JOURNAL OF COMPUTER AND SYSTEM SCIENCES: 4, 368–376 (1970)

What Makes Some Language Theory Problems Undecidable*

J. HARTMANIS AND J. E. HOPCROFT

Computer Science Department, Cornell University, Ithaca, New York 14850

Received September 3, 1969

In the theory of automata and formal languages, the undecidability of various properties has been studied for specific classes of languages. Here we abstract the essence of various proofs of undecidability and find wide classes of properties and general conditions on families of languages such that these proofs of undecidability hold. The paper also illustrates the manner in which the degree of undecidability of a property changes as we consider more and more complicated families of languages.

INTRODUCTION

In the theory of automata and formal languages, the undecidability of various properties has been studied for specific classes of languages. The purpose of this paper is to abstract the essence of why certain properties in language theory are undecidable and then give proofs of their undecidability without relying on specific peculiarities of the different families of languages.

The results of this paper show that many properties of families of languages are undecidable because of the "ability to count" or "compare" in these languages. Intuitively, if a family of languages contains the set $\{a^nb^n \mid n \geqslant 1\}$ and is closed under some simple closure properties, then it has the "ability to count"; for example, the AFL generated by $\{a^nb^n \mid n \geqslant 1\}$ is such a family of languages. To capture this intuitive concept of "counting" we define a counter machine and then relate the valid computations of a counter machine to Turing machine computations and to families of languages containing the set $\{a^nb^n \mid n \geqslant 1\}$. On the one hand, it is shown that every recursively enumerable set is a homomorphic image of the set of valid computations of some counter machine. On the other hand, it is shown that if a family of languages $\mathscr{L}$ (with some closure properties) contains the set $\{a^nb^n \mid n \geqslant 1\}$ then the valid computations of any counter machine can be represented as an intersection of two languages in $\mathscr{L}$. These results are then used to derive several undecidability results for families of languages containing the set $\{a^nb^n \mid n \geqslant 1\}$. For example, for any recursive AFL containing $\{a^nb^n \mid n \geqslant 1\}$ the problems of containment, equivalence and equivalence to

* This research has been supported in part by National Science Foundation Grant GJ-155 and GJ-96.

Σ^* are undecidable and of Turing degree 1. To illustrate the manner in which the degree of undecidability of a property changes, as we consider more complicated families of languages, we impose additional closure properties on families of languages and establish the degree of unsolvability for various properties. This approach reveals a well defined structuring of the undecidable problems and permits a systematic study of these problems and their relation to various families of automata.

The paper concludes with the conjecture that if two different AFL's both contain the set $\{a^n b^n \mid n \geqslant 1\}$ then the problem of deciding whether a language from one of the AFL's is in the other is undecidable of Turing degree 2. The conjecture is supported by an illustrative proof of a special case of this conjectured result and the knowledge that such proofs can be given for most common AFL's. Nevertheless, no general proof exists.

Preliminaries

In this section we make precise the concepts used in this paper and derive preliminary results. We assume that the reader is familiar with the basic concepts in formal languages [1] and with the notion of Turing reducibility [2].

We shall be concerned with families of languages which possess properties common to many families of languages which arise naturally in the study of automata theory and formal languages.

Let Σ be an infinite set of symbols. A *family of languages* $\mathscr{L}$ is a nonempty collection of languages, containing at least one nonempty language, with the property that for each L in $\mathscr{L}$ there is a finite set $\Sigma_1 \subseteq \Sigma$ such that $L \subseteq \Sigma_1^*$.[1]

In [3] an abstract model of an automaton was introduced which is sufficiently general to encompass most known types of automata. It was shown that any family of languages defined by a class of automata must have certain basic properties. Namely, the class must contain all regular sets and be closed under inverse finite state transducer mappings,[2] and marked $+$.[3]

[1] For each set of strings A and B, $AB = \{xy \mid x \text{ in } A, y \text{ in } B\}$, $A^+ = \bigcup_{i=1}^{\infty} A^i$ where $A^{i+1} = A^i A$ for each $i \geqslant 1$ and $A^* = A^+ \bigcup \{\epsilon\}$ where ϵ is the empty string.

