Astérisque

GIORGIO AUSIELLO

Difficult logical theories and their computer approximations

Astérisque, tome 38-39 (1976), p. 3-21
<http://www.numdam.org/item?id=AST_1976__38-39__3_0>

© Société mathématique de France, 1976, tous droits réservés.

L'accès aux archives de la collection « Astérisque » (http://smf4.emath.fr/Publications/Asterisque/) implique l'accord avec les conditions générales d'utilisation (http://www.numdam.org/conditions). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM
Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques
http://www.numdam.org/

Société Mathématique de France
Astérisque 38-39 (1976) p.3-21

DIFFICULT LOGICAL THEORIES AND THEIR COMPUTER APPROXIMATIONS

by

Giorgio AUSIELLO

ABSTRACT

The reason why decidable theories may have a very hard decision problem is examined for the particular case of weak second order theory of successor. Following Hartmanis (1975), it is then shown, both in machine independent terms and referring to Turing machines, that the limitations to the practical feasibility of decision procedures are intrinsic in the concept of formal proof system because in any sufficiently powerful formal system we have classes of trivial theorems which are very hard to prove in the system. Analogous results for the subrecursive case rule out the possibility of defining meaningful approximations of hard theories by resource bounded approximations.

1.- INTRODUCTION

In recent years several interesting results in the field of computational complexity have shown that many decidable theories have a very hard decision procedure. For example, it has been shown by M. Fischer and Rabin (1974) that the theory of integers under addition (Presburger arithmetic)

has infinitely many theorems whose shortest proof takes as long as $2^{2^{cn}}$ steps even to be checked (where n is the length of the theorem and $c > 0$).

This kind of results have been considered by Rabin (1974) an impediment to artificial intelligence because most of artificial intelligence projects rely on the ability of automatically proving theorems of some axiomatizable logical theories and the said results show that even by restricting ourselves to "small" decidable theories we still have theorems that are not feasibly provable.

In practice, the reason why mathematicians are nevertheless capable of proving hard theorems is based on the fact that mathematicians usually make use of some heuristic techniques based on the knowledge of the meaning of the theorems they want to prove and this fact dramatically reduces the search space. The problem of defining "measures based on semantic content on spaces of combinatorial problems" has hence been raised by Rabin in order to overcome the said difficulties in the automatic proof of theorems.

Of course, the idea that the "useful" and "meaningful" theorems of a theory somehow might belong to a subset of the theory which might be easier than the whole theory is very interesting (even if it is not clear how the proof procedure might be provided of such a knowledge of the meaning of theorems). The problem would hence be that of defining suitable concepts of "approximation" such that, for example, a sequence of sets of theorems is given whose limit is the theory itself and where all easier and more useful theorems can be achevied in the first elements of the sequence.

In this paper we first show the facts which are at the base of the inherent difficulty of certain decidable theories. As we will see these facts are related to the expressive power of the theories in the sense that the more expressive is the theory (in particular the shorter are the theorems which express properties of Turing machines computations) the harder

is the theory. In the second part, we will concentrate on the existence of arbitrarily hard trivial theorems in all creative theories. This fact has been shown to be an intrinsic consequence of the concept of formal proof system and can be immediately observed in machine independent terms. An analogous result can be shown for every sufficiently expressive subrecursive theory : for any proof procedure there are infinitely many trivial theorems on which the procedure takes as long as it takes on the hardest theorems of the theory.

Consequences of these facts are that if we consider approximations defined by resource bounded computations there is no hope that we can achieve all trivial theorems at any level of the approximating sequence of sets of theorems. Besides since infinite sets of trivial theorems exist and can be effectively found,the subrecursive theories cannot be immune (with respect to lower complexity levels) and are infinitely often speedupable.

2.- HARD DECIDABLE THEORIES

The deepest contributions to the characterization of the inherent difficulty of decidable theories have been given in recent years by the work of Meyer (1972,1973,1974), Stockmeyer (1974), Ferrante (1974), Rackoff(1974); Fisher and Rabin (1974) (systematically presented by Meyer and Stockmeyer (1975)). These studies have been motivated by classical problems of the theory of computational complexity : the search for techniques for proving lower bounds and the search for "natural" hard sets. Anyway the characterization of lower and upper bounds for decidable logical theories is relevant for all studies on mechanizable logics (automatic theorem proving,algebraic manipulation, automatic program verification and symbolic testing).Finally, from the point of view of mathematical logic these results bring an insight in the expressive power of logical theories and are a necessary evolution of the studies on decidability and undecidability of theories (as we will

see the reason that makes some theories hard to decide is deeply related to the reason that makes other theories undecidable, namely the ability of stating that a given Turing machine will halt within a given amount of resource).

