14th Annual Symposium on Automata and Switching Theory,
OCT 1973,
Iowa City

ON THE SIZE OF SETS OF COMPUTABLE FUNCTIONS[†]

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

Summary

We investigate the size of sets of computable functions using category-theoretic methods (in the sense of the Baire Category theorem). Constructive definitions of nowhere dense and meagre set are given and applied to several problems. In particular we apply it to subrecursive degree structures and to a comparison of the power of deterministic and nondeterministic time bounded oracle machines.[1]

Introduction

In this paper we investigate the size of sets of computable functions.

Let R be the set of total recursive functions. In the theory of computing many subsets of R are shown to be nontrivial; i.e. they are nonempty and have nonempty complement. So far nothing has been said about the size of those sets. For example we know that there are honest and dishonest functions. In this paper we give a precise meaning to the statement: "Only a few functions are honest".

There are two possible approaches to this problem. One uses the notion of meagre set (set of first category), the other one uses measure theory. We will explore the first approach in this paper; some results obtained using the second approach are mentioned in section VI.

In section III we give the basic definitions and prove some important properties of the class of meagre sets: closure under subset and effective union and the Baire Category Theorem. Even readers familiar with the category methods should not skip this section since our definitions differ considerably from the standard definitions. All our definitions are constructive.

In section IV we give some examples of meagre sets. We show that recursively presentable classes, complexity classes and classes of honest functions are meagre. Further we compare determininistic and nondeterministic polynomial time bounded oracle machines.[1]

In section V we apply the theory of subrecursive degree structures. We show that most functions are incomparable to any given function and that maximal antichains are not effectively listable. As a corollary we derive a result about "helping" which was originally proved by Nancy Lynch.[3]

Section VI is a brief conclusion.

[†]This research was supported by NSF Grant GJ-579.

Notation

N : the set of natural numbers, $N=\{0,1,2,\dots\}$

P : the set of partial recursive functions

$\{\phi_i\}_{i=0}^{\infty}$ is an acceptable indexing of P

R : the set of total recursive functions

f,g : elements of R

T : the set of finite functions

s,t : elements of T

$\ell h(\ )$: if $t \in T$ and domain $t = \{0,1,\dots,n-1\}$ then $\ell h(t) = n$

t^0 : if $t \in T$ then $t^0 \in R$

$$t^0(x) = \begin{cases} t(x) & \text{if } x < \ell h(t) \\ 0 & \text{otherwise} \end{cases}$$

$\sqsubseteq$: for $t_1,t_2 \in T$, $f \in R$ $t_1 \sqsubseteq t_2$ ($t_1 \sqsubseteq f$) means dom $t_1 \subseteq$ dom t_2 and $t_2|\text{dom } t_1 = t_1$ ($f|\text{dom } t_1 = t_1$)

$\sqsubset$: $t_1,t_2 \in T$; $t_1 \sqsubset t_2$ means $t_1 \sqsubseteq t_2$ and dom $t_1 \subset$ dom t_2

v,w: denote computable functions from $N^i \times T$ to T ($i = 0,1,2$)

$<,>,p_1,p_2$: $<,>$ denotes a computable pairing function $<,> : N^2 \to N$. p_1,p_2 denote its computable inverses.

Basic Definitions and Basic Theorems

In this section we will define the basic notions of nowhere dense and meagre set. At the end of the section we will prove the Baire Cateogry Theorem.

In general topology the notion of meagre set is standard. (See [6]). Unfortunately the standard definition is not useful for our purpose. We indicate two reasons for that: Topology usually deals with uncountable domains like the set of real numbers or the set of all functions which map integers into integers. Our domain - the set of total recursive functions - is countable. Furthermore nonconstructive principles are usually employed in exitence proofs (e.g. Baire Category Theorem). We have to make sure that all constructions yield recursive functions and can therefore use only constructive principles.

We start by defining a neighborhood structure on the set of total recursive functions.