[2] A *finite state transducer* G is a 6-tuple $(K, \Sigma, \Delta, q_0, \delta, F)$ where K, Σ and Δ are finite sets of *states*, *input symbols* and *output symbols*, respectively; q_0 in K is the *initial* state and $F \subseteq K$ is the set of *final* states. The function δ maps $K \times \Sigma$ into finite subsets of $K \times \Delta^*$. The domain of δ can be extended to $K \times \Sigma^*$ as follows. For each q in K, a in Σ and x in Σ^*, let $\delta(q, \epsilon) = \{(q, \epsilon)\}$ and $\delta(q, xa) = \{(p, w) \mid w = w_1 w_2$ and for some p', (p', w_1) is in $\delta(q, x)$ and (p, w_2) is in $\delta(p', a)\}$. G is said to be *deterministic* if $\delta(q, a)$ contains a single element for each q in K and a in Σ. The mapping $G : \Sigma^* \to \Delta^*$ defined by $G(x) = \{y \mid \exists q$ in F such that (q, y) is in $\delta(q_0, x)\}$ is called a finite state transducer mapping. $G^{-1}(y) = \{x \mid y \text{ is in } G(x)\}$ is called the *inverse finite state transducer mapping*.

[3] Let Σ_1 be a finite set of symbols and let a be a symbol not in Σ_1. Then for $L_1 \subseteq \Sigma_1^*$, $(aL_1)^+$ is the marked $+$ of L_1.

Since many families of languages are defined by automata we shall be especially concerned with families which possess these properties.

Families of languages defined by one-way nondeterministic automata have been characterized by their closure properties [4]. These families, called abstract families of languages, have additional structure in which we are also interested. An *abstract family of languages* (abbreviated AFL) is a family of languages closed under the operations of $\bigcup$, $\cdot$, $+$, ϵ–free homomorphism, inverse homomorphism, and intersection with regular sets. We note in passing that if an AFL $\mathscr{L}$ contains a language L such that ϵ is in L, then $\mathscr{L}$ contains all regular sets.

To give precise meaning to the idea of "counting" we now introduce the concept of a *counter machine*. Intuitively, the counter machine is a one register machine (the register is capable of holding an arbitrary integer) which can multiply the content of its register by one of a finite number of multiplicands, branching if the resulting product is not an integer. More precisely, let K be a finite set of states with two distinguished elements q_0 and q_f, $q_0 \neq q_f$, called the *initial* and *final* states. Let $C = \{2, 3, 5, 7, 1/2, 1/3, 1/5, 1/7\}$. A *quadruple* is an element of $K \times C \times K \times K$. A *counter machine* M is a set of quadruples such that no two quadruples have the same first component and no quadruple of M has q_f as the first component nor q_0 as the last component.

Let a be a new symbol. Let a^n, n an integer, be the string consisting of n a's. A *configuration* of M is a string of the form qa^i, q in K. For each integer n, we write $qa^n \vdash pa^{kn}$ if (q, k, p, r) is a quadruple of M and kn is an integer. We write $qa^n \vdash ra^n$ if (q, k, p, r) is a quadruple of M and kn is not an integer.

Fact 1. It is known that if $L \subseteq \{1, 2\}^*$ is the set of strings accepted by a Turing machine, then a counter machine M can be effectively constructed such that $q_0a^{2^i} \overset{*}{\vdash} q_fa^j$ if and only if $i = x_1 + 3x_2 + 3^2x_3 + \cdots + 3^{n-1}x_n$ and $x_1x_2 \cdots x_n$ is in L.[4] This result, in a slightly different formulation, is due to M. Minsky [5] and an exposition of it can be found in Chapter 6 [1].

Let $\alpha_0 = q_0a^{2^i}$, α_1, α_2, ..., α_{n-1}, $\alpha_n \in q_fa^*$ be configurations of a counter machine M where $\alpha_j \vdash \alpha_{j+1}$, $0 \leqslant j < n$. Then $q_0a\, q_0a^2\, q_0a^4 \cdots q_0a^{2^{i-1}}\, \alpha_0\alpha_1 \cdots \alpha_n$ is said to be a *valid computation* of M. Let L_M denote the set of all valid computations of M and let $\bar{L}_M$ denote the set of all invalid computations of M (i.e., $\bar{L}_M$ is the complement of L_M with respect to some Σ^* containing L_M, Σ finite).