Let Φ be any acceptable measure of computational complexity as defined by Blum (1967) (see also Hartmanis and Hopcroft (1971)). In particular we will refer to time and tape for (one tape, one read-write head) Turing machines.

DEFINITION 1.- Given any recursive function t, a complexity class of functions is the class of recursive functions defined by

$$C_t^{\Phi} = \{f \mid (\exists i)(\varphi_i = f)\ (\overset{\infty}{\forall}_i)[\Phi_i(x) \leqslant t(x)]\}\ .$$

The function t is called an upper bound on the complexity Φ of all functions f in C_t^{Φ}.

DEFINITION 2.- Lower bound on the complexity of a function f is any function t such that $(\forall\ i)\ (\varphi_i = f)\ (\overset{\infty}{\exists}\ x)\ [\Phi_i(x) > t(x)]$.

Quite often, especially when we deal either with word functions or with sets of words, it is more meaningful to express the complexity as a function of the size (length) of the input. For this reason we will use the following notations :

DEFINITIONS 3.- Let Σ be an alphabet and $S \subseteq \Sigma^*$ and $f : \Sigma^* \to \Sigma^*$.

We say $\mathrm{Comp}(f) \leqslant t$ (a.e.) iff $(\exists\ i)\ (\varphi_i = f)\ (\overset{\infty}{\forall}\ x)\ [\Phi_i(x) \leqslant t(|x|)]$

$\mathrm{Comp}(f) > t$ (i.o.) iff $(\forall\ i)\ \ (\varphi_i = f)\ (\overset{\infty}{\exists}\ n)\ (\exists\ x)$

$$[\Phi_i(x) > t(n) \wedge n = |x|].$$

Analogously we say $\mathrm{Comp}(S) \geqslant t$ $(< t)$ if $\mathrm{Comp}(f_S) \geqslant t$ $(< t)$ and with f_S we denote the characterizatio function of S. In case we deal with steps or squares required by a Turing machine to compute f we write Time(f) or Tape (f).

The essential problem of the characterization of the complexity of a set S is to find the smallest possible upper bound and the highest possible lower bound for the characteristic function of the set.

The basic abstract technique for establishing lower bounds is the (upward) diagonalization over a complexity class.

<u>FACT 1</u>.- <u>Given any recursive function</u> t <u>there is a decidable set</u> S <u>whose complexity has lower bound</u> t.

<u>PROOF</u>.- Let r be a recursive function such that $(\forall\, i)\ (\overset{\infty}{\exists}\, x)\ [r(x) = i]$. Let the characteristic function of S be defined as follows :

$$f_S(x) = \begin{cases} 1 \dot{-} \varphi_{r(x)}(x) & \text{if } \Phi_{r(x)}(x) \leqslant t(x) \\ 1 & \text{otherwise.} \end{cases}$$

Let e be an index for f_S and let $\bar{x}$ be such that $r(\bar{x}) = e$. Since the first alternative gives a contradiction we have that for all such $\bar{x}$ $\varphi_e(\bar{x}) = 1$ and $\Phi_e(\bar{x}) > t(\bar{x})$.

Q.E.D.

First of all the limitations with this technique are that in general we are given a set S and we want to establish a good lower bound t on its complexity and not vice-versa. Besides, even if we are interested (as people were five years ago) in finding sets with hard (for example, exponential) decision problems the diagonalization technique does not necessarily produce a set which is "natural" (for example, which has an algebraic or logical definition independent from the required computational properties).

The interesting technique which has been developped by the mentioned authors consists in showing that sometimes, in a sufficiently complex set A we can encode a set B whose difficulty is characterized by a diagonalization procedure, thus proving a lower bound on the complexity of A.

This technique actually relies on (downward) diagonalization results first obtained by Hartmanis and Stearns (1965) and by Hartmanis, Lewis and Stearns (1965) and it will be sketched through an example. In particular

the example that we want to show relies on the following :

DEFINITION 4.- A recursive function t is tape constructable iff there is a Turing machine which, started on any input word of length $n \geqslant 0$, halts having used exactly $t(n)$ tape squares.

FACT 2.- If t is tape constructable then there is an $A \subseteq \{0,1\}^*$ such that $\mathrm{Tape}(A) \leqslant t$ and $\mathrm{Tape}(A) > g$ (i.o.) for any g such that $\lim_{n \to \infty} \frac{g(n)}{t(n)} = 0$.

By using essentially these proof methods the following results (surveyed by M. Fisher and Rabin (1974) and by Meyer (1974)) have been proven :

RESULT 1.- (exponential lower bounds). For any of the theories T of the following list there is a constant $c > 0$ such that $\mathrm{Time}(T) > 2^{cn}$ (i.o.) :

- the theory of real numbers under addition,
- the theory of complex numbers under addition,
- the theory of finite cyclic groups,
- the theory of rings of characteristic p.