(3.1) Definition: Let $t \in T$; i.e. t is a finite function. Then t determines the basic open neighborhood.

$$U_t = \{f \in R;\ t \sqsubseteq f\}$$

($t \sqsubseteq f$ stands for f extends t)

Usually a nowhere dense set is defined as follows: A is nowhere dense if for all non-empty open sets U there is a non-empty open set V such that $V \subseteq U$ and $V \cap A = \emptyset$. Using Skolem-functions we can rewrite this definition: A is nowhere dense if there is a function F mapping nonempty open sets into nonempty open subsets such that for all open nonempty sets U $F(U) \cap A = \emptyset$. We can assume w.l.o.g. that F is defined only on basic open neighborhoods and that F maps basic open neighborhoods into basic open neighborhoods. This observation together with the fact that basic open neighborhoods are determined by elements of T leads to a constructive definition of nowhere dense set.

(3.2) Definition: Let $A \subseteq R$, $V: T \to T$ be computable. The pair (A,v) is called (effectively) nowhere dense if

(a) for all $t \in T$ $t \sqsubseteq v(t)$

(b) there is an $n \in N$ such that for all $t \in T$

$$\ell h(t) > n \Rightarrow U_{v(t)} \cap A = \emptyset$$

v is called a verifying function for A.

Two remarks might be useful at this point. The definition (3.2) is not the only conceivable definition of nowhere dense set. We chose this definition because it allows simple intuitive proofs. A nowhere dense set is a pair (A,v) where A is a subset of R and v is a computable function from T to T. We will sometimes say that $A \subset R$ is nowhere dense; the verifying function v is understood in those cases. The reader should keep in mind that in order to prove $A \subset R$ nowhere dense we have to exhibit a computable verifying function for it.

The reader can view R as the infinite splitting tree which is pruned by all non recursively enumerable paths. A computable function $f \in R$ corresponds to a path through the tree, a finite function $t \in T$ corresponds to a node in the tree. A basic open neighborhood U_t consists of all paths which go through the subtree determined by t. A subset $A \subset R$ is a set of paths. It is nowhere dense if there is an effective method for going from subtrees to smaller subtrees such that the smaller subtree is disjoint from A; i.e. the paths in A do not go through the smaller subtree.

Meagre sets are usually defined as countable unions of nowhere dense sets. We will define them as effective countable unions of nowhere dense sets. By this we mean that we can put together the verifying functions of the nowhere dense components to a single computable function. However the set union in the following definition is meant in the classical sense.

(3.3) Definition: Let M be a subset of R, $\{M_i\}_{i=0}^{\infty}$ be a sequence of subsets of R, v be a computable function from N x T to T. The triple (M, $\{M_i\}_{i=0}^{\infty}$, v) is called a meagre set if

(a) $M = \bigcup_{i=0}^{\infty} M_i$

(b) (M_i, v_i) is nowhere dense for all i.

Again we will sometimes say that a subset M of R is meagre; the decomposition $\{M_i\}_i$ and the verifying function v are understood in these cases.

It is worth noting that nowhere dense sets are meagre.

We will now prove the basic properties of meagre sets: closure under subset and effective union and the Baire Category theorem.

(3.4) Theorem (Closure under subset): If M_1 is meagre and $M_2 \subseteq M_1$ then M_2 is meagre.

Proof: obvious.

(3.5) Theorem (Closure under effective union): Let v: $N^2 \times T \to T$ be a computable function, $\{M_i\}_{i=0}^{\infty}$ be a sequence of meagre sets such that v_i can serve as a verifying function for M_i. Then $M = \bigcup_i M_i$ is meagre.

Proof: There are decompositions $M_i = \bigcup_{j=0}^{\infty} M_{i,j}$ such that $(M_{i,j}, \lambda t[v(i,j,t)])$ is meagre for all i,j. Set

$$N_k = M_{p_1(k),p_2(k)}$$

and

$$w(k,t) = v(p_1(k),p_2(k),t).$$

Then $M = \bigcup_k N_k$ and for all k the pairs (N_k, w_k) is nowhere dense. Hence M is a meagre set. Q.E.D.

Theorem (3.5) implies in particular that the class of meagre sets is closed under finite union.