Fact 2. (a) It is undecidable, and of Turing degree 1,[5] to determine if a Turing machine accepts the empty set and hence if L_M is empty ($\bar{L}_M = \Sigma^*$). (b) It is undecidable, and of Turing degree 2, to determine if a Turing machine accepts infinitely many strings and hence if L_M is infinite ($\bar{L}_M$ not cofinite).

[4] The empty string corresponds to $i = 0$.

[5] Degree 1 is the degree of the halting problem. Degree 2 is the jump of degree 1.

LEMMA. *Let $L \subseteq \{1, 2\}^*$ be a recursively enumerable set. Then there is an effective procedure to obtain a counter machine M such that L is a homomorphic image of L_M.*

Proof. Since L is recursively enumerable,

$$L^R = \{a_1 a_2 \cdots a_n \mid a_n a_{n-1} \cdots a_1 \text{ is in } L\}$$

is recursively enumerable. Let M (by Fact 1) be a counter machine with states K, initial state q_0 and final state q_f such that $q_0 a^{2^i} \overset{*}{\vdash} q_f a^j$ if and only if

$$i = x_1 + x_2 3 + x_3 3^2 + \cdots + x_n 3^{n-1}$$

and $x_1 x_2 \cdots x_n$ is in L^R.

Our task is now to construct another counter machine $\hat{M}$ such that if $i = x_1 x_2 \cdots x_n$ is in L we can retrieve $x_1 x_2 \cdots x_n$ by a homomorphism from the corresponding valid computation $q_0 a^{2^i} \overset{*}{\vdash} q_f a^j$. To do this we add new states and quadruples to M which serve to first convert $a^{2^{x_1 + 3x_2 + \cdots + 3^{n-1}x_n}}$ to $a^{2^{x_n + 3x_{n-1} + \cdots + 3^{n-1}x_1}}$ and in so doing, the sequence of states in this part of the valid computation is such that one of two special states (8 or 9) appears each time an x_i is determined to be a 1 or 2, respectively. Then the homomorphism h which maps these two states onto 1 and 2, respectively, and every other symbol on the null string, yields $x_1 x_2 \cdots x_n$, as desired. The details of the construction follow:

$\hat{M}$ has states $K \cup \{1, 2, \ldots, 18\}$, start state 1 and final state q_f. $\hat{M}$ has all quadruples of M plus the quadruples[6] below.

(1, 1/2, 2, 3)	(7, 1/3, 4, —)	(13, 5, 14, —)
(2, 3, 1, —)	(8, 5, 11, —)	(14, 5, 11, —)
(3, 1/3, 4, q_0)	(9, 5, 10, —)	(15, 1/5, 16, 17)
(4, 1/3, 5, 8)	(10, 5, 11, —)	(16, 2, 15, —)
(5, 1/3, 6, 9)	(11, 1/2, 12, 15)	(17, 1/7, 18, 3)
(6, 7, 7, —)	(12, 5, 13, —)	(18, 3, 17, —).

States 1 and 2 convert the contents of the counter from 2^i to 3^i ($1a^{2^i} \overset{*}{\vdash} 3a^{3^i}$) causing the counter machine $\hat{M}$ to enter state 3. This is an initialization step and states 1 and 2 are never entered again. States 3 through 18 comprise a loop. One traversal of the loop transforms the counter's contents from 2 raised to the power $3^{m-1}x_1 + 3^{m-2}x_2 + \cdots + x_m$ times 3 raised to the power $x_{m+1} + 3x_{m+2} + \cdots + 3^{n-m-1}x_n$ to 2 raised to the power $3^m x_1 + 3^{m-1}x_2 + \cdots + x_{m+1}$ times 3 raised to the power $x_{m+2} + 3x_{m+3} + \cdots + 3^{n-m-2}x_n$ and determines if x_{m+1} is 1 or 2. Either state 8 or 9

[6] If the last component of a quadruple is irrelevant, it has been left blank.

is entered in the process depending on whether x_{m+1} is 1 or 2 respectively. Specifically, states 3, 4, 5, 6 and 7 cause

$$3a^{2^i3^i} \overset{*}{\vdash} \begin{cases} q_0 a^{2^i7^{\lfloor j/3 \rfloor}} & j = 0 \bmod 3^7 \\ 8a^{2^i7^{\lfloor j/3 \rfloor}} & j = 1 \bmod 3 \\ 9a^{2^i7^{\lfloor j/3 \rfloor}} & j = 2 \bmod 3. \end{cases}$$