RESULT 2.- (super exponential lower bounds)

2.1.- For any of the theories T of the following list there is a constant $c > 0$ such that $\mathrm{Time}(T) > 2^{2^{cn}}$ (i.o.) :

- the theory of natural numbers under addition (Presburger arithmetic)
- the theory of finite Abelian groups.

2.2.- There is a constant c such that $\mathrm{Time}(T) > 2^{2^{2^{cn}}}$ where T is the theory of natural numbers under multiplication.

RESULT 3.- (non elementary lower bounds). For any of the theories T of the following list there is a constant $c > 0$ such that :

$$\mathrm{Time}(T) > 2^{2^{\cdot^{\cdot^{2^{n}}}}} \Big\}\, c \log n$$

- the weak monadic second order theory of successor,
- the theory of linear orders,
- the theory of two successors and prefix,
- the theory of single unary function,
- the theory of pure finite types,
- the theory of integers under addition and the predicate "x is a power of 2 and x divides y,
- the theory of pairing functions.

3.- THE PROOF OF INTRINSIC DIFFICULTY

The example that we will consider is the weak second order theory of the integers with the predicates $x = y + 1$ and $x \in X$.

Let LSIS be the language of the theory and let WSIS $\subseteq$ LSIS be the set of true sentences.

THEOREM 1.- (Meyer (1973)). Let Z be a Turing machine which started on a sentence $x \in$ LSIS eventually halts in a given final state iff $x \in$ WSIS ; then for every $K \geqslant 0$ there are infinitely many n for which Z takes at least

$$t_k(n) = \left. 2^{2^{\cdot^{\cdot^{\cdot^{2^n}}}}} \right\} k$$

steps and tape squares for some sentences of lenght n.

SKETCH OF THE PROOF.- The proof requires the following :

DEFINITION 5.- Let $A, B \subseteq \Sigma^*$; A is reducible to B in polynomial space ($A \leqslant_{PTAPE} B$) iff there is a polynomial p and a total recursive function $f : \Sigma^* \to \Sigma^*$ such that :

i) $x \in A$ iff $f(x) \in B$

ii) $Tape(f) \leqslant p$.

LEMMA 1.- Suppose that for any k, given any set $A \subseteq \{0,1\}^*$ such that $Tape(A) \leq t_k$ (a.e.) we had $A \leq_{PTAPE} WSIS$; then for every k, $Tape(WSIS) > t_k$ (i.o.).

PROOF.- By fact 2, given any k, there is a set A_k such that $Tape(A_k) \leq t_k$ (i.o.) and $Tape(A_k) \leq t_{k+1}$ (a.e.). Since $A_k \leq_{PTAPE} WSIS$, given A_k and x there is a sentence $F[A_k,x]$ such that $x \in A_k$ iff $F[A_k,x] \in WSIS$ and $F[A_k,x]$ can be computed within space $p_k(|x|)$, for some polynomial p_k. If $F[A_k,x] \in WSIS$ could be decided (a.e.) within tape $t_{k-1}(|F[A_k,x]|) \leq t_{k-1}(p_k(|x|))$ then $x \in A_k$ could be decided within tape $p_k(|x|) + t_{k-1}(p_k(|x|)) \leq t_k(|x|)$ (a.e.) which would contradict the hypothesis. Since there are infinitely many $\bar{x}$ such that $\bar{x} \in A_k$ requires more than $t_k(|\bar{x}|)$ squares to be decided, there are infinitely many sentences $\bar{F}$ such that $\bar{F} \in WSIS$ requires more than $t_{k-1}(|\bar{F}|)$ squares (namely $\bar{F} = F[A_k,\bar{x}]$).

Q.E.D.

To complete the proof we have to show that $A \leq_{PTAPE} WSIS$ for any set A which is computable within tape t_k. In order to show this fact we first show how we can express a Turing machine computation with a "regular-like" expression β such that the computation uses elementary space iff the language of β is empty and then we show how we can express the fact that the expression β is empty as a sentence in LSIS. Thus we need :

DEFINITION 6.- For any Σ, γ-expressions over Σ and the languages they represent are defined as follows :

- for any $\sigma \in \Sigma$, $\bar{\sigma}$ is a γ-expression, $L(\bar{\sigma}) = \{\sigma\}$
- if α and β are γ-expressions

$(\alpha \cup \beta)$	is a γ-expression,		$L(\alpha \cup \beta) = L(\alpha) \cup L(\beta)$
$(\alpha \,.\, \beta)$	"	"	$L(\alpha \,.\, \beta) = L(\alpha) \,.\, L(\beta)$
$\neg(\alpha)$	"	"	$L(\neg(\alpha)) = \Sigma^* - L(\alpha)$
$\gamma(\alpha)$	"	"	$L(\gamma(\alpha)) = \bigcup_{x \in L(\alpha)} \Sigma^{\lvert x \rvert}$

<u>DEFINITION 7</u>.- <u>Empty</u> $(\Sigma) = \{\alpha \mid \alpha$ <u>is a</u> γ-<u>expression over</u> Σ <u>and</u> $L(\alpha) = \emptyset\}$.