(3.7) Theorem (Baire Category Theorem): Let U_t be a basic open neighborhood and let M be a meagre set. Then $U_t - M \neq \emptyset$.

Proof: Let $\{M_i\}_i$ be a decomposition of M and v: $N \times T \to T$ be a computable function such that (M_i, v_i) is nowhere dense for all i. We will construct an element f of $U_t - M$. f is constructred in stages.

stage 0: $t_0 \leftarrow t$

stage m: $t_{m+1} \leftarrow v(p_1(m), t_m)$

Since by definition of $v, t_m \sqsubset t_{m+1}, f = \bigcup_m t_m$ is a total function. It is obvious that f is recursive and that $f \in U_t$. It remains to show that $f \notin M$. Assume otherwise. Then $f \in M_i$ for some i. Since v_i is a verifying function for M_i there is an $n \in N$ such that

$$\ell h(t) > n \Rightarrow U_{v_i(t)} \cap M_i = \emptyset.$$

Let m be such that $\ell h(t_m) > n$ and $p_1(m) = i$. Such a m exists since $t_m \sqsupset t_{m+1}$ for all m. Then $f \sqsupset t_{m+1} = v_i(t_m)$ and hence $f \notin M_i$.

Q.E.D.

The Baire Category Theorem implies that open sets are non-meagre. In particular R is non-meagre.

First examples of meagre sets

In this section we explore the definitions in several directions. First we consider the most simple sets of recursive functions: finite sets and recursively presentable sets. Both turn out to be meagre. Then we investigate complexity classes and sets of honest functions. Finally we compare deterministic and nondeterministic polynomial time bounded oracle-machines. We show that for a co-meagre set of oracles nondeterministic machines allow us to compute more functions.

The most simple sets are the singleton sets. They are nowhere dense.

(4.1) Lemma (Singleton sets are nowhere dense): Let $g \in R$. Then $\{g\}$ is nowhere dense and an index of a verifying function can be uniformly found from an index for g.

Proof: $v(t) = t \cup \langle \ell h(t), g(\ell h(t)) + 1\rangle$ is a verifying function for $\{g\}$. For all $t \in T$ $v(t) \not\sqsubseteq g$ and hence $U_{v(t)} \cap \{g\} = \emptyset$. The second claim is obvious. Q.E.D.

(4.2) Theorem (Recursively presentable sets of recursive functions are meagre): If $f \in R$ is such that for all i $f(i)$ is the index of a recursive function then $\{\phi_{f(i)}(\ ); i \in N\}$ is meagre.

Proof: By lemma (4.1) the singleton sets $\{\phi_{f(i)}(\ )\}$ are nowhere dense and therefore meagre. Furthermore the index of a verifying function is uniform in $f(i)$. Therefore we can apply the theorem about closure under effective union (theorem (3.5)) and infer the meagreness of $\{\phi_{f(i)}(\ ); i \in N\}$. Q.E.D.

Many classes of recursive functions are defined by restrictions on the syntax of the recursion equations which are used in the definition of members of the class. (e.g. class of primitive recursive functions, class of elementary recursive functions,...) All these classes are recursively presentable.

(4.3) Corollary: The class of primitive recursive functions is meagre.

We now turn to the theory of computational complexity. Complexity classes are one of its basic constructs. Let b be a recursive function. b determines the complexity class $R_b = \{f \in R$; there is an $i \in N$ such that

$$\phi_i(\ ) = f \text{ and}$$
$$\Phi_i(x) \le b(x) \text{ for almost all } x\}$$

Here Φ_i stands for the step counting algorithm of ϕ_i in the sense of Blum. For sufficiently large b R_b is recursively presentable and hence meagre. We give a direct proof for this fact.

(4.4) Theorem (Complexity Classes are meagre): Let $b \in R$; then $R_b = \{f \in R$; there is an $i \in N$ such that

$\phi_i(\ ) = f$ and $\Phi_i(x) \le b(x)$ for almost all $x\}$ is a meagre set.

