UNSOLVABILITY CONSIDERATIONS IN COMPUTATIONAL COMPLEXITY*

F. D. Lewis

Department of Computer Science
Cornell University
Ithaca, New York

0. ABSTRACT

The study of Computational Complexity began with the investigation of Turing machine computations with limits on the amounts of tape or time which could be used. Latter a set of general axioms for measures of resource limiting was presented and this instigated much study of the properties of these general measures. Many interesting results were shown, but the general axioms allowed measures with undesirable properties and many attempts have been made to tighten up the axioms so that only desirable measures will be defined.

In this paper several undecidability aspects of complexity classes and several sets associated with them will be examined. These sets will be classified by their degree of unsolvability and restrictions will be placed on measures so that these degrees are identical. This gives rise to a new criterion for the "naturalness" of measures and to suggestions for strengthening the measures of complexity.

1. INTRODUCTION

The aim of computational complexity is to classify and study the functions which are computable. This is usually done by placing them into some context using an important characteristic of the function.

Sub-recursive hierarchies were first used to divied these functions into classes and exhibit some of their properties. Examples of this are Grzegorczyk's hierarchy of the primitive recursive functions [6] and the nested recursive functions of Péter [13]. These methods classify functions by their structure, placing limits on the operations which are used to build functions.

In automata theory the recursive functions were classified by limiting some basic resource used in computation. This resource-bounded complexity began with the consideration of Turing machine computations using a limited amount of tape [11, 18] or time [7].

All of the recursive functions were placed in complexity classes according to how difficult they were to compute, or how much of a "natural" resource they used. These classes exhibited many interesting properties and were studies extensively for a number of years. A result due to Ritchie [14], combined with the Union Theorem [9] indicates that the sub-recursive hierarchies are reproduced in the complexity hierarchies when tape length and time are used as measures.

Later a general set of axioms for measures of computation [1] was presented. This involves taking some admissiable enumeration [16] of Turing machines (or partial recursive functions) denoted by: $M_o, M_1, M_2, \dots$ and assigning a measure or step counting function $\Phi_o, \Phi_1, \Phi_2, \dots$ to each machine. The set of measures is designated as Φ and these measures must obey the following two rules:

1) $M_i(x)$ halts $\iff$ $\Phi_i(x)$ halts
2) There is a recursive function C such that for all i, m, and n:

$$C(i,m,n) = \begin{cases} 1 & \text{if } \Phi_i(m) = n \\ 0 & \text{otherwise .} \end{cases}$$

This first axiom indicates that whenever some function computes a value, then a cost of computation can be associated with it. And according to the second rule, the question "Does it cost n to compute $M_i(m)$?" is always recursive.

These axioms allow a very general set of measures for which many interesting results have been derived. The complexity classes formed from the general measures have been studied extensively in regard to their structural [9,3] and naming properties [2]. Also the properties of operators [4,10] have been noted.

Unfortunately, many of the measures allowed under the axioms are so general that not all of the intuitively derivable properties are preserved. Under time and tape as measures, complexity classes are recursively enumerable (r.e.) but measures can be defined so that some of the classes are not. Other properties such as finite invariance and infiniteness are not preserved by measures either.

Therefore some condition must be added to the original two axioms to preserve properties throughout the complexity classes and if possible eliminate the undesirable properties. It is reasonable to expect complexity classes to be r.e. and to

* This research was supported in part by National Science Foundation Grants GJ-155 and GJ-579.

conform by possessing the same properties. This notation will be formalized below.

2. STATE OF THE ART (WITH TWO AXIOMS)

2.1 Complexity Classes

Most of the interesting results in computational complexity have been about the complexity classes or the classes of t-computable recursive functions. These classes are defined as follows:

Definition: The class of t-computable functions is:

$R_t^\Phi = \{\text{total } f \mid \text{there is some } M_i = f \text{ such that } \Phi_i(x) \le t(x) \text{ almost everywhere}\}$.

These classes have been studied in detail for time and tape measures [7,18] and some of the desirable properties which were found carry over to complexity classes defined from general measures. When measures were restricted slightly, then more properties of the natural measures carried over.

One important property of complexity classes under the "natural" measures was the fact that the classes were r.e. In [19], Young wondered if this were true for classes defined from general measures. Regretfully it is not, and this result is presented here and also has been shown independently by Robertson and Landweber [15]. Before this can be show, however, some preliminaries are in order.

To describe classes and their members more intuitively a set of algorithms for computing them must be given. This set "presents" the class and is defined:

Definition: The set A is a presentation for the class: $\underline{C}$ iff A contains an index for each member of $\underline{C}$ and all elements of A are indices for members of $\underline{C}$.

The presentation which immediately comes to mind is the complete presentation or index set for a class.

