Independence Results in Computer Science?*

DEBORAH JOSEPH AND PAUL YOUNG

Computer Sciences Department, Purdue University,
Lafayette, Indiana 47907

Received July 21, 1980; revised December 18, 1980

The adequacy of weak theories of arithmetic as axiomatic bases for Computer Science is investigated. The formal system studied in Lipton (*in* "Proceedings, 19th Symposium of the Foundations of Computer Science," pp. 193–200, 1978) is shown to be inadequate for Computer Science in the sense that it has a model with the following defects: (i) finite (i.e., bounded) sets may be undecidable in the model, (ii) standard recursively enumerable sets which are undecidable may have linear time decision procedures in the model, and (iii) programs which simply loop a bounded number of times may not terminate in the model (demonstrating the inadequacy of the formal system for proving program termination).

1. INTRODUCTION

In recent papers, Lipton [23], DeMillo and Lipton [5, 6], Homer and Reif [18], Joseph [19], and Joseph and Young [20] have considered the consequences of using formal systems which are weaker than Peano Arithmetic for investigating and analyzing problems which arise in Computer Science. In another paper, O'Donnell [27] has shown that certain natural termination statements about programs written in strongly typed languages are independent of full Peano Arithmetic. The O'Donnell and Lipton papers follow an earlier suggestion of Hartmanis [15] and Hartmanis and Hopcroft [14] that the $P = NP$ question and similar questions may be independent of Peano Arithmetic. However, the O'Donnell and Lipton papers established the first "natural" independence results for Computer Science. (These results followed the results of Paris and Harrington establishing the independence of an extension of the finite version of Ramsey's Theorem from Peano Arithmetic [28, 29]. McAloon [26] contains a collection of articles discussing and extending the Paris and Harrington results.)

In [23], Lipton shows that certain statements of general complexity theory which are provable in Peano Arithmetic are nevertheless independent of weaker, restricted

* This paper contains minor corrections, changes and revisions to a preliminary version which appeared in the Proceedings of the 12th Annual ACM Symposium on the Theory of Computing (May 1980). Part of the work reported here appears in the first author's 1981 doctoral dissertation writen at Purdue University under the direction of the second author. The paper was prepared with the support of NSF Grant MCS-7609232 AO2, Purdue University. The first author's current address is: Computer Science Dept., Cornell University, Ithaca, N. Y., 14853.

0022-0000/81/050205–18$02.00/0

subsets of Peano Arithmetic which do not permit the use of complicated, "nonconstructive," forms of inductive reasoning. Lipton argues from this that because Computer Scientists do not normally (and perhaps should not) rely on such nonconstructive proof methods, such statements, which require more complicated inductions for their proofs, should be considered to be outside the proper domain of Computer Science. He argues that such independence results help explain why these results of general complexity theory may have little relevance to Computer Science. In general he argues that a "constructive" subset of the Peano axioms should be adequate for Computer Science.

On the other hand, in recent work by O'Donnell [27] (and in related work by Fortune [9]) it is show that for strongly typed languages such as MODEL, the question of termination for what in less sophisticated languages would be trivial straight-line code is independent of an extension of the standard axioms for Peano Arithmetic. O'Donnell's work raises the disturbing possibility that increased programming language sophistication will lead to inherently difficult program management and analysis. To overcome this difficulty, O'Donnell suggests that it may be necessary to increase the power of the formal metasystem (e.g., Peano Arithmetic) which is normally used for program analysis and verification.

Although there has been considerable additional work discussing limitations of formal proof techniques for the theory of computation [8, 11, 37] and for Computer Science [1, 12–15, 30], these papers show only very general consequences of incompleteness: the stated results hold for all sufficiently powerful formal systems for Computer Science. Only the work of O'Donnell and of Lipton directly addresses the question of just how powerful axioms for Computer Science should be, and these two authors make rather radically different suggestions.

This paper addresses the latter question: How powerful should a set of axioms be if it is to be adequate for Computer Science? In particular, in this paper we investigate the adequacy of the system of [23] as a formal system for Computer Science.

2. Recursive Decidability and Algorithm Efficiency in Basic Number Theory

The metasystem studied by Lipton is obtained by taking as axioms all the Π_2 theorems[1] of Peano Arithmetic. This subtheory is called "Constructive Arithmetic" by Lipton. However, it has been studied extensively by earlier authors, [10, 17, 25, 31, 33], and is now generally referred to as Basic Number Theory (B). Since, "constructive" often has a different meaning, for example as used by Heyting [16] or by Constable [3], we shall follow the earlier convention. As Lipton points out, B is a

[1] There is some leeway possible in the defining syntax of the class of Π_2 statements, although the definition is invariant under reasonable changes [17]. Our results will hold for any syntactically recursive class such that if $(\forall x)(\exists y)\,\sigma(x, y)$ is any true formula of the class, then there is a total recursive function f such that $\sigma(x, f(x))$ is true for all x.

fairly powerful subsystem of Peano Arithmetic. For example, it is adequate for proving the consistency of every subtheory of arithmetic obtained by taking only the Σ_n axioms of Peano Arithmetic. It is also the case that within B one can prove not only the equivalence of all the standard models of computation, but also that they are polynomially related. That is, one can prove that the class of functions computable by Turing machines, RAMs or Markov algorithms is exactly the class of partial recursive functions and that the computation times are polynomially invariant. Therefore within models of B a function is easily computable if and only if there is a Turing machine which easily computes it. B also properly contains those simple, finitely axiomatizable theories, such as Robinson's System [34], known to be adequate for obtaining the Gödel incompleteness results. On the other hand, Lipton shows that this system is not adequate for establishing the existence of a (provably) decidable set such that neither the set nor its complement has an infinite easy to decide subset. In addition, he shows that B is inadequate for obtaining the Blum speed-up results for standard sets.[2]

In interpreting these results, Lipton states that "It is standard folklore that all the normal work of mathematicians who deal with finite objects can be proved in this theory" and that "we as Computer Scientists rarely employ nonconstructive methods" (i.e., those not formalizable within B). He goes on to argue that the strength of B serves as evidence that complexity theoretic results which are independent of B are perhaps not relevant to Computer Science, or in any case require proof techniques which are more complex than those of interest to Computer Scientists. However, we suspect that such independence results will often suggest the independence of even more basic statements. The results of this paper illustrate that this is the case not only for B but also for slightly stronger theories. By constructing a model in which some undecidable sets become decidable in linear-time and in which closely related bounded sets are undecidable, we show that, in some sense, theories like B are inadequate for distinguishing very fundamental concepts of decidability and algorithm efficiency. We believe that results such as ours indicate the inadequacy of such weak theories as an axiomatic basis for Computer Science: very few Computer Scientists will accept as adequate axioms which allow bounded sets to be undecidable and allow undecidable sets to be decidable in linear-time.