Proof: In this proof we use for the first time a nontrivial decomposition. Let

$$M_k = \begin{cases} \{\phi_{p_1(k)}(\ )\} & \text{if } \phi_{p_1(k)}(\ ) \text{ is total and} \\ & \Phi_{p_1(k)}(x) \le b(x) \text{ for} \\ & x > p_2(k) \\ \emptyset & \text{otherwise.} \end{cases}$$

Obviously $R_b = \bigcup_k M_k$. It remains to exhibit verifying functions for the M_k's. The following program computes a verifying function for M_k.

$v_k(t)$: if $\ell n(t) < p_2(k)$

then ¢ $p_2(k)$ serves as the n in definition (3.2); the output has to extend the input ¢ $t \cup \langle \ell n(t), 0\rangle$

else ¢ either $M_k = \emptyset$ and the output just has to extend the input or $M_k \ne \emptyset$ and $\Phi_{p_1(k)}(\ell n(t)) \le b(\ell n(t))$ ¢

if $\Phi_{p_1(k)}(\ell n(t)) \le b(\ell n(t))$

then ¢ $M_k \ne \emptyset$ ¢

$t \cup \langle \ell n(t), \phi_{p_1(k)}(\ell n(t))+1\rangle$

else ¢ $M_k = \emptyset$ ¢

$t \cup \langle \ell n(t), 0\rangle$

The comments should suffice to convince the reader that v_k computes a verifying function form M_k. The index of v_k is uniform in k. Therefore R_b is meagre. Q.E.D.

We now consider sets of honest functions. Let $r \in R^2$. The set of r-honest functions is given by

$$\{f \in R;\ \text{there is an } i \text{ such that } \phi_i(\ ) = f \text{ and } \Phi_i(x) \le r(x,\phi_i(x)) \text{ for almost all } x\}$$

Note that the set of r-honest functions is not recursively presentable.

(4.5) Theorem: For every $r \in R^2$: The set of r-honest functions is meagre.

Proof: Let

$$M_k = \begin{cases} \{\phi_{p_1(k)}(\ )\} & \text{if } \Phi_{p_1(k)}(x) \le r(x, \phi_{p_1(k)}(x)) \\ & \text{for } x \ge p_2(k) \\ \phi & \text{otherwise} \end{cases}$$

Obviously $\bigcup_k M_k$ = set of r-honest functions.

The following program computes a verifying function for M_k.

$v_k(t)$: if $\ell n(t) < p_2(k)$

then ¢ $p_2(k)$ serves as the n in definition (3.2); in the then case the output just has to extend the input ¢

$t \cup <\ell n(t), 0$

else ¢ $\ell n(t) \ge p_2(k)$; if $M_k \ne \emptyset$ then by definition of M_k

$\Phi_{p_1(k)}(\ell n(t)) \le r(\ell n(t), \phi_{p_1(k)}(\ell n(t)))$;

we 'guess' that

$\phi_{p_1(k)}(\ell n(t)) = 0$ ¢

if $\Phi_{p_1(k)}(\ell n(t)) \le r(\ell n(t), 0)$

then ¢ we know that $\phi_{p_1(k)}(\ell n(t))$ converges ¢

$t \cup <\ell n(t), \phi_{p_1(k)}(\ell n(t)) + 1>$

else ¢ either $M_k = \emptyset$ or $\phi_{p_1(k)}(\ell n(t)) \ne 0$ ¢

$t \cup <\ell n(t), 0>$

The comments should suffice to convince the reader that v_k computes a verifying function for M_k. An index for v_k can be uniformly computed. Therefore the set of r-honest functions is meagre. Q.E.D.

In the theory of subrecursive degrees a different definition of honest function is usually used. For example Machtey ([4]) calls a function f primitive recursive honest if it can be computed by a program whose running time is bounded by a function primitive recursive in f. A proof similar to the proof of theorem (5.4) shows that the class of primitive recursive honest functions is meagre.