Definition: The set A is the index set for the class $\underline{C}$ iff:

$$A = \{i \mid M_i = f \in \underline{C}\} .$$

Usually the index set for the class of functions $\underline{C}$ will be designated $\Omega\underline{C}$. ($\Omega\underline{C}$ is used for classes of functions rather than $\Theta\underline{C}$ since Θ has been used in the literature for classes of sets.)

Definition: A class is r.e. iff it has an r.e. presentation.

An interesting type of class is one where an algorithm for any member of the class can be matched with an element of some standard presentation for the class. These classes are used below and are defined as follows.

Definition: $\underline{C}$ is a matchable class iff there is a recursive g and recursive presentation A such that:

$$i \in \Omega\underline{C} \Rightarrow g(i) \in A \text{ and } M_i = M_{g(i)} .$$

Examples: a) $\underline{\text{Const}}$ = {the constant functions} . Let $A = \{a_0, a_1, a_2, \ldots\}$ where $\forall x[M_{a_k}(x) = k]$ and for any x define $g(i) = a_{M_i(x)}$.

b) $\underline{C} = \{f_n\}$ for some recursive set of functions such that $\forall x[f_n(x) < f_{n+1}(x)]$.

c) $\underline{C} = \{f_n\}$ for some recursive set of functions such that $\forall n[f_n(0) = n]$.

Now a measure will be constructed using a matchable class so that an R_t^Φ from this new measure is not r.e. Any of the examples will work, but (a) will be used for reasons of clarity.

Theorem 2.1: For any recursive t there is a measure $\hat\Phi$ such that $R_t^{\hat\Phi}$ is not r.e.

Proof: Let $A = \{a_0, a_1, \ldots\}$ be a recursive presentation of the class $\underline{\text{Const}}$ where for all x , $M_{a_k}(x) = k$. Then for any measure Φ , consider:

$$\hat\Phi_i(x) = \begin{cases} 0 & \text{if } i = a_k \text{ and } M_k(k) \text{ does not halt in } x \text{ steps} \\ t(x) + \Phi_i(x) + 1 & \text{otherwise .} \end{cases}$$

Note that $R_t^\Phi = R_0^\Phi$ and $M_{a_k} \in R_0^\Phi \Leftrightarrow M_k(k)$ never halts.

a) $\hat\Phi$ is a measure since:

1) $\hat\Phi_i(x)$ halts $\Leftrightarrow M_i(x)$ halts since all M_{a_k} are total and Φ is a measure

2) $\hat C(i,m,n) \equiv \hat\Phi_i(m) = n$ is recursive since is can be described:

$$C(i,m,n) = \begin{cases} C(i,m,n-t(m)-1) & \text{if } n > t(m)+1 \text{ and a) } i \notin A \text{ or b) } i = a_k \text{ and } M_k(k) \text{ halts in } m \text{ steps} \\ 1 & \text{if } n = 0 ,\ i = a_k \text{ and } M_k(k) \text{ doesn't halt in } m \text{ steps} \\ 0 & \text{otherwise .} \end{cases}$$

b) $R_t^{\hat\Phi} = R_0^{\hat\Phi}$ is not r.e.

Assume $R_t^{\hat\Phi}$ is r.e. and let $B = \{b_0, b_1, \ldots\}$ be an r.e. presentation of it. Due to the construction of

$\hat{\Phi}$ all b_k must be indices for constant functions and in fact:

$M_{b_i}(x) = k \iff M_{b_i} = M_{a_k}$.

Therefore the set $\{a_{M_{b_0}(x)}, a_{M_{b_i}(x)}, \ldots\}$ is an r.e. set and is exactly the set $\{a_k | M_k(k)$ never halts$\}$.

This set is obviously recursively isomorphic to the well known set $\overline{K} = \{k | M_k(k)$ never halts$\}$ which is not r.e.

So from this contradiction, it can be concluded that R_t^{Φ} is not r.e.

In the previous proof the productive set $\overline{K}$ was used in order to produce the non-r.e. class R_t^{Φ} . This class is productive [5] but if some set other than $\overline{K}$ is used then an immune (contains no infinite r.e. subclasses) class would have been formed.

2.2 Isomorphism Types

Most of the important properties in automata theory are preserved under isomorphisms. These are the recursively invariant properties and when sets are classified under recursive isomorphisms, all the sets in any isomorphism type possess the same properties. These concepts are defined:

Definition: P is a recursively invariant iff for any 1-1, onto, recursive function f , if the set A has P then so does f(A) .

Definition: A is recursively isomorphic to B ($A \equiv B$) iff there is a 1-1, onto, recursive f such that $B = f(A)$.

By a theorem of Myhill [12], the isomorphism types (sets equivalent under isomorphisms) are the same as the equivalence classes (or 1-degrees) under 1-1 reducibility. The reducibilities used here are defined as follows.