On the other hand, the fact that such simple decidability and complexity statements as well as the more abstract results discussed in [23] are independent of such weak theories may still be of interest for: (i) better understanding exactly how powerful proof tools must be in Computer Science, (ii) classifying the "proof-theoretic" complexity of results in Computer Science, and (iii) perhaps as precursors for independence results either for richer axiom systems or for more interesting statements. Similar sentiments are expresed in [6].

Before presenting our results, it is useful to discuss the proof of Lipton's result and extensions which have been made to it. The chief result of [23] is that it is consistent with B to believe that every (provably) recursive set, R, has an infinite easy to decide

[2] For somewhat related work, see [37, Theorem 3].

subset either in R or in $\bar{R}$. In [22], Leivant extends this by observing that the set need not be provably recursive and by showing that even then the result is independent not just of B, but of the stronger theory, called T_{Π_2}, whose axioms are all *true* Π_2 arithmetic statements. (That is, all true sentences of the form $(\forall x)(\exists y)\,\sigma(x,y)$, with σ bounded.) Lipton's and Leivant's proofs use essentially the same techniques—both use a compactness argument to construct a nonstandard model in which either the set R or its complement has an infinite easily decidable subset. Both of their proofs show slightly more than is stated in their theorems. For this reason and because our proof builds on theirs, it is useful to sketch a proof of their results. The outline which we present will omit details in an attempt to point out intuitively what are the essential features of their argument and to motivate our proof which follows.

We begin with the standard model of Peano arithmetic, N, and form a nonstandard model, N_1, of the theory of N which contains a nonstandard constant a_0. Next we consider the sequence $\{g^i(a_0): i \in N\}$ for a monotonically increasing, total recursive function g. Clearly, for any such function this sequence is a subset of N_1. Suppose that we choose g to be a total recursive function which majorizes every provably recursive function and we form the structure $M = \{x: x \in N_1 \;\&\; x < g^i(a_0)$ for some $i \in N\}$. Since g majorizes the Skolem functions for all of the provable Π_2 sentences, M is a model of B. Alternatively, if we choose g to be a nonstandard total recursive function which majorizes every standard total recursive function and form M in the same manner, then M is a model of T_{Π_2}. (In this case, care must be taken in the construction of N_1 to ensure that a program (index) for g is less than $g^i(a_0)$ for some $i \in N$ since later in the proof we will require that g be computable in M.) At this point some observations can be made:

First let $G = \{g^i(a_0): i \leqq a_0\}$ and let F be any subset of G. In N_1 there is a number (in effect a table) b_F such that for all $i \leqq a_0$, $g^i(a_0) \in F$ iff $p_i \mid b_F$, where p_i is the ith prime. But no matter what F is, b_F is the product of fewer than a_0 primes, and hence $b_F \in M$. In fact, by the choice of g, $b_F < a_1$. Now in the model M, consider any set S. Either S or $\bar{S}$ intersects G infinitely often. Without loss of generality, suppose the former. Then $S \cap G$ is just the restriction to the set M of some subset F of G. Thus for this set F, $x \in S \cap G$ iff $x \in F$. Since g grows at least exponentially, if we also choose g with the additional property that it is linearly honest, then given $x \in M$, in linear-time we can test whether there is an $i \in N$ such that $g^i(a_0) = x$ and if one is found, we can then use b_F to test whether $x \in S$ in linear-time. Thus in effect, Lipton's and Leivant's proofs really show that there is a model of B or T_{Π_2} in which *every* set has an infinite easily decidable subset in either it or its complement.

Of course if the set S is arithmetically definable, then the *interpretation* of its arithmetic definition in the model N_1 when restricted to the set M need not be the same as the *interpretation* of its arithmetic definition in the model M. On the other hand, if S is (provably) recursive then S behaves nicely with respect to membership in both N and M. That is,

$$\text{for every } x \in N, N \models x \in S \text{ iff } M \models x \in S;$$

$$\text{for every } x \in M, M \models x \in S \text{ iff } N_1 \models x \in S.$$

The first of these sentences is true simply because our axioms include all of the true (or provable in the case of B) Π_2 sentences. The second follows from the fact that g majorizes all of the (provably in the case of B) recursive functions, hence for any standard recursive set S it majorizes the runtime of a decision procedure for S, forcing it to behave in the same manner in M as in N_1.

In general, one can only show that a set behaves nicely with respect to membership if the set is (provably) recursive. Nevertheless, we will show that there are sets which are undecidable in every model of *PA* but in models similar to M not only contain infinite easily decidable subsets but are themselves easily decidable. What's more they behave nicely with respect to membership.

The class of undecidable sets which we construct will be a subclass of the simple sets. In this regard it is interesting to note that the most extensively studied models for B and T_{Π_2} are the *existentially complete models*. It is well known that in passing from the standard model to existentially complete models of T_{Π_2} the complements of all simple sets become bounded but remain undecidable ([17]). Since we want a model in which undecidable sets remain infinite and coinfinite but become rapidly decidable, and since we will be working with simple sets, the models we construct will not be existentially complete. It is also the case that in the existentially complete models the natural numbers are bounded and definable but neither recursive nor recursively enumerable, while in our model they are recursively enumerable in increasing order but not recursive. Nevertheless, logicians familiar with B and T_{Π_2} will find results such as ours not surprising.