State q_0 is the exit from the loop and is entered once all x_m have been computed. State 8 causes the count to be multiplied by 5 ($8a^{2^i7^j} \overset{*}{\vdash} 11a^{2^i5^17^j}$) and states 9 and 10 cause the count to be multiplied by 5^2 ($9a^{2^i7^j} \overset{*}{\vdash} a^{2^i5^27^j}$). Note that state 8 is entered if x_m is a 1 and state 9 is entered if x_m is a 2 as is claimed. States 11, 12, 13 and 14 convert the count from $2^i5^k7^j$ to $5^{3i+k}7^j$ and state 15 is entered. States 15 and 16 convert the count from 5^i7^j to 2^i7^j. Finally states 17 and 18 convert the count from 2^i7^j to 2^i3^j and the loop is reentered via state 3.

Finally $h(L_{\hat{M}}) = L$, as was to be shown.

Properties of Languages

In this section we shall consider properties defined on classes of languages arising in automata theory which are sub-families of the recursively enumerable sets. We shall be particularly concerned with the way the decidability of properties change as we examine more and more complicated sub-families.

Since we are primarily concerned with families of languages arising in automata theory and their usual representations, we can avoid the tedious technical difficulties associated with a complete and rigorous treatment of representations by using only the standard representations of automata theory. For example, finite automata for the regular sets, context-free grammars for the context-free languages, linearly bounded automata for context sensitive languages and so on.

Furthermore, we consider only families of recursive sets such that there is an effective procedure for enumerating the names of the sets and that there is an algorithm such that given the name of a set and a string, the algorithm will determine whether or not the string is in the set. When we say that a family of languages is closed under an operation such as union, we mean that the family is effectively closed in the sense that given names of languages L_1 and L_2, we can effectively determine the name of $L_1 \cup L_2$.

Let $\mathscr{L}_d$ be the smallest family of languages containing the set $\{a^nb^n \mid n \geqslant 1\}$, containing all regular sets and closed under inverse deterministic finite state transducer mappings and marked $+$. Let $\mathscr{L}$ be the smallest AFL containing $\{a^nb^n \mid n \geqslant 0\}$. We now show that L_M is the intersection of two sets in $\mathscr{L}_d$ and that $\bar{L}_M$ is in $\mathscr{L}$. As a corollary we obtain the result that every recursively enumerable set is the homomorphic

[7] $[j/3]$ means integer part of $j/3$.

image of the intersection of two languages from $\mathscr{L}_d$. Note that this is a stronger result than Lemma 4.2 of [6] where it was shown that every recursively enumerable set is the homomorphic image of a finite intersection of languages from $\mathscr{L}_d$.

THEOREM 1. *Let M be a counter machine with states K, start state q_0 and final state q_f. Then L_M is the intersection of two languages in $\mathscr{L}_d$.*

Proof. Let $L = \{qa^nq'a^{kn} \mid n \geqslant 1, (q, k, q', r)$ is a quadruple of M and either kn is an integer or $q = q' = q_0$ with $k = 2\} \cup \{qa^nra^n \mid n \geqslant 1, (q, k, q', r)$ is a quadruple of M and kn is not an integer$\}$. Let $L_1 = L^*(\{\epsilon\} \cup q_fa^*)$ and let $L_2 = q_0aL^*(\{\epsilon\} \cup q_fa^*)$. Clearly $L_M = L_1 \cap L_2$. Thus, we need only show that L_1 and L_2 are in $\mathscr{L}_d$.

To do this one needs only define a deterministic finite state transducer G such that $L_1 = G^{-1}[(c\{a^nb^n \mid n \geqslant 1\})^*]$. To simplify the construction we assume without loss of generality that the quadruple for q_0 has $k = 2$. Since $\mathscr{L}_d$ is closed under marked $+$ and inverse deterministic finite state transducer mappings, this shows that L_1 is in $\mathscr{L}_d$. In order to understand G, let (q, k, q^1, r) be a quadruple of M. Then G maps L_1 into $(c\{a^nb^n \mid n \geqslant 1\})^*$ by mapping $q_0a^nq_0a^{2n}$ onto $ca^{2n}b^{2n}$; qa^nra^n onto ca^nb^n, and $qa^nq^1a^{kn}$ onto $ca^{kn}b^{kn}$ if $k > 1$ and onto ca^nb^n otherwise.