Definition: A is 1-1 reducible to B ($A \leq_1 B$) iff there is a 1-1, recursive function g such that for all x :

$x \in A \iff g(x) \in B$.

Definition: A is 1-1 equivalent to B ($A \equiv_1 B$) iff $A \leq_1 B$ and $B \leq_1 A$.

Definition: A is Turing reducible to B ($A \leq_T B$) iff there is a machine with B written on one tape which can decide membership in A .

Definition: A is Turing equivalent to B ($A \equiv_T B$) iff $A \leq_T B$ and $B \leq_T A$.

Hierarchies which result from the reducibilities outlined above can be used to describe sets very precisely, but almost no intuitive information about a set is given by its place in the hierarchy. Therefore another hierarchy, the Arithmetical Hierarchy, will be used in conjunction with the 1-degrees. This hierarchy reflects the structure of a set according to the number of alternating quantifiers in the expression of its membership problem. The membership problem for a Σ_n set will begin with a "$\exists$" and contain n alternating quantifiers, while a Π_n set begins with a "$\forall$" .

A pictorial representation of the Arithmetical Hierarchy appears below. The lines slanting down towards the right denote the upper boundaries of the Σ_n areas, while the lines slanting down to the left from the top of the Π_n areas.

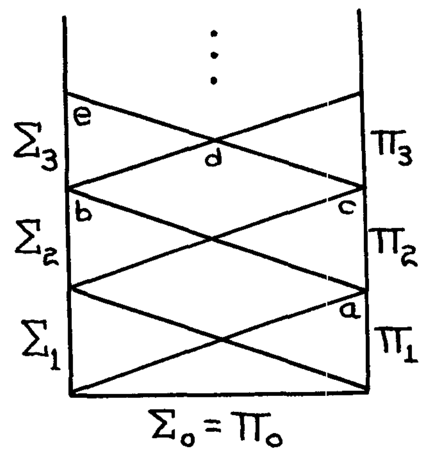

Figure 1.

The locations of the following well-known index sets are indicated in the diagram.

a) $\Theta\phi = \{i | M_i$ never accepts$\}$

b) ΘFinite $= \{i | M_i$ accepts a finite set$\}$

c) $\Theta N = \{i | M_i$ accepts everything$\}$
$= \Omega$Total
$= \{i | M_i$ is a total function$\}$

d) ΩBound $= \{i |$ the range of M_i is bounded$\}$
$= \{i | i \in \Theta N$ and $\exists k \forall x [M_i(x) \leq k]\}$

e) ΘCofinite $= \{i | M_i$ accepts a cofinite set$\}$

These sets are all complete or maximal in their respective locations in the hierarchy when ordered by 1-degrees.

2.3 Index Sets of Complexity Classes

Now the total presentations for the complexity classes $R_t^{\Phi}(\Omega R_t^{\Phi})$ will be classified by the methods outlined above. This classification will be used to suggest a criterion for measures and possibly help to discover method for strengthening the axioms.

First, the index sets of complexity classes formed from the "natural" measures qill be classified. This will be done for time (the number of steps in a computation) but the proof may be done more elegantly via two theorems in section 3, but it is presented here to emphasize that the r.e.-ness of R_t^Φ and the fact that each R_t^Φ contains all finite variants for some function are important properties of time as a measure.

<u>Theorem 2.2</u>: $\Omega R_t^\Phi \equiv \Omega\underline{\text{Bound}}$ for Φ = time.

<u>Proof</u>: a) $\Omega R_t^\Phi \leq_1 \Omega\underline{\text{Bound}}$

Since R_t^Φ is an r.e. class [7] for time as a measure, let $A = \{a_o, a_1, \ldots\}$ be an r.e. presentation for R_t^Φ. Then for any M_i define the machine $M_{g(i)}$ as follows:

$$M_{g(i)}(0) = \begin{cases} 0 \text{ if } M_i(0) = M_{a_o}(0) \\ 1 \text{ if } M_i(0) \neq M_{a_o}(0) \\ \text{diverge if } M_i(0) \text{ diverges.} \end{cases}$$

Then assuming that for input $n - 1$:

$$M_{g(i)}(n-1) = \begin{cases} j \text{ if } M_i(k) = M_{a_j}(k) \\ j+1 \text{ if } M_i(k) \neq M_{a_j}(k) \\ \text{diverge if } M_i \text{ diverges for any } x \leq k. \end{cases}$$

Then if $M_i(k) = M_{a_j}(k)$ and consequently M_i agreed with M_{a_j} on all $x \leq k$ let:

$$M_{g(i)}(n) = \begin{cases} j \text{ if } M_i(k+1) = M_{a_j}(k+1) \\ j+1 \text{ if } M_i(k+1) \neq M_{a_j}(k+1) \\ \text{diverge if } M_i \text{ diverges for any } x \leq k+1. \end{cases}$$