<u>LEMMA 2</u>.- <u>Let</u> k <u>be given and</u> $A \subseteq \{0,1\}^*$; <u>if</u> $\mathrm{Tape}(A) \leqslant t_k$ <u>then there is</u> Σ <u>such that</u> $A \leqslant_{PTAPE} \mathrm{Emty}(\Sigma)$.

<u>PROOF</u>.- Let the set A be given. By hypothesis there is a Turing machine $Z_{\bar{A}}$ which recognizes the complement of A on tape t_k. Let the machine $Z_{\bar{A}}$ operate on input $b^{f_k(n)} x\, b^{f_k(n)}$ where $|x| = n$ and $f_k(n) \geqslant t_k(n)$ and suppose $Z_{\bar{A}}$ stops in an accepting final state iff $x \in \bar{A}$ iff $x \notin A$. It is possible to show that the computation of $Z_{\bar{A}}$ on the given input can be expressed by a γ-expression β whose length is bounded by $c(|x\alpha_k|)$ where α_k is the γ-expression for $\Sigma^{f_k(n)}$ and $|x| = n$. Since we can also show that $|\alpha| \leqslant p(n)$ for some polynomial p, we have that given x we can derive β on polynomial tape such that $x \in A$ iff $Z_{\bar{A}}$ rejects x iff $\beta \in \mathrm{Empty}(\Sigma)$. We still have to show how to construct such a γ-expression β which represents a $Z_{\bar{A}}$ computation rejecting x iff $x \in A$. First let us observe that if $\mathrm{Comp}(Z_{\bar{A}}, b^{f_k(n)} x\, b^{f_k(n)})$ is the set of strings corresponding to an accepting computation of $Z_{\bar{A}}$, it must be the following set

$$\{ \circledast\ \mathrm{id}(Z_{\bar{A}},x,0) \circledast\ \mathrm{id}(Z_{\bar{A}},x,1) \circledast \ldots \circledast\ \mathrm{id}(Z_{\bar{A}},x,t) \circledast \}$$

where :

i) $\mathrm{id}(Z_{\bar{A}},x,0) = b^{f_k(n)} (q_o,u)\, w\, b^{f_k(n)} \qquad u\, w = x \quad |x| = n$

ii) $(\forall\ e)$ $\mathrm{id}(Z_{\bar{A}},x,e+1)$ is the i.d. which follows $\mathrm{id}(Z_{\bar{A}},x,e)$ as required by the transition function of $Z_{\bar{A}}$,

iii) $(\forall\ e)$ $|\mathrm{id}(Z_{\bar{A}},x,e)| = 2\, f_k(n) + n$,

iv) $\mathrm{id}(Z_{\bar{A}},x,t)$ is the first i.d. containing the final state.

For this reason if we want to define the set $\neg\, \mathrm{Comp}(Z_{\bar{A}}, b^{f_k(n)} x\, b^{f_k(n)})$ we can charactize it in the following way :

i) words that do not begin by $\circledast\ b^{f_k(n)} (q_o,u)\, w\, b^{f_k(n)} \circledast$ or

ii) words that do not contain the final state q_F, or

iii) words that do not end with $\circledast$, or

iv) words that violate the transition function.

These four sets of words can be described by the formulae

$$\beta_1 = \neg(\circledast \,.\, (L(\alpha_k) \cap b^*) \,.\, (q_o,u) \,.\, (L(\alpha_k) \cap b^*) \,.\, \circledast \,.\, \Sigma^*)$$

$$\beta_2 = \neg(\Sigma^* \,.\, (\{q_F\} \times T) \,.\, \Sigma^*)$$

$$\beta_3 = \neg(\Sigma^* \,.\, \circledast)$$

$$\beta_4 = \bigcup_{\sigma_1\sigma_2\sigma_3 \in \Sigma} (\Sigma^* . \sigma_1\sigma_2\sigma_3 . L(\alpha_k) . \Sigma^{n-1} . L(\alpha_k) . (\Sigma - f_Z(\sigma_1\sigma_2\sigma_3)) \,.\, \Sigma^*)$$

