PRACTICAL DECIDABILITY *

by

Andrzej Ehrenfeucht
Department of Computer Science
University of Colorado
Boulder, Colorado

Report #CU-CS-008-72 Dec. 1972

* This research was supported in part by NSF Grant GJ 660.

1. INTRODUCTION AND RESULTS

1.1 Let A be a finite alphabet including the blank symbol $\square$ and A* the set of finite nonempty words over A - $\{\square\}$. For any $a = a_1 \ldots a_n \in A^*$ we put $\lambda(a) = n$. A set $X \subseteq A^*$ will be called a problem and a finite or infinite word $p = p_1 p_2 \ldots$, where $p_i \in A$ will be called a program. Let T be an ordinary one tape, erasing, printing and moving left or right Turing machine for the alphabet A with one initial state s_0 and one immobility state s_1. We shall say that T and p are adequate for X if for every word $a_1 \ldots a_n \in A^*$, whenever T in state s_0 is applied to a tape

$$\ldots \square \square a_1 a_2 \ldots a_n \square p_1 p_2 \ldots$$

with the head of T looking at a_1, then T will stop (i.e., reach s_1) if and only if $a \in X$.

We shall say that the problem X is practically decidable [of polynomial complexity] if there exist T and p adequate for X and a constant c such that for every $a \in X$ T stops after no more than $c\lambda(a)$ steps [$(\lambda(a))^c$ steps]. We shall say that X is of maximum complexity if for every T and p adequate for X there is a constant $c > 1$ and infinitely many $a \in X$ such that T visits at least $c^{\lambda(a)}$ cells of the tape before stopping.

The term maximum complexity is justified by the following proposition:

Proposition 1. For every set $X \subseteq A^*$ there are T and p adequate for X and a c such that for every $a \in X$ T stops after no more than $c^{\lambda(a)}$ steps.

Proof: Let p be a list of the members of X such that a precedes b if $\lambda(a) < \lambda(b)$ and separated by single blanks. Then it is routine to define T and c satisfying *Proposition 1.*

1.2 It is the purpose of this paper to discuss a different interpretation of the above notions of complexity (Section 3) and to prove (Section 2) the following theorem about arithmetic.

Let L be a formal language for first order arithmetic with countably many individual variables and finitely many other symbols among which there are symbols for 1, for the functions $x + y$, xy and x^y, and for the relation $<$. Let all the symbols of L be coded by words of A* so that justapositions of such words are unambigously interpretable as ordered sequences of symbols of L (it follows from this that A has at least 2 letters different from $\square$ and 2 are of course sufficient).

Let BA (bounded arithmetic) be the set of all words of A* corresponding in this way to sentences (i.e., formulas without free variables) of L with bounded quantifiers (i.e., only quantifiers of the kind $(\exists x < t)$ or $(\forall x < t)$, where t is any term) which are true in the natural interpretation of L over the universe $\{1, 2, \ldots\}$. It is well known that BA is a decidable set.

Theorem 2. BA is of maximum complexity.

The intuitive meaning of Theorem 2 is that the size of the computer needed to decide the validity of arithmetical formulas with bounded quantifiers has to grow exponentially with the length of the formulas.

Remark. The fact that A has to have at least 2 letters different from $\square$ in the above coding of L follows also from Theorem 2 and the following easy proposition.

Proposition 3. If $A = \{\quad, a\}$, then every problem $X \subseteq A^*$ is of polynomial complexity.

In fact it is easy to define a p and a T adequate for X such that for every $a \in X$ T stops after no more than $c(\lambda(a))^2$ steps (and visits no more than $2\lambda(a) + 1$ cells of the tape).

1.3 Let us mention the following open questions related to Theorem 2.

Is Theorem 2 valid if BA is substituted by any of the following problems: (a) this part of BA in which the symbol for exponentiation does not appear; (b) the set of all equations which belong to BA.

Of course BA contains rather deep theorems of small length (if the coding is natural) e.g.,

$$\forall\, x, y, z < 10^{10^{10}}\ [x^5 + y^5 \neq z^5].$$

The problems in (a) and (b) seem much simpler.

1.4 One can formulate related notions of complexity in which the program p no longer appears (or equivalently $p = \emptyset$ the empty word). The questions (a) and (b) are open even now.

With $p = \emptyset$ Proposition 1 is no longer true as the following theorem of Albert Meyer [5] shows. Let WMS be the weak monadic second order theory of the function $x + 1$ over the universe $\{1, 2, \ldots\}$, coded similarly as BA was coded above. Let us define $t(0, n) = n$ and $t(k + 1, n) = 2^{t(k, n)}$.