We now state our main result:

THEOREM. *There exists a provably (in PA) undecidable set S and a model M of T_{Π_2} such that*:

[A] *In M, S is infinite, coinfinite, and decidable in linear-time*;

[B] *In M, there exists a finite (bounded) set F which is definable in $L(M)$ but is undecidable in M. Furthermore, in M, F is recursively enumerable in increasing order and can be put in effective one-to-one correspondence with $\bar{S}$.*

[C] *Finally, S behaves nicely with respect to membership if the domain of the model is restricted to standard elements. That is, if $a \in N$ then $M \vDash a \in S$ if and only if $N \vDash a \in S$.*

Outline of the proof. The details of the proof are presented in Sections 3 and 4, however, a brief outline of the proof is given here. As a point of departure, we recall the fact mentioned earlier that in existentially complete models of T_{Π_2} the complements of all simple sets become finite. This suggests that with luck we should be able to force some subclasses of simple sets to be infinite and coinfinite but still rapidly decidable in some models of T_{Π_2}. Subclasses of simple sets have been extensively studied in the literature.[3] Unfortunately, none of these subclases seems exactly tailored to our needs, so we introduce still another subclass of simple sets:

[3] See, for example, [4], [7] or [35].

2.1. Definition. A set S is *still-another-simple-set* if there is an index (program) e such that the domain of $\varphi_e = S$ and for every linearly honest, strictly monotonically increasing, total function h which grows at least as rapidly as $k_e * x^3$ (where k_e is a constant depending on e) and for every integer n there exist elements $t_0, t_1, \ldots, t_n$ of $\bar{S}$ such that for all $i < n$:

(1) $t_0 > n$,

(2) $t_{i+1} = h(t_i)$,

(3) $(\forall x)[t_i < x < t_{i+1} \Rightarrow x \in S]$,

(4) $(\forall x)[x < t_i \,\&\, x \in S \Rightarrow \Phi_e(x) < t_{i+1}]$.[4]

Such sets are obviously recursively enumerable, however they are all undecidable; they are in fact hypersimple. However, given a still-another-simple-set, S, it is not hard to construct a model M of T_{Π_2} in which S is decidable in linear-time. One first constructs a nonstandard model, N_1, of full arithmetic by adding new constants a_0, a_1, $a_2, \ldots$, axioms which force

(1) $a_{i+1} = g(a_i)$,

(2) $a_i \in \bar{S}$,

(3) $(\forall x)[a_i < x < a_{i+1} \Rightarrow x \in S]$,

(4) $(\forall x)[x < a_i \,\&\, x \in S \Rightarrow \Phi_e(x) < a_{i+1}]$,

and axioms which force g to be a nonstandard but honest monotone recursive function which, in the model N_1, majorizes every standard recursive function. Next if we let

$$M = \{x \mid N_1 \models x < a_i \text{ for some } i\}$$

then in the same fashion as Leivant, by using the fact that g majorizes all standard total recursive functions, we can easily show that M is a nonstandard model of T_{Π_2}. Property (4) above then asserts that g majorizes the runtime of Φ_e, making S decidable.

To complete the proof of part [A] of the main theorem, we must of course know that still-another-simple-set exists. The construction of such a set is by a straightforward, purely recursion theoretic, moveable markers argument included in Section 4.

To prove part [B] of the theorem, we show that one can define in $L(M)$ a finite set F which is not decidable in the model M of T_{Π_2}. There are at least two definitions of finiteness. One is that a set is finite if it is bounded by some natural number (possibly a nonstandard one). A second is that a set is finite if it can be put in one-to-one

[4] A function h is linearly honest if its runtime is bounded by some constant multiple of its value. The reader may assume without loss of generality that the measure Φ used here behaves like Turing machine time. The restriction requiring $h(x) > k_e * x^3$ is added for technical convenience in the proof and could be eliminated. The notion of immunity used here has some similarities to those used in busy beaver constructions [4], and to those used by Leivant [22], and Young [36].

correspondence with a natural number. Normally, these definitions are equivalent. However, this is not the case in T_{Π_2}. In fact, within the model M the set F can be put in one-to-one correspondence with the set $\bar{S}$ which is cofinal in the model. Our set F is finite only in the sense that it is bounded by a_1.

The definition of the finite set F is strongly tied to the decision procedure for the simple set S. We use the following program to enumerate the elements of F:

2.2. DEFINITION.
Program P:

```
BEGIN
     SET $y = 0$ and $x = -1$;
     REPEAT
          SET $x = x + 1$;
          IF $x \in \bar{S}$ THEN
               (PRINT $y$ and SET $y = y + 1$);
     UNTIL false
END
```

From this program we define F to be $\{y \mid P \text{ prints } y\}$. For the proof to work it is necessary to show that P is a program in M and that the run-time for each iteration of the program is in M if the value of x is in M.

Because M is a model of all true Π_2 sentences, we can show that in M bounded sets with no maximal element cannot be recursive (a limited form of the Overspill Principle). Thus F cannot be recursive. (Similarly, if we change the above program P to P' by replacing the test"$x \in \bar{S}$" by "$x \in \bar{S}$ & $x > a_0$" then P' prints N, and N is undecidable for the same reason.)

In addition to being "pathological" because it is a bounded set which is not decidable, F is also "pathological" because, although F is undecidable, the program P enumerates F in increasing order in M.

The program P can also be used to illustrate that standard techniques for proving program termination do no always work in models of T_{Π_2}. For a decision procedure d, let $P(d, z)$ be the following program:

Program $P(d, z)$:

```
BEGIN
     SET $y = 0$ and $x = -1$;
     REPEAT
          SET $x = x + 1$;
          IF $d(x)$ returns "true" THEN
               (PRINT $y$ and SET $y = y + 1$);
     UNTIL $y > z$.
END
```

In any model of PA in which d has the following properties,

(i) d is a decision procedure which always terminates,

(ii) $(\forall x)[d(x)$ returns "true" $\Rightarrow (\exists y > x)[d(y)$ returns "true"]],

(iii) $(\exists x)$ $[d(x)$ returns "true"],

$P(d, z)$ terminates. However this need not be the case for T_{Π_2}, since there are decision procedures for which the above program fails to halt in M. (It might be noted, however, that in these cases the runtimes are in $N_1 - M$, and so the programs halt in N_1 although they do not halt in M.)