The definition of G follows:

$$G = [\{p_0, p_1, p_2\} \cup K \cup (K \times \{1, 2, \ldots, 7\}) \cup (\{p\} \times \{2, 3, 5, 7\}),$$
$$\times \{a\} \cup K, \{a, b, c\}, p_0, \delta, \{p_1, p_2, p_3\}]$$

where $\delta(p_0, q) = (q, c)$; $\delta(q, a) = (q, a^k)$ and $\delta(q, q^1) = (p_1, \epsilon)$ provided (q, k, q^1, r) is a quadruple of M and $k > 1$ or $q = q^1 = q_0$ and $k = 2$; $\delta(p_1, a) = (p_1, b)$; $\delta(p_1, q) = (q, c)$; $\delta(p_1, q_f) = (p_2, \epsilon)$; $\delta(p_2, a) = (p_2, \epsilon)$; $\delta([q, a) = ([q, 1], a)$, $\delta([u, i], a) = [(q, i + 1], a)$, $1 \leqslant i \leqslant i/k$, $\delta([q, 1/k], a) = ([q, 1], a)$, $\delta([q, 1/k], q^1) = ([p, 1/k], \epsilon)$, $\delta([q, i], r) = (p_0, \epsilon)$, $1 \leqslant i < 1/k$ provided (q, k, q^1, r) is a quadruple of M and $k < 1$; and finally $\delta([p, 1/k], a) = ([p, 1/k], b^{1/k})$, $\delta([p, 1/k], q) = (q, c)$, $q = q_f$ and $([p, 1/k], q_f) = (p_2, \epsilon)$ for $k = 1/2, 1/3, 1/5$ and $1/7$.

Now $L_1 = G^{-1}((c\{a^nb^n \mid n \geqslant 1\})^*)$ and thus L_1 is in $\mathscr{L}_d$. By a similar construction L_2 is also in $\mathscr{L}_d$.

COROLLARY. *Let $L \subseteq \{1, 2\}^*$ be a recursively enumerable set. Then L is the homomorphic image of the intersection of two sets in $\mathscr{L}_d$.*

This corollary is a stronger version of a previously known result. Namely that every recursively enumerable set was a homomorphic image of the intersection of two deterministic context-free languages [7].

THEOREM 2. *$\bar{L}_M$ is in the least* AFL $\mathscr{L}$ *containing $\{a^nb^n \mid n > 0\}$.*

Proof. Let $L = \{qa^npa^j \mid$ conditions (i), (ii), (iii) hold$\}$.

(i) $q = p = q_0$ implies $j \neq 2n$.

(ii) (q, k, p, r) a quadruple of M and kn an integer implies $kn \neq j$.

(iii) (q, k, r, p) a quadruple of M implies either that kn is an integer or $j \neq n$.

Now

$$\begin{aligned}\bar{L}_M = {} & (K \cup \{a\})^* L(Ka^*)^* \\ & \cup [(\{a\} \cup K)^* - q_0 aK(\{a\} \cup K)^*] \\ & \cup [(\{a\} \cup K)^* - (\{a\} \cup K)^* q_f a^*] \\ & \cup [(\{a\} \cup K)^* KK(\{a\} \cup K)^*].\end{aligned}$$

Since $\mathscr{L}$ contains all regular sets and is closed under union and product we need only show that L is in $\mathscr{L}$. By a construction similar to that in Theorem 1 we can show that L is a transducer mapping of $\{a^nb^n \mid n \geqslant 0\}$. Thus $\bar{L}_M$ is in the least AFL $\mathscr{L}$ containing $\{a^nb^n \mid n \geqslant 1\}$.

We now show how various decision problems escalate as the underlying family of languages becomes more complicated. In what follows one may think of $\mathscr{L}_d$ as representing families similar to the deterministic context-free languages, the least AFL containing $\{a^nb^n \mid n \geqslant 0\}$ as representing families similar to the context-free languages and a family containing $\mathscr{L}_d$ and closed under intersection as representing families defined by tape bounded Turing machines.

THEOREM 3. *Let $\mathscr{L}$ be any family of recursive sets which contains the* AFL *generated by $\{a^nb^n \mid n \geqslant 1\}$. Then containment, equivalence and equivalence to Σ^* are undecidable on $\mathscr{L}$ and are of Turing degree* 1.