where $L(\alpha_k) = \Sigma^{f_k(n)}$, $\Sigma = \{\circledast\} \cup T \cup (K \times T)$, K et T are respectively the set of states and the set of characters of $Z_{\overline{A}}$ and $f_Z : \Sigma \times \Sigma \times \Sigma \to \Sigma$ is the transition function of $Z_{\overline{A}}$ $\beta_1\beta_2\beta_3\beta_4$ can be easily put in the form of γ-expressions. Now, if $\beta = \neg(\beta_1 \cup \beta_2 \cup \beta_3 \cup \beta_4)$, $L(\beta) = \emptyset$ iff $(\beta_1 \cup \beta_2 \cup \beta_3 \cup \beta_4) = \Sigma^*$ iff there is no accepting computation of $Z_{\overline{A}}$ on input $b^{f_k(n)} x\, b^{f_k(n)}$.

Q.E.D.

Finally we have to show

<u>LEMMA 3</u>.- <u>For any</u> Σ, $\mathrm{Empty}(\Sigma) \leq_{PTAPE} \mathrm{WSIS}$

<u>PROOF</u>.- The proof will be given for the case $\Sigma = \{0,1\}$. Let α be a γ-expression over Σ. We construct $F_\alpha \in$ LSIS such that $F_\alpha(n,m,M)$ is true iff $[[n<m$ and $\sigma_{i_n} \ldots \sigma_{i_{m-1}} \in L(\alpha)]$ or $[n=m$ and $\epsilon \in L(\alpha)]]$

$$\text{where } \sigma_{i_k} = \begin{cases} 0 & \text{if } k \notin M \\ 1 & \text{if } k \in M \end{cases}$$

$\alpha = \overline{0}$ $\quad F_\alpha(x,y,X)$ is $[y = x+1 \wedge \neg(x \in X)]$

$\alpha = \overline{1}$ $\quad F_\alpha(x,y,X)$ is $[y = x+1 \wedge x \in X]$

$\alpha = (\beta . \delta)$ $F_\alpha(x,y,X)$ is $(\exists z)[x \leqslant z \wedge z \leqslant y \wedge F_\beta(x,z,X) \wedge F_\delta(x,y,X)]$

$\alpha = \gamma(\beta)$ $F_\alpha(x,y,X)$ is $(\exists x_o)[F_\beta(x,y,X_o)]$

$\alpha = (\beta \cup \delta)$ $F_\alpha(x,y,X)$ is $F_\beta(x,y,X) \vee F_\delta(x,y,X)$

$\alpha = \neg(\beta)$ $F_\alpha(x,y,X)$ is $x \leqslant y \wedge \neg F_\beta(x,y,X)$.

It is clear that given α, a Turing machine (on polynomial tape $p(|\alpha|)$) can produce the sentence $F = \neg(\exists x)(\exists y)(\exists X)[F_\alpha(x,y,X)]$ such that F is true iff $\alpha \in \text{Empty}(\{0,1\})$.

Q.E.D.

This completes the proof of theorem 1. Q.E.D.

4.- ON THE LENGTH OF PROOFS OF TRIVIALITIES

In this paragraph we want to show that among the hardest theorems of a decidable theory (such as the theory considered on the preceding paragraph) there are always infinitely many trivial theorems which might be recognized by very simple algorithms (finite automata, for example). A consequence of this fact is that any proof procedure for such theories can be sped up infinitely often by a substantial amount,thus showing how natural i.o. speed-ups are, in contrast with the "unnatural" a.e. speedupable functions introduced by Blum (1967) and the a.e. hard functions studied by Blum and Gill (1974).

Let us first consider the case of axiomatizable theories which have a creative set of theorems (Peano's arithmetic, for example). The following results (first discovered by Blum) have been given a simple proof by Hartmanis (1975). This proof is related to the techniques for the proof of the intrinsic difficulty of theories which have been used before (as it will be clearer in the next pages) and can be stated in machine independent terms.

Let $L_o = \{ < i,j,x > \mid \varphi_i(< i,j,x >)$ is defined and

$\varphi_i(< i,j,x >) \neq 1$

or $\Phi_j(< i,j,x >)$ is defined and

$\Phi_i(< i,j,x >) \geqslant \Phi_j(< i,j,x >) \}$.

<u>THEOREM 2</u>.- i) L_o <u>is creative</u> ;

ii) <u>there is a function</u> h <u>such that for any program</u> k <u>for the semicharacteristic function of</u> L_o <u>and for any function</u> g <u>there is an infinite subset</u> $A \subseteq L_o$ <u>which can be recognized within resource</u> h <u>but on which</u> φ <u>takes at least resource</u> g.