Now we turn to a comparison of deterministic and nondeterministic polynomial time bounded oracle-machines. Let $f \in R$. Dpol(f) denotes the set of all 0-1 valued functions which can be deterministically computed from f in polynomial time. Analogously NDpol (f) is defined. Oracle calls count as a single step. We also suppose that addition and multiplication are basic instructions. The idea for the proof of (4.6) is due to Ted Baker ([1]).

(4.6) Theorem: The set $\{f;\ \text{Dpol}(f) \ne \text{NDpol}(f)\}$ is co-meagre.[1]

Proof: We have to show that

$$\{f;\ \text{Dpol}(f) = \text{NDpol}(f)\}$$

is co-meagre. Let $R_0[\ ]$ be the following nondeterministic polynomial time bounded operator.

$$R_0[f](n) = \begin{cases} 1 & \text{there is a } x \text{ such that } 2^n \le x < 2^{n+1} \text{ and } f(x) \ne 0 \\ 0 & \text{otherwise.} \end{cases}$$

We show that the larger set

$$M = \{f;\ R_0[f](\ ) \in \text{Dpol}(f)\}$$

is meagre. By closure under subset (theorem (3.4)) this implies the theorem.

Let $\{Pol_i[\ ]\}_{i=0}^{\infty}$ be an enumeration of the deterministic polynomial time bounded oracle machines. Then

$$M = \{f;\ R_0[f](\ ) \in \text{Dpol}(f)\} = \bigcup_k \{f;\ R_0[f](\ ) = Pol_k[f](\ )\} = \bigcup_k M_k$$

W.l.o.g. we can assume that $\{\Pi_i\}_{i=0}^{\infty}$ is a list of polynomials such that Π_i is an upper bound for the running time of $Pol_i[\ ]$. If Π_i is a polynomial then there is an integer n(i) such that $\Pi_i(x) < 2^x$ for $x \ge n(i)$. n(i) will serve as the n in definition (3.2). If $Pol_i[f](x)$ converges in k steps at most k oracle calls can occur during the computation. We give now a program for the verifying function v_k of M_k.

$v_k(t)$: $n \leftarrow \mu x\ [2^x \ge \ell h(t)]$

¢ since t is a finite initial segment t is undefined on the interval $[2^n, 2^n+1, \ldots, 2^{n+1}, \ldots, 2^{n+1}-1]$. The length of this inverval is 2^n ¢

¢ if t is a finite function t^0 denotes the following total function:

$$t^0(x) = \begin{cases} t(x) & \text{if } x < \ell n(t) \\ 0 & \text{otherwise} \end{cases}$$ ¢

<u>if</u> $Po\ell_k[t^0](n)$ halts in less than 2^n steps ¢ since Π_k is a bound on the running-time of $Po\ell_k[\ ]$ this occurs for sufficiently long t ¢

<u>then</u>

c ← set of all arguments of oracle calls in the computation of $Po\ell_k[t^0](n)$

¢ by the remark preceeding the program $[2^n,\ldots,2^{n+1}-1]-c \neq \emptyset$, ¢

m ← some element of $[2^n,\ldots,2^{n+1}-1]-c$

p ← $\max\{c, 2^{n+1}-1\}$

¢ the output is the following finite function t' ¢

domain t' = {0,...,p}

$$t'(x) = \begin{cases} t(y) & \text{for } y < \ell n(t) \\ 1 & \text{if } y = m \text{ and } Po\ell_k[t^0](n) = 0 \\ 0 & \text{otherwise} \end{cases}$$

¢ the set $[2^n,\ldots,2^{n+1}-1] \cup c$ belongs to the domain of t'; t' coincides with t^0 on c and therefore $Po\ell_k[t'](n)$ $Po\ell_k[t'](n) = Po\ell_k[t^0](n)$; but by definition of t' $R_0[t'](n) \neq Po\ell_k[t^0](n)$ ¢

<u>else</u> $t \cup \langle \ell n(t), 0\rangle$

Again we hope that the comments are sufficient to convince the reader from the correctness of v_k. Q.E.D.