If, however, $M_i(k) \neq M_{a_j}(k)$, then start the comparison over with $M_{a_{j+1}}$ as follows:

$$M_{g(i)}(n) = \begin{cases} j+1 \text{ if } M_i(0) = M_{a_{j+1}}(0) \\ j+2 \text{ if } M_i(0) \neq M_{a_{j+1}}(0) \\ \text{diverge if } M_i(0) \text{ diverges} \end{cases}$$

Therefore $M_{g(i)}$ is total only if M_i is total and if for some $a_j \in A$, $M_i = M_{a_j}$ then for all $x: M_{g(i)}(x) \leq j$. Thus:

$$i \in \Omega R_t^\Phi \iff g(i) \in \Omega\underline{\text{Bound}}.$$

b) $\Omega\underline{\text{Bound}} \leq_1 \Omega R_t^\Phi$.

For any r.e. R_t^Φ there is a function b which majorizes the class [2]. Also, for time as a measure, every R_t^Φ contains at least one function and all its finite variants. Therefore, select some f in R_t^Φ so that all of its finite variants are also in R_t^Φ, and for any M_i define:

$$M_{g(i)}(x) = \begin{cases} f(x) \text{ if } M_i(x) \leq \max[M_i(0), \ldots, M_i(x-1)] \\ b(x)+1 \text{ if } M_i(x) > \max[M_i(0), \ldots, M_i(x-1)] \\ \text{diverge if any of } M_i(0), \ldots, M_i(x) \text{ diverge} \end{cases}$$

Therefore, $M_{g(i)}$ is total if M_i is total, and if M_i is bounded then $M_{g(i)}$ will be a finite variant of f and therefore a function in R_t^Φ.

An interesting property of time as a measure is indicated by this result. All of the index sets of the complexity classes fall into the same 1-degree; in fact, the 1-degree of $\Omega\underline{\text{Bound}}$. This means that all of the ΩR_t^Φ possess exactly the same properties. The following condition on measures is suggested by this fact.

<u>Definition</u>: Φ <u>conforms</u> (on ΩR_t^Φ) iff for all recursive t_1 and t_2,

$$\Omega R_{t_1}^\Phi \equiv \Omega R_{t_2}^\Phi.$$

This definition can be extended to Turing reducibility in the obvious manner. (Since all index sets are cylinders, m-conformity is the same as conformity.) An interesting fact is that all measures T-conform.

<u>Theorem 2.3</u>: Every Φ T-conforms (on ΩR_t^Φ).

<u>Proof</u>: The "oracle machine" $M^{\Omega\underline{\text{Equal}}}$ which has

$\Omega\underline{\text{Equal}} = \{\langle i,j\rangle \mid \forall x[M_i(x) = M_j(x)]\}$

(or all pairs of identical machines) written on its reference tape will be used in this proof. $\Omega\underline{\text{Equal}}$ is a well known Π_2-complete set, and therefore can decide membership problems in the Turing degree containing (b), (c), and (d) in Figure 1.

Also, the following two presentations are used:

$A = \{a_o, a_1, \ldots\}$ = presentation of R_t^Φ

$B = \{b_o, b_1, \ldots\}$ = presentation of

$$\mathcal{P} - R_t^\Phi$$

where $\mathcal{P}$ is the class of all partial recursive functions.
The set A used will be defined later as I_t^Φ, and is in Σ_2, while the set B can be r.e. [15]. Therefore, A and B can be listed by $M^{\underline{\Omega Equal}}$.

This machine operates (when given input i) by listing the sequence $a_o, b_o, a_1, b_1, \ldots$ and checking the pairs $\langle a_o, i\rangle$, $\langle b_o, i\rangle$, etc. against the pairs on the reference list.

Sooner or later some $\langle a_m, i\rangle$ or $\langle b_n, i\rangle$ will match a pair on the reference tape and then the machine halts and outputs:

$$M^{\underline{\Omega Equal}}(i) = \begin{cases} 1 & \text{if } \langle a_m, i\rangle \in \underline{\Omega Equal} \\ 0 & \text{if } \langle b_n, i\rangle \in \underline{\Omega Equal}\ . \end{cases}$$

Therefore $\Omega R_t^\Phi \leq_T \underline{\Omega Equal}$ and since $\underline{\Omega Equal} \equiv \Theta N \leq_1 \Omega R_t^\Phi$ (shown below), the result follows.

This result indicates that the general measures conform in rather a rough way. Unfortunately, Turing reducibility is fairly crude and allows sets with differing properties to be included in the same degree. On the other hand, 1-reducibility is much stricter and as was pointed out above, sets in the same 1-degree possess the same recursively invariant properties. However, general measures do not conform, and this fact is shown in the following sequence of results.