Proof. Equivalence is Turing reducible to containment since $L_1 = L_2$ if and only if $L_1 \subseteq L_2$ and $L_2 \subseteq L_1$. Equivalence to Σ^* is a special case of equivalence and hence Turing reducible to equivalence. By fact 2 equivalence to Σ^* is at least of degree 1 since $\bar{L}_M$ is in $\mathscr{L}$. It remains to show that containment is of at most degree 1.

But $L_1 \subseteq L_2$ is equivalent to $\forall x$ (x in $L_1 \Rightarrow x$ in L_2). Since L_1 and L_2 are recursive (x in $L_1 \Rightarrow x$ in L_2) is a recursive predicate and hence containment is at most degree 1.

Remark. In [8] it was shown that if $\mathscr{L}$ is a family of languages effectively closed under union and under concatenation by regular sets and if equivalence to Σ_1^* is undecidable, for $\mathscr{L}$ then every nontrivial property on $\mathscr{L}$ which is true for all regular sets and is preserved by inverse gsm, union with $\{\epsilon\}$, and intersection with regular sets is undecidable. Thus we note in passing that if $\mathscr{L}$ is an "effective" AFL containing $\{a^nb^n \mid n \geqslant 0\}$, then any such property is undecidable. Furthemore, if the property can be expressed as a recursive predicate preceeded by a single quantifier, then the property is precisely of Turing degree 1.

THEOREM 4. *Let $\mathscr{L}$ be a family of recursive sets which contains the* AFL *generated by* $\{a^n b^n \mid n \geqslant 1\}$. *Let $\mathscr{L}'$ be a family of languages such that*

(1) *$\mathscr{L}'$ contains all co-finite languages.*
(2) *$L \subseteq \Sigma_1^*$ in $\mathscr{L}'$ implies $\Sigma_1^* - L$ is in $\mathscr{L}'$.*
(3) *$\mathscr{L}'$ is a recursively enumerable family of recursive sets.*
(4) *There exists a recursively enumerable set which is not the homomorphic image of a set in $\mathscr{L}'$.*

Then for L in $\mathscr{L}$, it is undecidable and of Turing degree 2 to determine if L is in $\mathscr{L}'$.

Proof. Let L be in $\mathscr{L}$. Let $L_1, L_2, \ldots$ be an enumeration of $\mathscr{L}'$. The predicate L is in $\mathscr{L}'$ is equivalent to $\exists i \, \forall x \, (x \text{ in } L \Leftrightarrow x \text{ in } L_i)$. Since the predicate $(x \text{ in } L \Leftrightarrow x \text{ in } L_i)$ is recursive, the predicate L is in $\mathscr{L}'$ is at most Turing degree 2.

Now let L be a recursively enumerable set which is not the homomorphic image of any set in $\mathscr{L}'$. Let T be a Turing machine accepting L. Given an arbitrary Turing machine T_i one can effectively construct a Turing machine T_i' such that T_i' accepts x if and only if T accepts x and there exists a y, $|y| > |x|$ such that T_i accepts y. By the Lemma following Fact 2 we can effectively obtain a counter machine M_i such that the set accepted by T_i' is a homomorphic image of L_{M_i}. If T_i accepts an infinite set, then L is a homomorphic image of L_{M_i} and hence L_{M_i} is not in $\mathscr{L}'$. Since $\mathscr{L}'$ is closed under complement $\bar{L}_{M_i}$ cannot be in $\mathscr{L}'$. If T_i accepts a finite set, then $\bar{L}_{M_i}$ is cofinite and hence in $\mathscr{L}'$. Thus $\bar{L}_{M_i}$ is in $\mathscr{L}'$ if and only if T_i accepts a finite set. Since $\bar{L}_{M_i}$ is always in $\mathscr{L}$, it is at least degree 2 to determine for arbitrary L in $\mathscr{L}$ if L is in $\mathscr{L}'$.