Application to Subrecursive Degree Structures

In this section we study subrecursive degree structures. The results will be equally valid for all subrecursive degree-structures which were investigated so far (e.g. primitive recursive degrees ([4]), elementary degrees ([5])). The degree-structure is usually induced by a recursively presentable list $\{Op_i\}_{i=0}^{\infty}$ of total recursive operators; e.g. the set of primitive recursive operators. They define a reducibility relation on R.

$f \leq g$ iff $f = Op_i[g](\ )$ for some i.

The relation $\leq$ is a partial ordering if the class of operators is closed under composition. We will not make this assumption. Dropping closure under composition does not change the proofs; however it allows us to state corollary (5.4). Let op = $\{f \in R;\ f = Op_i[t^0](\ )$ for some $i \in N$ and $t^0 \in T\}$. op is recursively presentable and hence a proper subset of R. If $\{Op_i[\ ]\}$ is the set of primitive recursive operators then op is the set of primitive recursive functions; it is the minimum degree in the degree structure induced by the reducibility ordering 'primitive recursive in'.

For $f \in R$ we consider the sets $A_f = \{g \in R;\ g \leq f\}$ and $B_f = \{g; f \leq g\}$. We will show that A_f is meagre for all $f \in R$ and that B_f is meagre for $f \in R$ - op. Conversely if $f \in R$ - op then the set of functions which are incomparable with f is comeagre. If $\leq$ is a partial ordering this suggests that the ordering is flat and wide.

(5.1) <u>Theorem</u>: For all $f \in R$ $A_f = \{g: g \leq f\}$ is meagre.

<u>Proof</u>: $A_f = \{g;\ g = Op_i[f](\ )$ for some i}. Therefore A_f is r.e. presentable. Theorem (4.2) implies that A_f is meagre. Q.E.D.

(5.2) <u>Theorem</u>: For all $f \in R$ - op $B_f = \{g;\ f \leq g\}$ is meagre.

<u>Proof</u>: We can decompose B_f in the following manner:

$$B_f = \bigcup_{i=0}^{\infty} \{g;\ f = Op_i[g](\ )\} = \bigcup_i M_i.$$

We give a program for a verifying function of M_i.

```
v_i(t) : t' ← t ;
         do while Op_i[t']( ) ⊑ f
            t' ← t' U <ℓn(t'),0>
         end;
         t'
```

We only have to show that v_i is a total function. Assume otherwise. Then there is a $t \in T$ such that $Op_i[t^0](\ ) \sqsubseteq f$. Since Op_i is a total recursive operator $f = Op_i[t^0](\ )$ and therefore $f \in$ op #.

(5.3) <u>Corollary</u>: For all $f \in R$ - op. The set of functions incomparable to f is comeagre.

<u>Proof</u>: $g \in R$ is incomparable to f if neither $g \leq f$ nor $f \leq g$; i.e. $f \in R - (A_f \cup B_f)$. But $A_f \cup B_f$ is meagre by (5.1), (5.2) and (3.5). Q.E.D.

We will now give an interpretation of these theorems for which $\leq$ is not a partial ordering. Nancy Lynch considered in her thesis ([3]) the notion "helping". Let $\{\phi_i[\ ]\}$ be an acceptable indexing of the relative algorithms. If σ denotes the function which is identical to 0 then $\{\phi_i[\sigma]\}$ is an acceptable indexing of the computable functions. $\mathrm{Comp}^h f > b$ i.o. stands for: if $\phi_i[h](\ ) = f$ then $\Phi_i[h](x) > b(x)$ for infinitely many x. Nancy Lynch derived the following result. (Theorem (6.3) of [3]).

(5.4) Corollary: There exists $g \in R_2$ with the following property: For any $b \in R_1$ and any recursive function f with

$$\mathrm{comp}\ f > g \circ b \text{ i.o.}$$

{$g \circ b$ stands for $\lambda x[g(x,b(x))]$} there exists arbitrarily complex recursive functions h with:

$$\mathrm{comp}^h f > b \quad \text{i.o.}$$

Proof: We prove the stronger result: There exists $g \in R_2$ such that: if $\mathrm{comp}\ g > f \circ b$ i.o. then $\{h;\ \mathrm{comp}^h f \leq b \text{ a.e.}\}$ is meagre. In theorem (4.4) we have shown that complexity classes are meagre. By theorem (3.5) the union of these two sets is meagre; the Baire category theorem (theorem (3.6)) establishes the claim.