Theorem (A. Meyer). There exists a constant $c > 0$ such that for every T such that $p = \emptyset$ and T are adequate for WMS there are infinitely many $a \in WMS$ such that T applied to a as above visits not less than $t([c \log \lambda(a)], \lambda(a))$ cells of the tape before stopping.

We do not know if this theorem is also valid with WMS substituted by BA nor if WMS is of maximum complexity.

Let us recall also a class of open questions studied by S. Cook [1] and R. Karp [3]. They are all equivalent to the problem of time necessary for checking tantologies of any of the classical formalisms of propositional calculus. Here not much tape is needed (it grows linearly with the length of the formula to be checked) but the number of steps required seem to grow exponentially with the length of the formula (strictly speaking, the number of variables in the formula) for every p and T adequate for this problem.

The above examples seem to indicate that the number of steps and the amount of tape are relatively independent measures of complexity of problems (although the first is not less than the second and the second is bounded by an exponential function of the first, since otherwise loops would appear).

2. *PROOF OF THEOREM 2.*

2.1 Let α be the number of letters in A and σ_T be the number of inner states of T. We can assume without loss of generality that $\alpha \geq 2$ (i.e., there is a letter different from $\square$). Let $A^*\lceil n = \{a \in A^* : \lambda(a) = n\}$, and $p\lceil m$ be the initial segment of p of length m. We shall say that p and T are n-adequate for $X \subseteq A^*$ if for every $a \in A^*\lceil n$ T applied to a and p as in Section 1.1 stops if and only if $a \in X$.

Lemma 4. For every n there is a set $X_n \subseteq A^*\lceil n$ such that if $p\lceil m$ and T are n-adequate for X_n then

$$m \geq ((\alpha - 1)^n - \alpha(\sigma_T - 1)\ (1 + \log_2(\alpha\sigma_T)))/\log_2\alpha . \tag{1}$$

Proof. There are $2^{(\alpha - 1)^n}$ sets $X \subseteq A^*\lceil n$, there are α^m possible sequences $p\lceil m$ and there are $(2\alpha\sigma_T)^{\alpha(\sigma_T - 1)}$ machines T with states for the alphabet A.

Therefore if m and σ are such that for every $X \subseteq A^*\lceil n$ there exists a T with σ states and a p such that $p\lceil m$ and T are n-adequate for X then

$$2^{(\alpha - 1)^n} \leq \alpha^m\ (2\alpha\sigma_T)^{\alpha(\sigma_T - 1)} ;$$

and (1) follows.

Corollary 5. If $\alpha > 2$ then there exists a problem $X \subseteq A^*$ which is of maximum complexity.

Proof. Let X_n be a set as in Lemma 4. We put $X = \bigcup_{n=1}^{\infty} X_n$. Let now p and T be adequate for X. It follows that for every n there exists an $a \in X_n$ such that T applied to a and p visits all of $p\lceil m$, where m satisfies (1). Thus, since $\alpha - 1 \geq 2$, m grows exponentially with n, Q.E.D.

2.2 Proof of Theorem 2. Let A_0 be an alphabet with 3 letters including $\square$. Let us order A_0 and consider a lexicographical ordering of all $a\varepsilon A_0^*$ with $\lambda(a) = n$ and finally a lexicographical ordering of all the sets $X \subseteq A_0^* \restriction n$. Let X_n^0 be the first set $X \subseteq A_0^* \restriction n$ satisfying Lemma 4 with $A = A_0$. If we treat the letters in $A_0 - \{\square\}$ as binary digits then every word $a\varepsilon A^*$ can be treated as a binary expansion of an integer $2^{\lambda(a)} + k(a)$ where $k(a) < 2^{\lambda(a)}$ from which the leftmost digit was deleted. Now it is routine (but tedious) to construct (by analyzing the definition of X_n^0) a formula $\Phi(v)$ of the language L (see 1.2) with one free variable v, with bounded quantifiers and the following property: $\Phi[2^n + k]$, where $k < 2^n$, is true if and only if the binary expansion of $2^n + k$ with the leftmost digit deleted belongs to X_n^0. (The only Φ which we know involves the symbol for the exponential function (2^x could replace x^y)). Therefore $\Phi[2^{\lambda(a)} + k(a)]$ is true iff $a\varepsilon X^0_{\lambda(a)}$ i.e., iff $a\varepsilon \bigcup_{n=1}^{\infty} X_n^0$.