2.3. Corollary. [A] *Suppose that C is any class of decision procedures then,*

$PA \vdash$ "*If* $d \in C$ *and* d *satisfies* (i), (ii) *and* (iii) *then for all integers* $z, P(d, z)$ *terminates,*"

[B] *There exists a nonempty class C of decision procedures such that,*

$T_{\Pi_2} \nvdash$ "*If* $d \in C$ *and* d *satisfies* (i), (ii) *and* (iii) *then for all integers* $z, P(d, z)$ *terminates.*"

Proof [A]: Since the decision procedure d is assumed to be total and for larger and larger inputs x returns the value "true" the loop index y is continually incremented. Hence eventually y becomes greater than z and the loop terminates. This argument can be formalized in PA which completes the proof.

[B]: Consider the model M constructed earlier and suppose that $z = a_1$ and d is a decision procedure for $\bar{S}$ such as the decision procedure which was outlined. There are fewer than a_1 elements in $\bar{S}$ so despite the fact that the loop index y is continually incremented it never becomes greater than a_1 and hence the loop does not terminate in M. So if we take C to be any syntactically (i.e., arithmetically) definable class of decision procedures which includes this decision procedure for $\bar{S}$ then [B] of the corollary follows. The fact that such a class can be defined within the language for PA is verified at the end of the proof of parts [A] and [C] of Theorem 3.1. ▮

3. Proof of the Main Theorem

3.1. Theorem. *Let S be a recursively enmerable set. If S is still-another-simple-set then there exists a model, M, of T_{Π_2} such that:*

[A] *In M, S is decidable in linear-time;*

[B] *In M, there exists a finite (bounded) set F which is definable in $L(M)$ but is undecidable in M. Furthermore, in M, F is recursively enumerable in increasing order and can be put in effective one-to-one correspondence with $\bar{S}$.*

[C] *Finally, S behaves nicely with respect to membership if the domain of the model is restricted to standard elements. That is, if $a \in N$ then $M \vDash a \in S$ if and only if $N \vDash a \in S$.*

Proof of [A] *and* [C]. Let S be a fixed still-another-simple-set and let e be an index (program) such that the domain of $\varphi_e = S$ and φ_e satisfies Definition 2.1. Since S is recursively enumerable there is Σ_1 formula such that $x \in S$ iff $(\exists y)\, s(x,y)$. We begin by constructing a model of T_{Π_2} in which S is easily decidable.

One can construct nonstandard models of T_{Π_2} from the standard model by adding new nonstandard constants and closing the structure under the total recursive functions. This technique has been refined by Hirschefeld and Wheeler [17] to construct nonstandard models of T_{Π_2} which have a variety of interesting properties. Similarly, any structure closed under all of the total recursive functions is a model of T_{Π_2}. This fact and techniques similar to those of Hirschfeld and Wheeler have been used by Kirby and Paris [21], by Lipton [23], and by Leivant [22], and we use them here.

To the theory and language of N we add new constant symbols to produce a nonstandard model of Th(N)[5], N_1, that contains a (nonstandard) total recursive function φ_g which majorizes all of the standard total recursive functions. Then a restriction, M, of N_1 is shown to be a model of T_{Π_2} and S is shown to be easily decidable in M.

Let g, a_0, a_1, a_2,... be constant symbols not in $L(N)$. In our construction g will be an index (program) and φ_g will denote the function which it computes. For each $n \geqq 1$, let Γ_n be the following collection of formulas:

Axioms which guarantee that φ_g is an honest, strictly monotone total recursive function which majorizes all of the standard total recursive functions and has a quickly testable predicate:

(1) The predicate which says that "$\varphi_g(x) = y$" can be checked in linear-time.

(2) $\varphi_g(x+1) > \varphi_g(x)$.

(3) φ_g is relatively honest. That is, $\Phi_g(x) \leqq k * |\varphi_g(x)|$ for some integer k.

(4) φ_g majorizes the first n total recursive functions. That is, let $\tau_0, \tau_1,\ldots$

be a (noneffective) indexing of the true Π_2 sentences of PA and define,

$$t_i(x) = (\min y)[\sigma_i(x,y)], \qquad \text{where} \quad \tau_i = \forall x\, \exists y \sigma_i(x,y).$$

Then, $\&_{i=0}^{n} (\forall x)\, \varphi_g(x) \geqq t_i(x)$.

Axioms which relate φ_g to the a_i's and which guarantee that the a_i's are in $\bar{S}$:

(5) $g \leqq a_0$ (this forces the program g to be relatively small)

(6) $\&_{i=0}^{n-1}\, a_{i+1} = \varphi_g(a_i)$,

(7) $\&_{i=0}^{n}\, a_i \in \bar{S}$,

(8) $\&_{i=0}^{n-1}\, (\forall x)[a_i < x < a_{i+1} \Rightarrow x \in S]$,

(9) $\&_{i=0}^{n-1}\, (\forall x)[x < a_i \,\&\, x \in S \Rightarrow \Phi_e(x) < a_{i+1}]$.

Finally, we define $\Gamma =_{\text{def}} \bigcup_n \Gamma_n$.

[5] Th(N) is the theory of the standard model N; i.e., Th(N) is the set of all sentences true in the standard model.

3.2. Lemma. *There exists a nonstandard model, N_1, of* $\mathrm{Th}(N) + \Gamma$.

Proof. By the compactness theorem it is sufficient to show that for any finite list,

$$t_0, t_1, \ldots, t_n$$

of the total recursive functions defined above and any finite collection of axioms Γ_n there is, in the standard model N, a total recursive function φ_g which satisfies Γ_n. Given any finite list of total recursive functions, there is certainly an honest, strictly monotone total recursive function which majorizes all the functions on the list. By making φ_g sufficiently large we can assume that the predicate "$\varphi_g(x) = y$" is linearly testable. Thus, given n, let φ_g be such an honest function which majorizes $t_0, t_1, \ldots, t_n$. Clearly φ_g satisfies (1)–(4) above. (Since Γ will force φ_g to majorize every standard total recursive function, any model satisfying (1)–(4) will be nonstandard.) By the definition of still-another-simple-sets, for this same φ_g there are constants $a_0, a_1, \ldots, a_n$ satisfying (5)–(9) above. ∎