<u>PROOF</u>.- i) For any $\bar{x}$, the function $\sigma = \lambda k(< k,k,\bar{x} >)$ is a productive function for L_o, in fact, given any $W_k \subseteq \bar{L}_o$ we can show $\sigma(k) \in \bar{L}_o \cap \bar{W}_k$ for $\sigma(k) \in W_k$ would imply that $\varphi_k(< k,k,\bar{x} >)$ is defined (by definition of W_k) and $\sigma(k) \in L_o$ (by definition of L_o), which is a contradiction and, on the other side, $\sigma(k) \in L_o$ would imply that $\varphi_k(< k,k,\bar{x} >)$ is defined (by definition of L_o) and $\sigma(k) \in W_k$ (by definition of L_o) which also is a contradiction.

ii) Let i_o be any program for the characteristic function of L_o, f_{L_o}, and let g be a recursive function. Let Φ_{j_o} be any total running time larger than g. Consider $A = \{ < i_o,j_o,x > \mid x \in N\}$. Since i_o and j_o are constants and x ranges over N, A can be easily recognized in any formalism. Let h be any upper bound to the complexity of A in the given formalism. On the other side, on input $< i_o,j_o,x >$, φ_{i_o} gives output 1 and takes at least Φ_{j_o} steps. In fact $\varphi_{i_o}(< i_o,j_o,x >) = 1$ and $\Phi_{i_o}(< i_o,j_o,x >) < \Phi_{j_o}(< i_o,j_o,x >)$ implies $< i_o,j_o,x >$ not in L_o which contradicts the fact $\varphi_{i_o} = f_{L_o}$, and $\varphi_{i_o}(< i_o,j_o,x >) = 0$ implies $< i_o,j_o,x > \in L_o$ which also contradicts the assumption $\varphi_{i_o} = f_{L_o}$. Q.E.D.

Theorem 2 shows that any decision procedure for L_o is arbitrarily

inefficient on a trivial set of elements. Since all creative sets are recursively isomorphic the theorem can be stated for every creative set.

COROLLARY.- Given any creative set C, let φ_i be a semi-decision procedure for C and let g be any recursive function there is a subset $A \subseteq C$ which can be recognized within resource h but on which φ_i takes at least resource g o h.

The proof of theorem 2 clearly relates the existence of hard trivial theorems to the ability of a formalism of encoding Turing machines computations. In the following theorems 4 and 5 that are independently due to Meyer (see Stockmeyer (1974)) and Hartmanis (1975), the connection is made more explicit and the degree of difficulty of the trivial theorems is related to the expressive power of the theory that is to the succintness of formulae which encode "large" Turing machine computations.

First let us consider again the diagonalization procedure of fact 1 and let us observe that among the hard elements of an hard set defined by a diagonalization procedure there are infinitely many elements which can be very easily recognized.

THEOREM 3.- Let Φ be any complexity measure. There is a function h such that given any function t there is a decidable set S for which any program e such that $\varphi_e = f_S$ takes at least resource t on an infinite subset A of S whose complexity has upper bound h.

PROOF.- The proof is implicit in the proof of fact 1 : the set A of S on which program e for S must take more than t steps is $R_e = \{\overline{x} \mid r(\overline{x}) = e\}$. This set, for any e can be easily recognized within a given resource bound h. For example we can take the values of r to be

$$0,\ 0,\ 1,\ 0,\ 1,\ 2,\ldots,\ 0,\ 1,\ldots,\ n,\ 0,\ 1,\ldots,\ n+1,\ 0,\ 1,\ldots$$

and take h to be the resource needed to recognize R_i in the given complexity measure Φ. h can be determined by "union" theorem (see Hartmanis and Hopcroft (1971)).

If we refer to Turing machines the result becomes more meaningful, as a variation of fact 2.

<u>THEOREM 4</u>.- <u>If</u> t <u>is tape constructable there exists a recursive set</u> A_t <u>such that</u>

i) $\text{Tape}(A_t) \leq t$ <u>and</u> $\text{Tape}(A_t) > g$ (i.o.) <u>for any</u> g <u>such that</u> $\lim_{n\to\infty} \frac{g(n)}{t(n)} = 0$;

ii) <u>if</u> Z_{i_o} <u>accepts</u> A_t <u>there is an infinite regular set</u> $R_{i_o} \subseteq A_t$ <u>such that</u> Z_{i_o} <u>uses at least</u> $\frac{t(n)}{|i_o|}$ <u>on every member of</u> R_{i_o} (*) .