COROLLARY. *It is of Turing degree 2 to determine if an arbitrary context-free, nondeterministic one-way stack or context-sensitive language is*

(1) *cofinite;*
(2) *regular;*
(3) *deterministic context-free;*
(4) *deterministic one-way stack language.*

THEOREM 5. *Let $\mathscr{L}$ be a family of recursive sets containing $\mathscr{L}_d$ and closed under intersection. Then the emptiness problem is of Turing degree 1 and the finiteness problem is of Turing degree 2.*

Proof. For each counter machine M, L_M is in $\mathscr{L}$ by Theorem 1. By Fact 2, emptiness is at least Turing degree 1 and finiteness at least Turing degree 2. But emptiness is equivalent to $\forall x \; (x \notin L)$ and finiteness is equivalent to $\exists n \, \forall x$ (either $|x| < n$ or x is in L).

We now consider the situation where we have two AFL's both of which contain $\{a^n b^n \mid n \geqslant 0\}$ and ask if a language in one AFL is in the other. We conjecture that

the question, if nontrivial, is of Turing degree 2. We can show this for the common AFL's such as the context-free languages, the context sensitive languages, the one-way stack languages, the least AFL containing $\{a^n b^n c^n \mid n \geqslant 0\}$ and many others but have no general proof. In each case the proof seems to hinge on a specific property of the AFL's in question. In Theorem 6 we prove the conjecture for a specific case which covers several of the above.

THEOREM 6. *Let $\mathscr{L}$ be the least* AFL *containing $\{a^{n^2} b^n \mid n \geqslant 0\}$ and $\{a^n b^n \mid n \geqslant 0\}$. Let $\hat{\mathscr{L}}$ be an* AFL *which is contained within the context-free languages, and which contains the set $\{a^n b^n \mid n \geqslant 0\}$. To determine if L is in $\hat{\mathscr{L}}$, for L in $\mathscr{L}$, is of Turing degree* 2.

Proof. Let M be a counter machine and let Σ_1 be a finite set such that $\bar{L}_M \subseteq \Sigma_1^*$. Let a be a symbol not in Σ_1^*. Let $L = \{a^i x \mid x \text{ in } \bar{L}_M \text{ or } i = |x|^2\}$. If $\bar{L}_M$ is cofinite, then

$$L = a^* \bar{L}_M \cup \{\text{finite set}\}$$

and hence is in $\hat{\mathscr{L}}$. Now assume $\bar{L}_M$ is not cofinite and that L is in $\hat{\mathscr{L}}$. Since L is in $\hat{\mathscr{L}}$, L must be context-free. Let G be a context-free grammar generating L. There exist arbitrarily long x such that $a^i x$ is in L if and only if $i = |x|^2$. Since the number of a's is the square of the length of x, for sufficiently long x there exists a variable A such that somewhere in the derivation of x, $A \overset{*}{\Rightarrow} y_1 A y_2$, $y_1 y_2$ in a^+. Thus, the number of a's can be increased without changing x resulting in a sentence not in L. Thus the assumption that L was in $\hat{\mathscr{L}}$ is false. We conclude that L is in $\hat{\mathscr{L}}$ if and only if $\bar{L}_M$ is cofinite. Hence it is of Turing degree 2 to determine if L in $\mathscr{L}$ is in $\hat{\mathscr{L}}$.

REFERENCES

1. J. HOPCROFT AND J. ULLMAN, "Formal Languages and their Relation to Automata," Addison Wesley, Reading, Mass., 1969.
2. H. ROGERS, "The Theory of Recursive Functions and Effective Computability," McGraw-Hill, New York, 1967.
3. J. HOPCROFT AND J. ULLMAN, "An Approach to a Unified Theory of Automata," *Bell. Sys. Technical J.* **46** (1967), 1793–1829.
4. S. GINSBURG AND S. GREIBACH, "Abstract Families of Languages," *Mem. Amer. Math. Soc.* **87** (1969), 1–32.
5. M. MINSKY, "Recursive Unsolvability of Post's Problem of 'Tag' and Other Topics in the Theory of Turing Machines," *Ann. of Math.* **74**(3) (1961), 437–555.
6. J. HARTMANIS AND J. HOPCROFT, "Structure of Undecidable Problems in Automata Theory," *IEEE Conference Record of Ninth Annual Symposium on Switching and Automata Theory*, New York, 1968, 327–373.
7. S. GINSBURG, S. GREIBACH, AND M. HARRISON, "One-Way Stack Automata," *J. Assn for Computing Machinery* **14**(2) (1967), 389–418.
8. S. GREIBACH, "A Note on Undecidable Properties of Formal Languages," *Mathematical Systems Theory* **2**(1) (1968), 1–6.