We take as the class of operators $\{Op_i\}_{i=0}^{\infty}$ the set of operators which can be computed within cost b a.e. The list $\{Op_i\}$ comes from the list $\{\phi_i[\ ]\}$ by attaching a clock for b. Then $\mathrm{comp}^h f \leq b$ a.e. implies $f \leq h$. It is a well-known fact about simulation that there exists a function $k_1 \in R_1$ such that $Op_i = \phi_{k_1(i)}[\ ]$ and $\Phi_{k_1(i)}[\ ] \leq c \circ b$ a.e. for a fixed computable function $c \in R_1$. Let $\{t_j\}$ be an enumeration of T. There is a function $k_2 \in R_2$ such that

$$\phi_{k_2(i,j)}[\sigma](\ ) = \phi_i[h_j^0](\ )$$

Set

$$g'(x,y) = \max_{i,j \leq x} \{\Phi_{k_2(i,j)}[\sigma](x);\ \Phi_i[h_j^0](x) \leq y\}$$

and $g(x,y) = g'(x,c(x,y))$. Then $op \subseteq \{f;\ \mathrm{comp}\ f \leq g \circ b \text{ a.e.}\}$. Let $f \in op$. Then for some i and j $f = Op_i[h_j^0](\ )$. Therefore

$$\mathrm{comp}\ f \leq \Phi_{k_2(k_1(i),j)}[\sigma](\ )$$

$$\leq g'(x, \Phi_{k_1(i)}[h_j^0](x)) \text{ a.e. } (x)$$

$$\leq g'(x, c(x,b(x)) \text{ a.e. } (x)$$

$$\leq g(x, b(x)) = g \circ b(x) \quad \text{a.e. } (x)$$

The claim is now an immediate consequence of theorem (5.2). Q.E.D.

We mentioned at the beginning of the section that the partial ordering $\leq$ (if it is one) is wide and flat. The next theorem supports this view. An antichain is a set of pairwise incomparable functions.

(5.5) Theorem: Maximal antichains are not recursively presentable.

Proof: Assume otherwise. Then there is a maximal recursively presentable antichain c. Since c is maximal $\bigcup_{f \in c} (A_f \cup B_f) = R$ and since c is recursively presentable $\bigcup_{f \in c} (A_f \cup B_f)$ is a meagre set. But the Baire category theorem tells us that R is not meagre. Q.E.D.

(5.6) Corollary: There are antichains which are not recursively presentable.

Proof: Zorn's lemma implies the existence of maximal antichains. The claim follows from theorem (5.5). Q.E.D.

Finally we investigate minimal degrees. For the next theorem we restrict ourselves to reducibility orderings. The reader should interpret $\leq$ as 'primitive recursive in' or 'elementary recursive in'. The classes of operators which induce these relations possess the closure properties which are required in the proof. In fact we only use that which we can compute the join of two functions and that we can split a given function f into two parts f_1 and f_2 such that $f = f_1$ join f_2. $f \in R - op$ is called minimal if $g \leq f$ implies either $f \leq g$ or $g \in op$.

(5.7) Theorem: The set of minimal functions is meagre.

Proof: Let $Op_1[f](n) = f(2n)$ and $Op_2[f](n) = f(2n+1)$. If f is minimal then

either $Op_1[f](\ ) \in op$ and $f \leq Op_2[f](\ )$

or $\quad f \leq Op_1[f](\ )$ and $Op_2[f](\ ) \in op$

or $\quad f \leq Op_1[f](\ )$ and $f \leq Op_2[f](\ )$

If $Op_1[f](\ ) \in op$ and $Op_2[f](\ ) \in op$ then $f \in op$; but f is supposed to be in R - op.