We now restrict the domain of the model N_1 and show that this restriction forms a model of T_{Π_2}. Let $M = \{x \mid N_1 \models x < a_i \text{ for some } i\}$. Note that M is closed under $+$ and $*$ in the sense of N_1, (this follows from the fact that $x * x \leqq \varphi_g(x)$) and note that axiom schema 5 guarantees that g is in M. It is easy to verify that our construction of M forces all of axiom schemas (1)–(9) to hold in the model M. In addition, we claim that M with this interpretation of $+$ and $*$ is a model of T_{Π_2}:

3.3. Lemma. *If $\tau \varepsilon \Pi_2$ and $N \models \tau$ then $M \models \tau$. (Therefore $M \models T_{\Pi_2}$.)*

Proof. If $\tau \in \Pi_2$ then $\tau = \forall x \exists y \sigma(x, y)$ where σ is a bounded formula. There is a total recursive function t such that $(\forall x)\, \sigma(x, t(x))$. But in N_1, φ_g majorizes t, and since M is closed under φ_g the sentence must be true in M also. ∎

The following lemmas show that the set S is nicely behaved with respect to membership as the domain of the model is restricted or extended.

3.4. Lemma. *If $x \in N$ then $M \models x \in S$ iff $N \models x \in S$.*

Proof. This lemma simply follows from the fact that S is recursively enumerable: Note that $x \in S$ iff $(\exists y)\, s(x, y)$, where $s(x, y)$ is the quantifier free predicate introduced earlier. So for each $n \in N$ either,

$$(\exists y)\, s(n, y) \qquad \text{or} \qquad (\forall y) \neg s(n, y)$$

is true. Since $M \models T_{\Pi_2}$, whichever sentence is true in N is true in M. Therefore the set S behaves with respect to membership the same way in M as it does in N. ∎

3.5. Lemma. *If $x \in M$ then $M \models x \in S$ iff $N_1 \models x \in S$.*

Proof. If $M \models x \in S$ then $N_1 \models x \in S$ since $x \in S$ is a Σ_1 sentence and Σ_1 sentences are preserved under extensions. To show the converse, suppose that $x \in M$

and $N_1 \models x \in S$. Since $x \in M$ there is an i such that $x < a_i$. Axiom 9 tells us that $\Phi_e(x) < a_{i+1}$ and since $a_{i+1} \in M$, $M \models x \in S$. ▮

3.6. Lemma. *In M, there is a linear-time decision procedure for S.*

Proof. First observe that by Axiom 9, if $x < a_0$ then $x \in S$ iff $\Phi_e(x) < a_1$. On the other and, if $x \geqq a_0$, then $x \in S$ iff x is not equal to any a_i. Thus in the model M, we can test for membership in S as follows:

If $x < a_0$, then $x \in S$ iff $\Phi_e(x) < a_1$.

Otherwise, compute a_0, $\varphi_g(a_0)$, $\varphi_g^2(a_0)$,... until either an i is found such that $\varphi_g^i(a_0) = x$ or $\Phi_g(\varphi_g^i(a_0)) > k * |x|$, where k is the constant which determines the (linear) honesty of φ_g. The number of iterations of the last computation is certainly bounded by a_0 so that the total difficulty of the above computation is clearly bounded by $\max\{a_1, a_0 * k * |x|\}$ which *is* linear *in the model M*. ▮

A more careful analysis observes that because φ_g grows so very rapidly, $|a_i| < \frac{1}{2}$ $|a_{i+1}|$ for all i, so that $\max\{a_1, 2k * |x|\}$ is a better bound. Although we forego the proof here, the decision procedure for those $x < a_0$ can be stored in a table which is finite in M and which is given by a program whose index is less than a_1. Doing this will bring the bound down to $2k * |x|$. (Note that k is a standard integer.)

The proof of Corollary 2.3 [B] required that we be able to syntactically define (i.e., define within the language of *PA*) a class of decision procedures which includes the decision procedure outlined above, but modified to decide $\bar{S}$. The definition of the class is dependent on four variables e, a, g and k. We will say that $d \in C$ if and only if $\exists e, a, g, k$ such that d is a well-formed program which consists only of statements which perform the following:

```
Given $x$, $d$ tests whether $x < a$.
  If $x < a$ then $d$ tests whether $\Phi_e(x) < \varphi_g(a)$.
    If $\Phi_e(x) < \varphi_g(a)$ then $d$ returns "false."
    If $\Phi_e(x) \geqq \varphi_g(a)$ then $d$ returns "true."
  If $x \geqq a$ then
    $d$ computes $\varphi_g^i(a)$
    until $\varphi_g^i(a) = x$ or $\Phi_g(\varphi_g^i(a)) > k * |x|$.
      If an $i$ is found such that $\varphi_g^i(a) = x$ then $d$ returns "true,"
      If an $i$ is found such that $\Phi_g(\varphi_g^i(a)) > k * |x|$ then $d$ returns "false."
```

Clearly, the above description yields an arithmetic definition of a class C of programs which includes a decision procedure for $\bar{S}$.

The Existence of Undecidable Finite Sets, Proof of Theorem 3.1[B]

The bounded set F and the program P which prints (and defines) F within M were defined in Definition 2.2. To analyze P's behavior and how P defines F in the model

M, it is convenient to change P to P', a program with domain F which enumerates $\bar{S}$ in increasing order:

Program $P'(z)$:

```
BEGIN
    SET $y=0$ and $x=-1$;
    REPEAT
        SET $x=x+1$;
        IF $x \in \bar{S}$ THEN SET $y=y+1$;
    UNTIL $y>z$;
    PRINT $x$.
END
```

It is clear that P' computes a monotonic function whose range is some initial segment of $\bar{S}$ in M. To see that the range of P' includes all of $\bar{S}$ in the model M, we note that to write an element c of $\bar{S}$, P' requires an input z which will certainly be less than c. $P'(z)$ will have to test all elements $x \leqq c$ for membership in $\bar{S}$. However, this latter membership test can be made linear in the model M, so in M, $P'(z)$ will require $O(c^2)$ steps. Since our axioms guarantee that φ_g majorizes x^2, c^2 is less than $\varphi_g(c)$ for all nonstandard elements c, and since the model M is closed under the function φ_g, program P operates successfully within the model M to enumerate all of $\bar{S}$. (Using the same techniques as those for deciding S, both S and $\bar{S}$ can also be enumerated in increasing order in time which is linear in the output.)