Fact: For any recursive function t and r.e. class of total functions $\underline{C}$, there is a measure Φ such that:

$$R_t^\Phi = \underline{C}\ .$$

Corollary: There are measures Φ which do not conform (on ΩR_t^Φ).

Proof: Immediate from the above fact and the existence of r.e. classes of total functions whose index sets are not in the same 1-degree.

2.4 Efficient Presentations of Complexity Classes

Another interesting set defined from measures is the set of algorithms which are t-computable; the efficient presentation for R_t^Φ.

Definition: The set of t-computable algorithms is defined:

$$I_t^\Phi = \{i \mid M_i \text{ is total and } \Phi_i(x) \leq t(x) \text{ almost everywhere}\}\ .$$

This set is contained in Σ_2 since its membership problem is easily written in Σ_2 form.

When tape and time are used as measures, the I_t^Φ for $t(x)$ = constant are recursive (this allows minimal growth rate [2]), while for increasing $t(x)$, I_t^Φ becomes Σ_2-complete (or $I_t^\Phi \equiv \Theta\underline{Finite}$). By slight modification these measures can be forced to conform on I_t^Φ. The changes are quite reasonable and involve making each machine read or copy its input.

Theorem 2.4: For M_i which copy their inputs, and Φ = tape length, Φ conforms (on I_t^Φ).

Sketch of Proof: The reduction of $\Theta\underline{Finite}$ to I_t^Φ is achieved by constructing a machine $M_{g(i)}$ which copies down its input, then simulated M_i upon this amount of tape. If M_i accepts anything new, then $M_{g(i)}$ exceeds $t(x)$ tape. Therefore, if M_i accepts a finite set, then $M_{g(i)}$ stays within the amount of tape the input takes up almost everywhere.

Since there are measures which have both recursive and Σ_2-complete I_t^Φ, even T-conformity is out of the question for general measures. The following result (shown without proof since the technique is similar to that of Theorem 2.1) shows that measures can be found where I_t^Φ is in various 1-degrees.

Lemma: For any set A and some infinite, recursive $B = \{b_o, b_1, \ldots\}$ the set:

$$C = \{b_i \mid i \in A\} \equiv_m A\ .$$

Theorem 2.5: For any recursive t and r.e. set A there is a measure Φ such that $I_t^\Phi \equiv_m A$.

An extension of the techniques used in Theorem 2.1 and 2.5 can be used to produce a measure where for every m-degree in $\Sigma_1 \cup \Pi_1$ there is some t such that I_t^Φ is in that m-degree. This gives a rather ugly, layered structure to the complexity classes.

2.5 Limits on Irregularities

Even though measures exist with irregularities or undesirable properties, these phenomena exist only in the complexity classes at the bottom of the hierarchy. An example of this is that as soon as the functions of finite support become t-computable (as they must sooner or later), then the complexity classes become r.e. [2]. As one would expect, all measures exhibit conformity above some point in the complexity hierarchy.

Definition: Φ conforms above t (on ΩR_t^Φ) iff for all recursive $s > t$: $\Omega R_s^\Phi \equiv \Omega R_t^\Phi$.

Theorem 2.6: For every Φ there is a recursive function t_o such that Φ conforms above t_o (on ΩR_t^Φ) .

Proof: The desired t_o is that such that $R_{t_o}^\Phi$ contains the functions of finite support. Then the proof proceeds like Theorem 2.2.

In order to show conformity to I_t^Φ a result due to Blum [1] is required.

Theorem 2.7 (Blum): For any measures Φ and $\hat{\Phi}$ there is a recursive function f such that for almost all x and all i : $\Phi_i(x) \leq f(x,\hat{\Phi}_i(x))$ and $\hat{\Phi}_i(x) \leq f(x,\Phi_i(x))$.

Theorem 2.8: For every Φ there is a recursive function t_o such that Φ conforms above t_o (on I_t^Φ).

Sketch of Proof: The required $t_o(x) = f(x,x)$ from Theorem 2.7. The result follow quite easily.

3. EVOLUTION OF A NEW AXIOM

From the evidence in the last section and in the literature it could be assumed that the original two axioms are too weak to characterize the natural measures of computation. Some new requirement must be added to the axioms in order to eliminate measures with undesirable properties.

In this section constraints will be placed upon measures to force conformity. This is done for the complexity hierarchies of recursive functions and primitive recursive functions. The theorems below will be stated without proof since they involve reductions which are similar to those exhibited above, or are rather straightforward.

3.1 Measures for Recursive Functions

These results are suggested by Theorem 2.2 and progress gradually towards conformity.