By closure under subset it is sufficient to show that the larger set

$$\{f;\ f \leq Op_1[f](\ )\} \cup \{f;\ f \leq Op_2[f](\ )\}$$

is meagre. Let

$$M_k = \{f;\ f = Op_k[Op_i[f](\ )](\ )\}.$$

Certainly $M = \bigcup_k M_k$. The following program computes a verifying function for M_k.

$v_k(t)$: $x \leftarrow$ the least y which is odd and not in the domain of t ;

$\quad z \leftarrow Op_k[Op_1[t^0](\ )](x)$;

¢ note that during this computation arguments of function calls are even ¢

m ← max(x, set of arguments of function calls in this computation)

¢ the output is a finite function t' such that ¢

dom t' = {0,...,m}

$$t'(y) = \begin{cases} t(y) & \text{for } y < \ell n(t) \\ 1 & \text{for } y=x \text{ and } z=0 \\ \sigma & \text{otherwise.} \end{cases}$$

It is obvious that

$$Op_n[Op_1[t']( )](x) = Op_n[Op_1[t^0]( )](x) \neq t'(x).$$

Therefore $M_k \cap U_{v_k(t)} = \emptyset$ for all $t \in T$.

Q.E.D.

Conclusions

We investigated in this paper the size of sets of computable functions using the notion of meagre set. We mentioned in the introduction that besides the category theoretic approach a measure theoretic approach is conceivable. Again classic measure theory is not useful for our purpose; two reasons were mentioned at the beginning of section III. But constructive measure theory as it was developed by Erret Bishop ([2]) is a suitable tool. Using constructive measure theory we were able to prove most of the results given above. The proofs are usually more complicated and less intuitive. We want to mention a result which can be obtained using measure theory but we were not able to obtain using meagre sets.

The theorem deals with subrecursive degree structures. We state it for primitive recursive degrees. The primitive recursive functions form a minimum degree; call it σ. If a,b are primitive recursive degrees then $a \cup b$ is their join and $a \cap b$ is their meet. The meet of two degrees may not always exist but it is well-defined for almost all degrees a,b. $a \leq b$ means that a is primitive recursive in b. The quantifier $\overset{\infty}{\forall} a$ means for almost all a in the sense of measure theory; $\overset{\infty}{\forall} a \ \phi(a)$ if the set of degrees $\{b; \neg \phi(b)\}$ has measure 0.

<u>Theorem</u>: The almost all theory of primitive recursive degrees is decidable ([9]).

Stillwell ([8]) has proved this result for Turing degrees.

References

[1] Baker, T.: private communication.

[2] Bishop, E., <u>Foundations of Constructive Analysis</u>, McGraw Hill, 1967.

[3] Lynch, N., "Relativization of the Theory of Computational Complexity", MAC TR-99, Project MAC, 1972.

[4] Machtey, M., "The Honest Subrecursive Classes are a Lattice", CSD TR 82, Purdue University, 1973.

[5] Meyer, A. R. and Ritchie, D. M., "A Classification of the Recursive Functions". Zeitschrift fuer math. Logik und Grundlagen der Mathematik, Vol. 18, pgs 71-82 (1972).

[6] Oxtoby, J. C., <u>Measure and Category</u>, Springer Verlag, 1971.

[7] Rogers, H., Jr., <u>Theory of Recursive Functions and Effective Computability</u>, McGraw Hill, 1967.

[8] Stillwell, J., "Decidability of the 'Almost All' Theory of Degrees", The Journal of Symbolic Logic, Vol. 37, pgs 501-506, Sept. 1972.

[9] Mehlhorn, K., "The 'Almost All' Theory of Subrecursive Degrees is Decidable", CSD TR 73-170, Cornell University, 1973.

[1]: John T. Gill III (Dept. of Electrical Engineering, Stanford University) submitted closely related work to this conference. He proved the existence of recursive sets C_1, C_2, C_3 such that

(a) Dpol (C_1) = NDpol (C_1)

(b) Dpol $(C_2) \neq$ NDpol (C_2)

(c) NDpol (C_3) is not closed under complementation.