F is the domain of the function computed by P', and it remains to show that F is not decidable in the model M. (The program P' can easily be modified so that our proof shows that N is also undecidable.) We begin by observing that in M, F clearly has fewer than $2a_0$ elements since it is in one-to-one correspondence with $\bar{S}$. Hence F has fewer than a_1 elements and is therefore bounded by a_1. Since $\bar{S}$ is cofinal in M, it is clear that neither F (nor N) can have a largest element, so we may complete the proof by showing that in M no bounded set without a largest element can be recursive.

To show this, suppose that F is any nonempty set which in M is recursive, bounded, and has no largest element. Let a be any (nonstandard) bound on F. Then the following set, F', will also be recursive,

$$F' = \{x \mid (\exists y < a)[x < y \,\&\, y \in F]\},$$

as is the complement T of F'. But if T is recursive, or even recursively enumarable, it is defined by a formula of the form $(\exists y)\, \sigma(x, y, \mathbf{c})$, where $\mathbf{c}$ is a (vector of) (possibly) nonstandard elements of M and σ is a bounded formula. But then the formula

$$(\forall \mathbf{w})(\forall x)[(\exists y)\, \sigma(x, y, \mathbf{w}) \Rightarrow (\exists y)[\sigma(x, y, \mathbf{w}) \& (\forall z < y) \neg \sigma(x, z, \mathbf{w})]]$$

is certainly true in N and since it is a Π_2 formula it holds in any model of T_{Π_2}. In particular this formula holds for the nonstandard model M, and so the result now follows by substituting the (nonstandard) constant(s) **c** for the variables **w**, since the last formula then asserts that T has a smallest element, say b, forcing $b-1$ to be a largest element both of F' and of F. (For further details, see [17, pp. 142–146].) Therefore, neither F nor N can be recursive in the model M.

We conclude by showing that still-another-simple-sets exist.

4. Construction of Still Another Simple Set

4.1. Theorem. *There exists a still-another-simple-set.*

4.2. Definition. Let T be an infinite set. A function f is said to *majorize* T if $(\forall n)[f(n) \geqq t_n]$, where $t_0, t_1, \ldots$ are the members of T in strictly increasing order.

4.3. Definition. A set T is *hyperimmune* if and only if T is not majorized by any total recursive function. A set S is *hypersimple* if S is recursively enumerable and $\bar{S}$ is hyperimmune. (It should be clear from the definition that hypersimple sets are undecidable. See [32] for further discussion.)

The notation which we use will be standard:

$\varphi_0, \varphi_1, \varphi_2, \ldots$: a provably acceptable indexing of the partial recursive functions,

$W_0, W_1, W_2, \ldots$: the domains of these functions,

$\Phi_0, \Phi_1, \Phi_2, \ldots$: a "nice", provable Blum measure for $\varphi_0, \varphi_1, \ldots$ (We assume throughout that this measure behaves like Turing machine time.)

$\langle\, ,\rangle$: a primitive recursive pairing function. That is, $\langle\, ,\rangle$ maps N^2 one-to-one and onto N and is strictly monotone in each of its arguments.

π_1, π_2: primary projection functions for $\langle\, ,\rangle$. (See [24] for details concerning $\langle\, ,\rangle$ and π_1 and π_2.)

Our construction uses a movable markers argument. At stage n of the construction each marker, Δ_k $(k \leqq n)$, will be positioned next to a unique integer p_k. As long as Δ_k remains active we will be looking for a "$\pi_2(k)$-sequence" for $\varphi_{\pi_1(k)}$ which can be made to satisfy (1)–(4) of Definition 2.1. If such a sequence, or the initial portion of such a sequence, can be found it is placed into a "*Protect Set*" for k. At any given stage of the construction the elements of S consist of exactly those elements which are not in a *Protect Set* but are bounded above by an element of some *Protect Set.*

Stage 0:

Place Δ_0 next to 0 making $p_0 = 0$;

Place p_0 into $\mathit{Protect}_0$;

Deactivate Δ_0;

This completes stage 0.

Stage s ($s > 0$): At the beginning of Stage s we have markers $\Delta_0, \Delta_1, \ldots, \Delta_{s-1}$ which are positioned beside integers $p_0, p_1, \ldots, p_{s-1}$, respectively.

```
For $k = 0$ to $s - 1$ do
    If $\Delta_k$ is active Then
        If $p_k \notin Protect_k$ Then
            Place $p_k$ into $Protect_k$;
            If $\pi_2(k) = 0$ Then Deactivate $\Delta_k$;
            If $x < p_k$ and $x$ has not been placed into $S$ and $x \notin \bigcup_{j<k} Protect_j$
                Then place $x$ into $S$.
        Set $z$ = max element of $Protect_k$;
        Compute $\varphi_{\pi_1(k)}(z)$ for $s$ steps;
        [That is, until $\varphi_{\pi_1(k)}(z)\downarrow$ or $\Phi_{\pi_1(k)}(z) = s$ whichever happens first.]
        If $\varphi_{\pi_1(k)}(z)\downarrow$ and $\varphi_{\pi_1(k)}$ is honest and monotonically increasing for $z$[6] Then
            Place $\varphi_{\pi_1(k)}(z)$ into $Protect_k$;
            If $|Protect_k| = \pi_2(k)$ Then Deactivate $\Delta_k$;
            If $z < x < \varphi_{\pi_1(k)}(z)$ Then Place $x$ into $S$;
            If $\exists j_0 > k$ such that $p_{j_0} \leqq \varphi_{\pi_1(k)}(z)$ Then
                [$\forall j \geqq j_0$, $\Delta_j$ must be moved and $Protect_j$ emptied]
                For $j = j_0$ to $s - 1$ do
                    [Reactivate $\Delta_j$ as follows]
                    Place $\Delta_j$ next to $\varphi_{\pi_1(k)}(z) + j$ making $p_j = \varphi_{\pi_1(k)}(z) + j$;
                    Empty $Protect_j$.
Activate $\Delta_s$ by placing $\Delta_s$ next to $p_{s-1} + 1$.
This completes stage $s$.
```

The next lemmas show that S is in fact provably undecidable.