Theorem 3.1: If $\underline{C}$ is a non-trivial class of total functions, then $\Theta N \leq_1 \Omega \underline{C}$.

When paired with the T-conformity theorem (Theorem 2.3), this result provides maximal bounds for the location of ΩR_t^Φ in the arithmetical hierarchy. This is pictured in Figure 2 and represents where the ΩR_t^Φ are located for general measures.

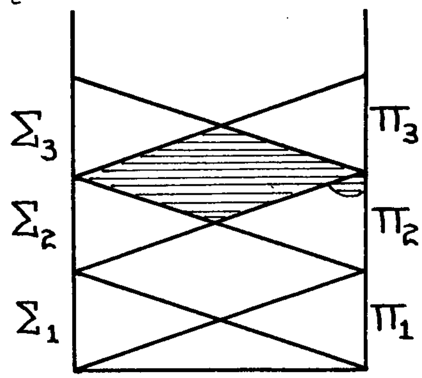

Figure 2.

The next step is to require that all complexity classes be r.e., which seems to be a reasonable restriction.

Theorem 3.2: If $\underline{C}$ is an r.e. class of total functions, then $\Omega \underline{C} \leq_1 \Omega\underline{\text{Bound}}$.

Combined with Theorem 3.1, this result restricts the location of ΩR_t^Φ to the shaded area in Figure 3. Conformity has not been achieved yet, so just making the complexity classes r.e. does not seem to be an adequate restriction.

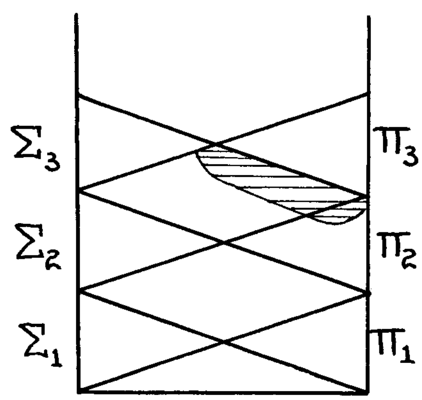

Figure 3.

Theorem 3.3: If $\underline{C}$ is an r.e. class of total functions containing all finite variations of some function, then $\Omega\underline{\text{Bound}} \leq_1 \Omega \underline{C}$.

When these classes are complexity classes, then containing finite variations implies r.e.-ness. At last ΩR_t^Φ has been confined to a single 1-degree as is shown in Figure 4. Of course, this degree contains all of the ΩR_t^Φ for tape and time as measures.

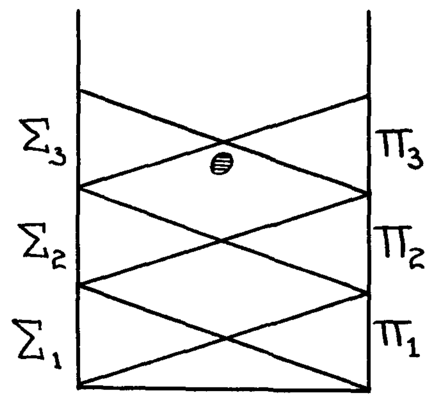

Figure 4.

3.2 Measures for Primitive Recursive Functions

Much of the work in automata theory has been concerned with the functions which are frequently computed; not with all of the computable functions. In this section the primitive recursive functions are denoted $p_0, p_1, \ldots$ and the step counting functions $P_0, P_1, \ldots$ are assigned to them. The complexity classes R_t^P are defined similarly to the R_t^Φ.

Several important index sets for classes of primitive recursive functions are defined:

a) ΩZero = {functions with zero in range}
= $\{i \mid \exists x[p_i(x) = 0]\}$

b) ΩEven = {even functions}
= $\{i \mid \forall x[p_i(x)$ is even$]\}$

c) ΩZero $\cap$ ΩEven

d) ΩFinSup = {functions of finite support}
$\{i \mid \exists k \forall x[x > k \Rightarrow p_i(x) = 0]\}$

and the location of each of these sets is indicated in Figure 5.

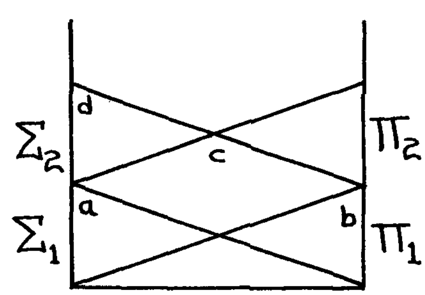

Figure 5.

The original Blum axioms for measures do not translate freely to the primitive recursive functions and so a series of restrictions will be placed on the measures. Several facts are in order at this point.

Facts: a) There is no non-trivial index set which are finite or cofinite.

b) ΩR_t^P is a Σ_2 set .