It is also routine to find a constant $c_0 > 0$ and a Turing machine T_0 such that T_0 applied to an $a\varepsilon A_0^*$ prints a constant term t(a) in L denoting the number $2^{\lambda(a)} + k(a)$, and such that $\lambda(t(a)) \leq c_0\lambda(a)$, where $\lambda(t(a))$ denotes the length of the BA-code of t(a). Moreover we can assume that T_0 does this in no more than $c_0\lambda(a)$ steps.

Let now q(a) be the BA-code for the formula $\Phi(t(a))$. Clearly $\phi(a)\ \alpha\varepsilon$ iff $a\ \bigcup_{n=1}^{\infty} X_n^0$. There is a constant $c_1 > 0$ such that $\lambda(\phi(a)) \leq c_1\lambda(a)$ for all $a\varepsilon A_0^*$ and a Turing machine T_1 such that T_1 applied to a prints $\phi(a)$ in no more than $c_1\lambda(a)$ steps. Hence if BA was not of maximum complexity then $\bigcup_{n=1}^{\infty} X_n^0$ would not be of maximum complexity contrary to the definition of this set. Q.E.D.

3. A DIFFERENT FORMALISM

The pair T, p used above is not as directly interpretable, say in biology or computer science, as one may wish for. We shall formulate now a different formalism which permits more direct interpretations of this sort. This is related to the concept of k-continuous functions studied in [2] and [4].

Let A be a finite alphabet with α letters and A^m be the set of sequences of letters of A of length m. A set $C \subseteq A^m$ will be called a k-cylinder ($k \leq m$) if $C = \{(a_1,\ldots,a_m) \in A^m : (a_{i_1},\ldots a_{i_k}) = (c_1,\ldots,c_k)\}$, for some $(c_1,\ldots,c_k) \in A^k$ and $1 \leq i_1 < \ldots < i_k \leq m$.

A function $f:A^m \to A$ will be called k-continuous if for every $a \in A^m$ there exists a k-cylinder C with $a \in C$ and such that $f(x) = f(a)$ for all $x \in C$. A function $f:A^m \to A^n$, where $f = (f_1,\ldots,f_n)$, will be called k-continuous if all the functions f_i are k-continuous. And f will be called uniformly k-continuous if every f_i depends on k variables at most.

Theorem 6. For every A and k there is a constant κ such that, for every $m \geq k$, every k-continuous function $f:A^m \to A$ depends on κ variables at most.

Proof. If $\alpha = 2$ then, by Theorem 15 of [2], exists and $\kappa \leq (2k - 1)\binom{2k - 2}{k - 1}$. The general case clearly follows from this by coding the letters of A with a 2-letter alphabet.

From Theorem 6 we get immediately the following corollaries.

Corollary 7. Every k-continuous function $f:A^m \to A^n$ is uniformly κ-continuous, where κ is as in Theorem 6.

Corollary 8. If $m \geq \kappa$ then there are no more than $\alpha^{n\alpha^{\kappa}}\binom{m}{\kappa}^n$ uniformly κ-continuous functions $f:A^m \to A^n$.

A function $f:A^m \to A^m$ can be interpreted as a computer (or an organism) in the following way. A sequence $(a_1,\ldots,a_m)\in A^m$ is the state of the computer (i.e., the content of its memory). f is the transition function i.e., $f(a_1,\ldots,a_m)$ denotes the next state of the computer. The condition that f, i.e., all f_i, be k-continuous is a very natural one, met by all actually existing computers and with a very small k if the time scale is fine enough. The following propositions show how Boolean nets and Turing machines can be interpreted as uniformly k-continuous functions $f:\{0,1\}^m \to \{0,1\}^m$ with quite small k (in the case of Turing machines with an infinite tape m is infinite).

Proposition 9. A Boolean net in which every operation (newron) is binary and works with a unit delay can be interpreted as a uniformly 2-continuous function $f:\{0,1\}^m \to \{0,1\}^m$, where m is the number of edges of the net.

This is obvious since the state of each edge at time t depends only on the state of 2 edges at time t - 1.

Proposition 10. A Turing machine T with σ states for an alphabet A of α letters with a limited tape of length n (n could be infinite too) can be interpreted as a uniformly $[5 + \log_\alpha \sigma]$-continuous function $f:A^m \to A^m$, where $m \leq 2n + \log_\alpha \sigma + 1$.