<u>PROOF</u>.- Let $A_t = \{i \circledast w \mid Z_i$ does not accept $i \circledast w$ on tape $\frac{t(|i \circledast w|)}{|i|}\}$ A_t can be easily seen to be recognizable within tape t(n) because on so much tape it is possible to simulate Z_i and accept $i \circledast w$ iff Z_i trues to use more tape than $\frac{t(|i \circledast w|)}{|i|}$ or if Z_i rejects $i \circledast w$ on less tape. If Z_{i_o} is recognizer for A_t, on the regular set $R_{i_o} = \{i_o \circledast w \mid w \in \Sigma^*\}$, Z_{i_o} must accept on at least tape $\frac{t(|i_o \circledast w|)}{|i_o|}$. If in fact Z_{i_o} rejects, then $i_o \circledast w$ is in A_t (which contradicts the assumption that Z_{i_o} recognizes A_t) ; if Z_{i_o} uses less tape and accepts, $i_o \circledast w$ is not in A_t (which also contradicts the assumption that Z_{i_o} recognizes A_t) then Z_{i_o} must accept R_{i_o} on more than $\frac{t(n)}{|i_o|}$ tape which implies $R_{i_o} \subseteq A_t$ and besides for any g such that $\lim_{n\to\infty} \frac{g(n)}{t(n)} = 0$ $\text{Tape}(A_t) > g$ on R_{i_o} .

Q.E.D.

Now let us apply these results to a logical theory T whose expressive power allows us to represent Turing machines computations, in particular to encode the set A_t of theorem 4. Let f and t be two tape constructable functions (with $f(n+1) > f(n)$). Suppose there is a Turing machine Z such that given $i \circledast w$ writes a sentence $F[A_t, i, w]$ in the language of the considered

---

(*) $i \in \{0,1\}^*$ is the code for Z_i.

theory and Z uses at most tape $f(|i \circledast w|)$, that is suppose that in formulae of length bounded by f we can diagonalize over Turing machine computations which use tape bounded by t ; then we can prove.

THEOREM 5.- Any decision procedure for the theory T requires at least tape $t(f^{-1}(n))$ on infinitely many sentences and for any Turing machine Z_{i_o} which recognizes T we can effectively find an infinite subset $T' \subseteq T$ such that T' is recognizable on linear tape but Z_{i_o} uses at least $c.t(f^{-1}(n))$ tape on T' for a constant c.

SKETCH OF THE PROOF.- Suppose Z_{i_o} operates on tape t_{i_o} such that $\lim_{n\to\infty} \frac{t_{i_o}(n)}{t(f^{-1}(n))} = 0$. Then we might recognize A_t by the following Turing machine Z_{k_o} : with input $i \circledast w$ Z_{k_o} computes $F[A_t,i,w]$ and applies Z_{i_o}. Clearly Z_{k_o} accepts A_z within tape $t_{i_o}(f(n))$. But

$$\lim_{n\to\infty} \frac{t_{i_o}(n)}{t(f^{-1}(n))} = 0 \text{ implies } \lim_{n\to\infty} \frac{t_{i_o}(f(n))}{t(f^{-1}(f(n)))} = \lim_{n\to\infty} \frac{t_{i_o}(f(n))}{t(n)} = 0$$

and this contradicts the property of A_t (shown in theorem 5) that A_t is not recognizable within tape g if $\lim_{n\to\infty} \frac{g(n)}{t(n)} = 0$. Now let $T' = \{F[A_t,k_o,w] | w \in \Sigma^*\}$. T' can be shown to be recognizable on linear tape. On the order hand, since Z_{k_o} takes tape $\frac{t(n)}{|k_o|}$ on $\{k_o \circledast w | w \in \Sigma^*\}$, Z_{i_o} must take at least tape $\frac{t(f^{-1}(n))}{|k_o|}$ on T' because $|F[A_t,k_o,w]| \leq f(|k_o \circledast w|)$.

Q.E.D.

5.- COMPUTER APPROXIMATIONS

One of the most natural concepts of approximation of functions are resource bounded approximations.

Let φ_i be the characteristic function of the recursive set A. Let f_k be defined in the following way :

$$f_k(x) = \begin{cases} 1 & \text{if } \Phi_i(x) \leqslant k \\ 0 & \text{otherwise .} \end{cases}$$

The sequence of sets $A_k = \{x \mid f_k(x) = 1\}$ approximates the set A in the limit $k \to \infty$.

