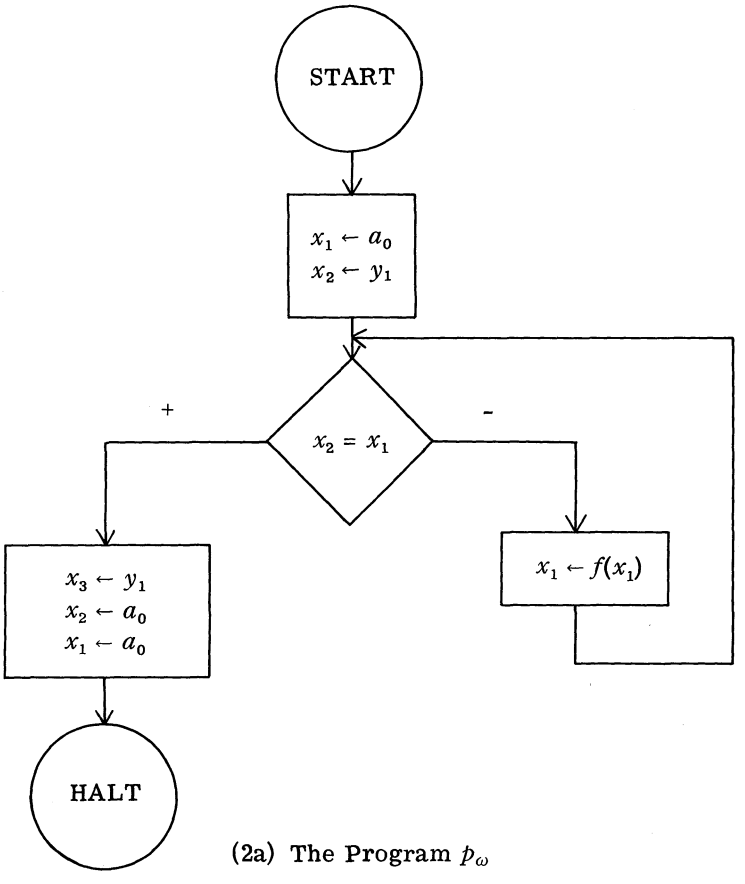


(1a) The Program p

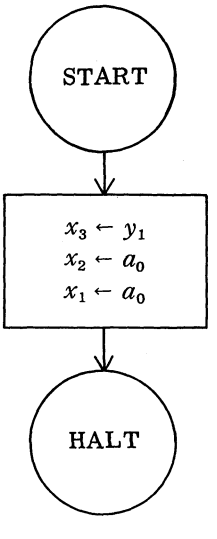


(1b) For each $n \geq 0$, the Program p_n

Figure 1



(2a) The Program p_ω



(2b) The Program p_f

Figure 2

each value n of x_j, p_ω terminates. That is, for π_ω an execution sequence for p_ω in $\mathfrak{N}$, $\mathrm{Max}(\pi_\omega)$ exists and

$$\pi_\omega(\mathrm{Max}(\pi_\omega))(j) = \pi_\omega(\mathrm{Max}(\pi_\omega))(3) = n$$
$$\pi_\omega(\mathrm{Max}(\pi_\omega))(1) = \pi_\omega(\mathrm{Max}(\pi_\omega))(2) = I(a_0).$$

Similarly, for each value n of x_j, p_f terminates, and if π_f is an execution sequence for p_f in $\mathfrak{N}$, $\mathrm{Max}(\pi_f)$ exists and

$$\pi_f(\mathrm{Max}(\pi_f))(j) = \pi_f(\mathrm{Max}(\pi_f))(3) = n$$
$$\pi_f(\mathrm{Max}(\pi_f))(1) = \pi_f(\mathrm{Max}(\pi_f))(2) = I(a_0).$$

Hence for each $\bar{n} \in N^\omega$, $\mathsf{E}(\mathfrak{N}, \bar{n}, p_\omega, p_f)$. Assuming that E is definable, for each $n \in N$, if $\bar{n} \in N^\omega$ is such that $\bar{n}(j) = n$, then $\mathfrak{N} \models \phi[\bar{n}]$ for appropriate ϕ. Hence, $\mathfrak{N} \models (\forall x_j)\phi$. Finally, observe that $(\forall x_j)\phi$ cannot have an uncountable model. This follows, since for each model $\mathfrak{A}$ of $(\forall x_j)\phi$, $\mathfrak{A} \models \phi[\bar{a}^j_a]$ for all $a \in A^\omega$, $a \in A$ and by the definition of ϕ and our choice of p_ω,

$$a \in \{I(f)^n(I(a_0)) \mid n \geq 0\},$$

a denumerable set. But this contradicts the Löwenheim-Skolem property, since $(\forall x_j)\phi$ is a sentence with a denumerable model but no uncountable model.

8 *Conclusion* By Theorems 3 and 4, the formulas defining the properties T and E cannot uniformly exist, if they are to be first order. Intuitively speaking, languages which either restrict variables of quantification to individual variables or restrict formulas to finite strings of symbols are not sufficient to definability.

In fact, the programs in Figure 1 correspond to the infinitary formulas

$$\bigvee_{n<\omega} a_1 = f^n(a_0),\ a_1 \neq a_0,\ a_1 \neq f(a_0),\ \ldots$$

for which compactness fails (see, e.g., Bell and Slomson [1]), and the program in Figure 2a corresponds to the infinitary formula

$$(\forall x_j)\left(\bigvee_{n<\omega} x_j = f^n(a_0)\right)$$

for which the Upward Löwenheim-Skolem Theorem fails.

A subsequent paper will explore this analogy in more detail.

REFERENCES

[1] Bell, J. L., and A. B. Slomson, *Models and Ultraproducts*, North-Holland, Amsterdam (1969).

[2] Engeler, E., "Algorithmic properties of structures," *Mathematical Systems Theory*, vol. 1 (1967), pp. 183-195.

[3] Luckham, D. C., D. M. R. Park, and M. S. Paterson, "On formalized computer programs," *Journal of Computer and System Sciences*, vol. 4 (1970), pp. 220-249.

[4] Manná, Z., "Properties of programs and the first-order predicate calculus," *Journal of the Association for Computing Machinery*, vol. 16 (1969), pp. 244-255.

[5] Manna, Z., and R. J. Waldinger, "Towards automatic program synthesis," in E. Engeler, ed., *Symposium on Semantics of Algorithmic Languages*, Springer, Heidelberg (1971).

[6] vanFraassen, B. C., *Formal Semantics and Logic*, MacMillan, New York (1971).

University of Wisconsin—Milwaukee
Milwaukee, Wisconsin