Initially the measures considered are required to be primitive recursive. This is reasonable since a function's cost of computation should be the same kind of function as the one being computed. This permits ΩR_t^P to fall in the shaded area of Figure 6. Unfortunately, this restriction does not necessarily even produce a hierarchy, since all functions can be assigned to the same complexity class.

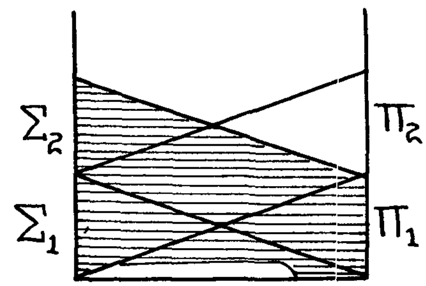

Figure 6.

Making a relationship between a function and its cost mandatory is the next restriction to be placed on measures.

Definition: f is a speed limit for P iff for all i , $f(x) \geq p_i(x)/P_i(x)$ almost everywhere

Theorem 3.4: If P has a primitive recursive speed limit then ΩEven $\leq_1 \Omega R_t^P$.

This speed limit can be measured in units of digits printed per cost. For time and tape on Turing machines the speed limit is between log(x) and 1 , depending on how they are defined.

This new restriction allows ΩR_t^P to range within the shaded area indicated on Figure 7. At last a hierarchy is assured, but the complexity classes do not have to be r.e.

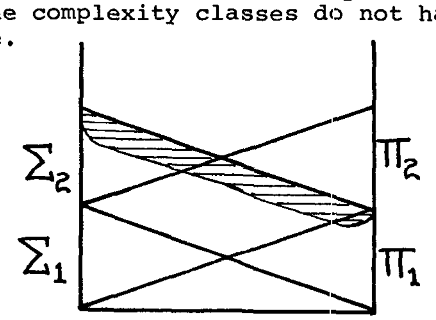

Figure 7.

If instead of speed limits, a restriction such as: for some $f \in R_t^P$ and $g \notin R_t^P$, if h is defined as:

$$h(x) = \begin{cases} g(x) & \text{if } x < n \\ f(x) & \text{if } x \geq n \end{cases}$$

for some n then this implies that $h \in R_t^P$ then $\Omega\underline{\text{Zero}} \leq_1 \Omega R_t^P$ by the obvious reduction. This forces the index sets for complexity classes to be in the area designated in Figure 8. No hierarchy is assumed, and R_t^P does not have to be r.e. as of yet.

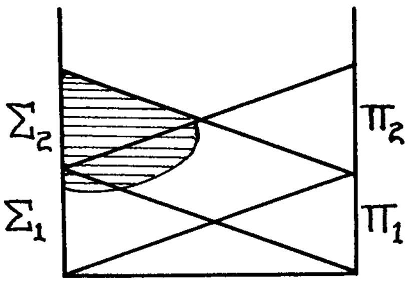

Figure 8.

Figure 9 indicates the result of combining the three restrictions mentioned above. This follows from the reduction $\Omega\underline{\text{Even}} \cap \Omega\underline{\text{Zero}} \leq_1 \Omega R_t^P$. The complexity classes are not all r.e. yet, but naturally there is a hierarchy.

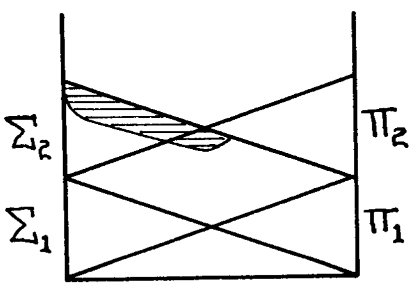

Figure 9.

Conformity is finally achieved in the next result

Theorem 3.5: If P has a primitive recursive speed limit and R_t^P contains all finite variants of some function, then $\Omega\underline{\text{FinSup}} \leq_1 \Omega R_t^P$.

The speed limit restriction is necessary to have a hierarchy. At last conformity has been provided for the primitive recursive functions, as is indicated in Figure 10.

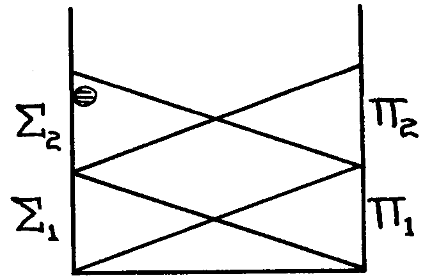

Figure 10.

In **fact**, the complexity classes are r.e. also.

4. CONCLUSION AND OPEN PROBLEMS

The major problem with general measures is that the desirable properties which some measures possess are not found in all measures. Therefore, it is desirable to isolate those measures which are natural and do not have any pathological properties.