Proof. Interpret the tape as a function from the integers into A (or else label the tape with integers) such that the square at which the head of the machine is working has label 0. Then the letter in the k'th square of the tape at time t depends only on the letters in the squares labeled k - 1, k, k + 1 and the state of the machine at time t - 1. If

we code the states by means of at most $\log_\alpha \sigma + 1$ letters, Proposition 10 readily follows.

Let now $X \subseteq A^*$ as in Section 1.1. We shall identify words in A^* with sequences of letters.

We put $X_n = \{a \in X : \lambda(a) = n\}$. Let $n \leq m$, $p \in A^{m-n}$, $\square \in A$ and $f: A^m \to A^m$. We shall say that f and p are <u>adequate</u> for X_n if $a \in X_n$ is necessary and sufficient for the existence of an integer r such that

$$f^r(a_1, \ldots, a_n, p_1, \ldots, p_{m-n}) = (\square, \ldots, \square),$$

where $$f^1(x) = f(x) \text{ and } f^{s+1}(x) = f(f^s(x)) \text{ for } s = 1, 2, \ldots$$

Let $C_k(n, X)$ $[C_k^o(n, X)]$ be the minimum $m \geq n$ for which there exists a k-continuous [uniformly k-continuous] function $f: A^m \to A^m$ which is adequate for X_n. Clearly $C_k(n, X)$ and $C_k^o(n, X)$ are measures of complexity of X_n and their rate of growth, when $n \to \infty$, measures the complexity of X.

We shall prove (Theorem 14) that in terms of these measures we can define the concepts of polynomial and maximum complexity introduced in Section 1. First we prove some more basic properties of our functions.

Proposition 11. (i) $C_k(n, X) \leq C_{k+1}(n, X)$; (ii) $C_k^o(n, X) \leq C_{k+1}^o(n, X)$; (iii) $C_k(n, X) \leq C_k^o(n, X)$.

Theorem 12. $C_k(n, X) \leq KC_2^o(n, X)$, where K is a constant which depends only on k and α.

To prove this theorem we need the following fundamental lemma (which is certainly known but we do not know the right reference).

Lemma 13. Let $\phi(m)$ be the smallest integer such that every function $f:A^m \to A$ can be represented by a word involving at most $\phi(m)$ binary function symbols. Then there are constants P and Q depending only on α such that

$$P\frac{\alpha^m}{\log m} \leq \phi(m) \leq Q\alpha^m \quad \text{for } m = 1, 2, \ldots$$

Proof. The lower estimate is obtained as follows. Since a word with s binary function symbols has at most s + 1 variables and thus can be represented in the bracket-free notation by a sequence of at most 2s + 1 letters and since there are α^{α^2} functions $f:A^2 \to A$ therefore,

$$(\alpha^{\alpha^2} + m + 1)^{2\phi(m)+1} \geq \alpha^{\alpha^m}$$

This yields P.

The upper estimate is obtained as follows. f can be represented in the form

(i) $$f(x_1,\ldots,x_m) = \Sigma_{i=1}^{\alpha} g_i(x_1, h_i(x_2,\ldots,x_m)),$$

where $A = \{a_1,\ldots,a_\alpha\}$,

$$g_i(x, y) = \begin{cases} y \text{ if } x = a_i \\ a_i \text{ otherwise,} \end{cases}$$

$$h_i(x_2,\ldots,x_m) = f(a_i, x_2,\ldots,x_m),$$

and $a_1 + x = x$. Clearly the representation (*) involves α (m-1)-ary functions and $2\alpha - 1$ binary functions.

Hence

$$\phi(m) \leq \alpha\phi(m - 1) + 2\alpha - 1, \quad \text{for } m > 2.$$

This yields Q.

Proof of Theorem 12. By Corollary 7 we have a constant κ depending only on k and α such that $C_k(n, X) \leq C^0_\kappa(n, X)$.

By Lemma 13 we can represent every function of κ variables as a composition of at most $Q\alpha^\kappa$ binary functions where Q depends only on α. Such a function can be computed by means of $Q\alpha^\kappa$ steps the result of each step being printed on a special place of the tape. Thus given $f_n:A^m \to A^m$ with $m \leq C^0_\kappa(n, X)$ adequate for X_n we can produce $f^*_n:A^{m^*} \to A^{m^*}$ which is uniformly 2-continuous adequate for X_n and with $m^* \leq mQ\alpha^\kappa$. Hence $C^0(n, X) \leq KC^0_2(n, X)$ and Theorem 12 follows.