More generally we can take the sequence of recursive functions $\{g_i\}_{i=0}^{\infty}$, where $g_{i+1} > g_i$, to be the set of resource bounds and we can give the following :

DEFINITION 8.- Given a partial recursive function φ_i, a resource bounded computation of φ_i is the function :

$$f_k(x) = \begin{cases} \varphi_i(x) & \text{if } \Phi_i(x) \leqslant g_k(x) \\ 0 & \text{otherwise} \end{cases}$$

and the sequence $\{f_k\}_{k=0}^{\infty}$ is said a resource bounded approximation of the (possibly non recursive) function :

$$f(x) = \begin{cases} \varphi_i(x) & \text{if } \varphi_i(x) \text{ is defined} \\ 0 & \text{otherwise.} \end{cases}$$

Non recursive functions which may be approximated by resource bounded computations are a particular case of the limiting recursive functions of Gold (1965). In the case of recursive functions we are interested in approximating hard functions with more simple functions defined by resource bounded computations of the given hard function. For example in the case of the weak second order theory of successor we might choose the sequence of resource bounds to be $\{t_k\}_{k=0}^{\infty}$ (where :

$$t_k(x) = \left. 2^{2^{\cdot^{\cdot^{\cdot^{2^{|x|}}}}}} \right\} k$$

and where the resource is the tape used by Turing machines), choose a particular machine Z which recognizes WSIS and define the following sets of theo-

rems :
$WSIS_k = \{w \in LSIS \mid w$ is decided by Z within tape $t_k(w)\}$ which give a resource bounded approximation of WSIS.

By defining such a hierarchy of sets of theorems we hope that the easier and more natural theorems fall within the first levels. Instead, as a consequence of theorem 1 we have the following :

FACT 3.- Let $\{t_k\}_{k=0}^{\infty}$ be a set of resource bounds. Given any k and given any proof procedure for WSIS there are infinitely many theorems which can be recognized within tape t_o but which are not reached at any level $WSIS_{k'}$, (with $k' \leqslant k$) of the computer approximation of WSIS.

As a consequence of theorem 5 analogous result can be shown for all logical theories which have enough expressive power to represent Turing machine computations.

Finally, for the case of creative sets we have the following :

FACT 4.- Let t be any recursive function. Given any k and any creative set B and any of its semicharacteristic functions Φ_i there is a function g_o and a subset $A \subseteq B$ such that

i) A can be recognized within resource g_o,

ii) no resource bounded computation of Φ_i with resource bounds $\{t^n \circ g_o\}_{n=0}^{\infty}$ gives A until we reach level $t^k.g_o$ of the approximation.

In conclusion resource bounded computations are not suitable to define a concept of approximation which gives all "easily recognizable" theorems : in a given formal system theorems may be very hard even if their meaning is trivial.

REFERENCES

[1] BLUM M. : (1967) A machine independent theory of the complexity of recursive functions. JACM, 14.

[2] BLUM M., GILL J. : (1974) On almost everywhere complex recursive functions. JACM, 21.

[3] FERRANTE J. : (1974) Some upper and lower bounds on decision procedures in logic. Doctoral Thesis, Dept. of Mathematics, M.I.T., Cambridge, Mass.

[4] FISHER M.J., RABIN M.O. : (1974) Super exponential complexity of Presburger arithmetic. Proj. MAC Tech. Memo 43 M.I.T., Cambridge, Mass.

[5] FLAJOLET P., STEYAERT J.M. : (1974) On sets having only hard subsets. Second colloquium on Automata, Languages and Programming, Saarbrücken, Germany.

[6] GOLD M. : (1965) Limiting recursion. J.S.L., 30.

[7] HARTMANIS J., STEARNS R.E. : (1965) On the computational complexity of algorithms. Trans. AMS, 117.

[8] HARTMANIS J., LEWIS P.M., STEARNS R.E. : (1965) Hierarchies of memory limited computations; IEEE Conference on Switching, Circuit Theory and logical Design.

[9] HARTMANIS J., HOPCROFT J.E. : (1971) An overview of the theory of computational complexity. JACM,18.

[10] HARTMANIS J. : (1975) On effective speed-up and long proofs of trivial theorems in formal theories . Dept. of Computer Science, Cornell University, Ithaca, N.Y.

[11] MEYER A.R. : (1972). The inherent difficulty of logical theories. School on Algorithms and Complexity, Oberwolfach, Germany.

[12] MEYER A.R. : (1973) Weak monadic second order theory of successor is not elementary recursive. Proj. MAC, Tech. Memo 38, M.I.T., Cambridge, Mass.

[13] MEYER A.R., STOCKMEYER L.J. : (1975) Inherent computational complexity of decision problems in logic and automata theory. In preparation.

[14] RABIN M.O. : (1974) Theoretical impediments to artificial intelligence. IFIP Congress, Stockholm, Sweden.

[15] RACKOFF C. : (1974) Complexity of some logical theories. Doctoral Thesis, Dept. of El. Eng., M.I.T., Cambridge, Mass.

[16] STOCKMEYER L.J. : (1974) The complexity of decision problems in automata theory and logic. Doctoral Thesis, Dept. of El. Eng., M.I.T. Cambridge, Mass.

Giorgio AUSIELLO
Istituto di Automatica
dell'Università di Roma
Via Eudossiana, 18
00184 ROMA (Italie)