One way to do this is to require that the complexity classes defined from measures all have identical properties. This was the rationale behind the definition of conformity, and therefore conformity seems to be a reasonable criterion for measures or any other axiom system.

From the phenomena exhibited previously, it would seem that a new axiom is needed. If this axiom were that every complexity class contains all the finite variants of at least one function, then conformity will be achieved. However, it may be possible to achieve conformity with some other axiom which is more subtle than the one proposed here.

Whenever pathological problems exist in complexity hierarchies, it has been shown that they exist only in the lower levels of the hierarchy. This means that almost all of the complexity classes for any measure belong to the same 1-degree, and that the measure conforms "almost everywhere". Conditions that occur in all but a finite number of places are accepted in automata theory as being desirable in most cases. In fact, the definition of the complexity classes R_t^Φ contains an "almost everywhere" clause.

But, in complexity hierarchies, the functions which are easiest to compute, and that are computed most often, occur at the bottom. These very functions are the ones computed in "real life" and therefore are quite important. Facts about their complexity should be meaningful, and so measures used should not have any pathological properties, even for a few classes at the bottom of the hierarchy.

Some open problems and areas for further study are as follows.

a) Conformity on I_t^{Φ} and I_t^{P} should be studied by placing restraints on the axioms for measures.

b) Define the class:

$$\underline{\text{Best}} = \{f \mid \text{there is a "fastest" program for } f\}$$
$$= \{f \mid \exists i [M_i = f \wedge \forall j (M_i = M_j \Rightarrow \Phi_i \leq \Phi_j)]\}$$

and study ΩBest with respect to conformity.

c) Speed limits, defined in the last section, should be investigated with regard to their effect on conformity and other properties of measures.

d) Possibly some properties of classes of recursive functions (along the lines of those in Dekker and Myhill) could be formulated which would have some significance when applied to R_t^{Φ}.

e) The criterion of conformity might be profitably applied to axiom systems in other areas of automata theory.

Acknowledgements

The author wishes to thank Professors J. Hartmanis and R. Constable for their patient encouragement and assistance during the formulation of these results.

REFERENCES

1. Blum, M. A Machine Independent Theory of the Complexity of Recursive Functions, JACM (1967), pp. 322-336.

2. Borodin, A. Complexity Classes of Recursive Functions and the Existence of Complexity Gaps, ACM Sym. on Theory of Comp., (May 1969), pp. 67-78.

3. _______, Constable, R., and Hopcroft, J. Dense and Non-Dense Families of Complexity Classes, Tenth Ann. Sym. on Switching and Automata Theory, IEEE (October 1969).

4. Constable, R. The Operator Gap, Tenth Ann. Sym. on Switching and Automata Theory, IEEE (October 1969).

5. Dekker, J.C.E. and Myhill, J. Some Theorems on Classes of Recursively Enumerable Sets, Trans. AMS, Vol. 89 (1958), pp. 25-59.

6. Grzegorczyk, A. Some Classes of Recursive Functions, Rozprawy Matematiyczne, 1953, pp. 1-45.

7. Hartmanis, J. and Stearns, R. On the Computational Complexity of Algorithms, Trans. AMS, Vol. 117 (1965), pp. 285-306.

8. Lewis, F.D. The Classification of Unsolvable Problems in Automata Theory, Cornell Tech. Report 70-49 (January 1970).

9. McCreight, E. and Meyer, A. Classes of Computable Functions Defined by Bounds on Computation, ACM Sym. on Theory of Comp. (May 1969), pp. 79-88).

10. Meyer, A. and Fischer P. On Computational Speedup, Ninth Ann. Sym. on Switching and Automata Theory, IEEE (October 1968), pp. 351-355.

11. Myhill, J. Linear Bounded Automata, University of Pennsylvania, Report No. 60-22 (June 1960).

12. ______. Creative Sets, Zeit fur Math. Logik und Grund der Math., Vol. 1 (1955), pp. 97-108.

13. Péter, Roza. Recursive Functions, 3d Edition, New York, 1967.

14. Ritchie, R. Classes of Predictably Computable Functions, Trans.AMS, Vol. 106 (1963), pp. 139-173.

15. Robertson, E.L. and Landweber, L.H. On Recursive Properties of Complexity Classes, these proceedings.

16. Rogers, H.G. Godel Numberings of Partial Recursive Functions, JSL, Vol. 23 (1958), pp. 331-341.

17. ______. Theory of Recursive Functions and Effective Computability, McGraw Hill, New York 1967.

18. Stearns, R., Hartmanis, J. and Lewis, P., II. Hierarchies of Memory Limited Computations, 1965 IEEE Conf. Rec. on Switching Theory and Log. Design, pp. 179-190.

19. Young, P. Toward a Theory of Enumerations, JACM, Vol. 16 (1969), pp. 328-347.