4.4. Lemma. *S is provably still-another-simple-set.*

Proof. We begin by proving that each marker is moved only finitely often. A marker can be forced to move only by a marker of lower index. Hence it suffices to show that each marker forces only finitely many moves of other markers. Let k be a fixed integer and suppose that we have reached a stage, s, at which no marker with a lower index than k ever forces another marker to move. Suppose that $k = \langle i,j \rangle$. Then Δ_k can only force another marker to move if there is a sequence

$$p_k, \varphi_i(p_k), \ldots, \varphi_i^n(p_k) \qquad \text{with} \quad n \leqq j$$

such that φ_i is honest and monotonically increasing for the sequence. At each stage in the construction when a new element in the sequence is discovered, Δ_k may force

[6] We say that $\varphi_{\pi_1(k)}$ is honest and monotonically increasing for z if,

(i) $z < \varphi_{\pi_1(k)}(z)$,

(ii) $\Phi_{\pi_1(k)}(z) \leqq c * |\varphi_{\pi_1(k)}(z)|$

for the constant c which indicates linear honesty in the complexity measure.

finitely many markers with higher index to move. However, if $\varphi_i^j(p_k)\downarrow$, then Δ_k is deactivated and never again forces other markers to be moved. On the other hand, if $\varphi_i^j(p_k)$ is not defined then there is some greatest $n < j$ such that $\varphi_i^m(p_k)$ converges for all $m \leqq n$. In this case Δ_k will never force another marker to move after the stage at which $\varphi_i^n(p_k)$ converges. By our hypothesis the markers of lower index than Δ_k never reactivate Δ_k. Hence Δ_k only forces finitely many markers to be moved finitely many times.

To show that S has the requisite properties to be still-another-simple-set, let h be any fixed monotonically inceasing linearly honest function which grows at least as rapidly as $k_e * x^3$, where k_e is a constant which will be specified later. We need to show that for any integer n there exist elements $t_0, t_1, \ldots, t_n$ in $\bar{S}$ such that for each $i < n$:

(1) $t_0 > n$,

(2) $t_{i+1} = h(t_i)$,

(3) $(\forall x)[t_i < x < t_{i+1} \Rightarrow x \in S]$,

(4) $(\forall x)[x < t_i \;\&\; x \in S \Rightarrow \Phi_e(x) < t_{i+1}]$.

Let $\varphi_d = h$ and let s be a stage in the construction when all the markers having lower index than $\Delta_{\langle d,n\rangle}$ have come to rest. (Since these markers have come to rest they never again force $\Delta_{\langle d,n\rangle}$ to move.) Since h is total, honest and monotonically increasing a sequence,

$$p_{\langle d,n\rangle}, h(p_{\langle d,n\rangle}), \ldots, h^n(p_{\langle d,n\rangle})$$

will eventually be produced and placed into $Protect_{\langle d,n\rangle}$. The sequence will never be removed and placed into S, since by our assumption all markers of lower index have already come to rest. Because of the manner in which markers are introduced, $p_{\langle d,n\rangle}$ is greater than n. At the time when elements of the above sequence were placed into $Protect_{\langle d,n\rangle}$, the intervening integers are placed into S. Hence, if we take t_0 to be $p_{\langle d,n\rangle}$ and t_i to be $h_i(t_0)$, then (1)–(3) are satsfied. Now let e be the index of the procedure described above. Then the domain of $\varphi_e = S$. Examining the procedure one sees that it takes $O(t_i^3)$ steps to place t_i into $Protect_{\langle d,n\rangle}$. At this point all elements less than t_i which will ever be place into S have been placed there. Therefore if for all $x\, h(x) > k_e * x^3$, where k_e is a constant depending on e, then $\Phi_e(x) < t_{i+1}$ for all $x < t_i$ which are members of S. This establishes (4). Since the above proof can be formalized in PA, S is provably still-another-simple-set. ▮

4.5. LEMMA. *If S is still-another-simple-set, then $\bar{S}$ is hyperimmune.*

Proof. Suppose not. Then there exists a total recursive function t such that t majorizes $\bar{S}$. Given t we can construct an honest monotonically increasing function h which majorizes t. h must also majorize $\bar{S}$ so for any i there are at least i elements of $\bar{S}$ less than or equal to $h(i)$. For any x_0 there are at most $x_0 + 1$ elements of $\bar{S}$ less than or equal to x_0. Suppose that x_0 is the beginning of an h-sequence in $\bar{S}$. Then there are at most $x_0 + 2$ elements of $\bar{S}$ less than or equal to $h(x_0)$ and there are at most $x_0 + 3$ elements of $\bar{S}$ less than or equal to $h^2(x_0)$. But if h majorizes $\bar{S}$ there must be at least $h(x_0)$ elements of $\bar{S}$ less than or equal to $h(h(x_0))$. For any sufficiently large h this is not possible. Thus $\bar{S}$ can not be majorized by any total recursive function and is therefore hyperimmune. ▮

4.6. COROLLARY. *$\bar{S}$ is provably hypersimple.*

Proof. Follows directly from Lemmas 4.4 and 4.5 and the fact that Lemma 4.5 can be proved in *PA*. ▮

5. CONCLUDING REMARKS

It is worth pointing out that when Lipton and Leivant construct the nonstandard models in which they obtain rapidly decidable sets, the sets themselves seem to be nonstandard, that is, their definitions of the sets explicitly use nonstandard constants. Thus when passing back from the nonstandard models to the standard model, not only do the fast algorithms disappear, so to do the sets which the algorithms were to decide. To paraphrase an early criticism of nonstandard models, [2], all that remains in the standard model are ghosts, both of the fast algorithms and of the now-departed sets which the algorithms were to decide. On the other hand, since still-another-simple-sets can be described by the behavior of standard programs, that is programs which do not use nonstandard constants in their definitions, when we pass from our nonstandard model to the standard model, these programs as well as the simple sets which they describe remain.

In conclusion, it should be pointed out that Lipton's results are stronger than the results of our main theorem in the following important way: Lipton shows that for *every* (provably) recursive set it is consistent with B to believe that either the set or its complement has an infinite easy to decide subset. On the other hand, our main theorem simply gives a small class of undecidable sets for which it is consistent to believe that its members are all easily decidable. Given standard incompleteness results, it is certainly not surprising that such sets exist. It remains to be seen just how large a class of *standard* sets can be shown to be easily decidable in models of theories like B or T_{Π_2}.