Theorem 14. (i) X is of polynomial complexity if and only if there exists a constant c such that $C_2(n, X) \leq n^c$ for $n = 1, 2, \ldots$

(ii) X is of maximum complexity if and only if there exists a constant $c > 1$ such that $C_2(n, X) > c^n$ for infinitely many n's.

To prove this theorem we need the following lemmas.

Lemma 15. For every uniformly k-continuous function $f:A^m \to A^m$ there exists a sequence of words $w_i \in A^{[c \log m]}$ for $i = 1, \ldots, m$ and a Turing machine T for the alphabet A, where c and T depend only on A and k, such that T applied to a tape on which a word

$$(*) \qquad a_1w_1a_2w_2 \ldots a_mw_m \qquad (a_i \in A)$$

is printed will transform this word into the word

(**) $$b_1w_1b_2w_2 \ldots b_mw_m,$$

where $(b_1,\ldots,b_m) = f(a_1,\ldots,a_m)$, and moreover T does not visit more than cm log m squares of the tape (if started at the beginning of (*)).

Proof. Since f is uniformly k-continuous each b_i depends on $a_{s(1,i)},\ldots,a_{s(k,i)}$ only. Let w_i code the sequence $s(1,i),\ldots,s(k,i)$ and the law of this dependence (a function $g_i:A^k \to A$). Moreover let w_i have about $k + \log_\alpha m$ blank spaces (which will be used as mailboxes). Clearly such w_i can satisfy $\lambda(w_i) \leq c \log m$. Now the construction of T satisfying Lemma 15 is routine: T brings the letters $a_{s(j,i)}$ into the mailbox of w_i then it computes b_i and prints it in place of a_i, then it restores w_i keeping only a_i in the mailbox (since it may be needed for the other b_i's) etc., Then T cleans the mailboxes to get (**).

Lemma 16. For every uniformly k-continuous function $f:A^m \to A^m$ there exists a word $w_m \varepsilon A^{[cm \log m]}$ and a Turing machine T_1 where c and T depend only on A and k such that T applied to any word aw_m, where $a\varepsilon A^m$, will transform this word into bw_m, where $b = f(a)$, and moreover T will not visit more than $[m + cm \log m]$ squares of the tape (if started at the beginning of aw_m.

Proof. Routine in view of Lemma 15.

Proof of Theorem 14. If X is of polynomial complexity (see Section 1.1), [is not of maximum complexity] then by Proposition 10 and Theorem 12 $C_2(n, X) \leq n^c$ for a suitable c [for every $c > 1$ there exists an N such that $C_2(n, X) < c^n$ for all $n > N$]. If $C_2(n, X) \leq n^c$ [for every

$c > 1$ there exists an N such that $C_2(n, X) \leq c^n$ for all $n > N$] then by Lemma 16 there exists a program p, namely $p = w_1 \square w_2 \square w_3 \square \ldots$ and a Turing machine T, such that T and p are adequate for X and if T applied to a and p as in Section 1.1 stops then it visits no more than $(\lambda(a))^c$ squares of the tape [for every $c > 1$ there exists an N such that if $\lambda(a) > N$ then T visits less than $c^{\lambda(a)}$ squares of the tape]. Q.E.D.

NOTE: A first draft of this paper was written in 1967 in the language of k-continuous functions, without using Turing machines. Theorem 14 shows the original definitions of polynomial complexity and maximum complexity. In the proof of Theorem 2 (Lemma 4), instead of an estimate of the number of Turing machines, Corollary 8 was used to the same effect.

The present version of this paper was written recently by Jan Mycielski.

REFERENCES

[1] S. Cook, The complexity of theorem proving procedures, Conference Record of Third ACM Symposium on Theory of Computing, 1970, p. 151-158.

[2] A. Ehrenfeucht and J. Mycielski, Interpolation of functions over a measure space and conjectures about memory, Journal of Approximation Theory, (to appear).

[3] R. Karp, Reducibility among combinatorial problems, in the collection Complexity of Computer Computations, ed. R. E. Miller and J. W. Thatcher, Plemum Press, N.Y., 1972.

[4] R. McKenzie, J. Mycielski and D. Thompson, On Boolean functions and connected sets, Math. Systems Theory 5 (1971), p. 259-270.

[5] A. R. Meyer, Weak SIS cannot be decided, Notices of the Am. Math. Soc. 19 (1972), A-598.

cah