REFERENCES

1. T. P. BAKER, On "provable" analogs of *P* and *NP*, *Math. Systems Theory* 12 (1979), 213–218.
2. G. BERKELEY, Letter to an infidel mathematician, *in* "Works of George Berkeley, Bishop of Cloyne" (A. A. Luce and T. E. Jessop, Eds.), Humanities, London, 1951.
3. R. L. CONSTABLE, "A Constructive Theory of Recursive Functions," Technical Report 73–186, Cornell University, 1973.
4. R. DALEY, On the simplicity of busy beaver sets, *Z. Math. Logik Grundlag. Math.* 24 (1978), 207–224.

5. R. A. DeMillo and R. J. Lipton, Some connections between mathematical logic and complexity theory, *in* "Proceedings, 11th Symposium on the Theory of Computation," pp. 153–159, 1979.
6. R. A. DeMillo and R. J. Lipton, The consistency of "$P = NP$" and related problems with fragments of number theory, *in* "Proceedings, 12th Symposium on the Theory of Computing," pp. 45–57, May 1980.
7. J. C. E. Dekker and J. Myhill, Retractable sets, *Canad. J. Math.* **10** (1958), 357–373.
8. P. C. Fischer, Theory of provably recursive functions, *Trans. Amer. Math. Soc.* **117**(5) (1965), 494–520.
9. S. Fortune, "Topics in Computational Complexity," Ph.D. dissertation, Cornell University, 1979.
10. D. C. Goldrei, A. Macintyre, and H. Simmons, The forcing companions of number theories, *Israel J. Math.* **14** (1973), 317–337.
11. D. Gordon, Complexity classes of provably recursive functions, *J. Comput. System Sci.* **18**(3) (1979), 294–303.
12. P. Hajek, Arithmetical complexity of some problems in computer science, *in* "Mathematical Foundations of Computer Science, (J. Gruska, Ed.), pp. 282–287, Springer-Verlag, Berlin/New York, 1977.
13. P. Hajek, Arithmetical hierarchy and complexity of computation, *Theoret. Comput. Sci.* **8** (1979), 227–237.
14. J. Hartmanis and J. E. Hopcroft, Independence results in computer science, *SIGACT News* **8**(4) (1976), 13–24.
15. J. Hartmanis, Relation between diagonalization, proof systems and complexity gaps, *in* "Proceedings, 9th Symposium on the Theory of Computing," pp. 223–227, 1977.
16. A. Heyting, Some remarks on intuitionism, in Constructivity in Mathematics" (A. Heyting, Ed.), pp. 69–71, North-Holland, Amsterdam, 1959.
17. J. Hirschfeld and W. H. Wheeler, "Forcing, Arithmetic, and Division Rings," Lecture Notes in Mathematics No. 454, Springer-Verlag, Berlin/New York, 1975.
18. S. Homer and J. H. Reif, The complexity of provable properties of first order theories, preprint, 1980.
19. D. Joseph, "On the Power of Formal Systems for Analyzing Linear and Polynomial Time Program Behavior," Ph.D. dissertation, Purdue University, 1981.
20. D. Joseph and P. Young, "Fast Programs for Initial Segments and Polynomial Time Computation in Weak Models of Arithmetic," preliminary version in "Proceedings, 13th Symposium on The Theory of Computing," pp. 55–61, May 1981.
21. L. Kirby and J. Paris, Initial segments of models of Peano's axioms, *in* "Proceedings, Conference Set Theory and Hierarchy Theory V; Bieutowice, Poland," Lecture Notes in Mathematics No. 619, Springer-Verlag, Berlin/New York, 1976.
22. D. Leivant, On easily infinite sets and on a statement of R. Lipton, preprint, 1979.
23. R. J. Lipton, Model theoretic aspects of complexity theory, *in* "Proceedings, 19th Symposium of the Foundations of Computer Science," pp. 193–200, 1978.
24. M. Machtey and P. Young, "An Introduction to the General Theory of Algorithms," Elsevier North-Holland, New York, 1978.
25. A. Macintyre and H. Simmons, Algebraic properties of number theories, *Israel J. Math.* **22** (1975), 7–27.
26. K. McAloon (Ed.), "Modèles de l'Arithmètique," Socièté Mathèmatique de France, Paris, 1980.
27. M. O'Donnell, A programming language theorem which is independent of Peano Arithmetic *in* "Proceedings, 11th Symposium on the Theory of Computation," pp. 176–188, 1979.
28. J. B. Paris, Some independence results for Peano Arithmetic, *J. Symbolic Logic* **43** (1978), 725–730.
29. J. B. Paris and L. Harrington, A mathematical imcompleteness in Peano arithmetic, *in* "Handbook of Mathematical Logic" (J. Barwise, Ed.), North-Holland, New York, 1978.
30. N. C. K. Phillips, The $P = NP$ question and recent independence results, *Quaestiones Inform.* **1** (1979), 26–29.

31. A. Robinson, Nonstandard arithmetic and generic arithmetic, *in* "Proceedings, 1971 International Congress Logic, Methodology, Philosophy, Science," 1971.
32. H. Rogers, Jr., "Theory of Recursive Functions and Effective Computability," McGraw–Hill, New York, 1967.
33. H. Simmons, Existentially closed models of basic number theory, *in* "Logic Colloquium 76" (R. Gandy and J. M. R. Hyland, Eds.), pp. 325–369, North-Holland, Amsterdam, 1977.
34. A. Tarski, A. Mostowki, and R.Robinson, "Undecidable Theories," North-Holland, Amsterdam, 1953.
35. P. Young, Linear orderings under one-one reducibility, *J. Symbolic Logic* **31** (1966), 70–85.
36. P. Young, Toward a theory of enumerations, *J. Assoc. Comput. Mach.* **16** (1969), 328–348.
37. P. Young, Optimization among probably equivalent programs, *J. Assoc. Comput. Mach.* **24**(4) (1977), 693–700; prelim abstract in "IEEE SWAT Proc., 1973").