V-08

SOUTHWESTERN AT MEMPHIS

DEPARTMENT OF MATHEMATICS

THE PRIORITY ARGUMENT

By

Leigh Norman Brasington

May 15, 1971

Submitted in partial fulfillment of the requirements for the Bachelor of Arts degree with Honors in Mathematics

THE PRIORITY ARGUMENT

Chapter 1: Recursive Function Theory

Recursive function theory is the study of algorithms of natural numbers. The priority argument is a general method for proving theorems in this theory. It provides a means for assigning priorities to infinitely many sets used in the enumeration of certain types of sets.

N is the set $\{0,1,2,3,\ldots,n,n+1,\ldots\}$ of natural numbers. Unless otherwise stated, all numbers are natural numbers, all sets are subsets of N, and all function are from N^k into N. A partial function is a function from a subset of N^k into N.

The fundamental notion of recursive function theory is the notion of an algorithm for computing a partial function. There are several equivalent precise definitions of algorithms. Among these are the definition of Turing (see Davis [1958]), Kleene [1952] and Markov (see Mendelson [1964]). For our purposes an intuitive description is adequate. An algorithm is a finite, ordered set of instruction such that a computer, using an element of N^k as input, can follow these instructions without the use of any intelligence. The output of the algorithm (if there is any) is an element of N and will be arrived at in a finite amount of time. Notice we have said the computer needs no intelligence to follow the instruction. We mean by this that it only needs to understand each instruction and be able to carry it out. If the computer can't understand or carry out an instruction, then it does nothing and there is no output. Thus if our computer understands only English and comes across the instruction "aflp-# nop zed" or the instruction "Express the ratio of the circumference of a circle to its diameter as a rational number" it will do nothing and there will be no output. Examples of algorithms are 1) the Euclidian algorithm for finding the greatest common divisor of two numbers and 2) the method of Eratosthene's sieve for finding the n^{th} prime number. 3) "Take x as input, Add x to itself one time, Subtract 1 from this result, Give this number as output" is an algorithm for finding 2x-1. Also 4) "Do nothing" and 5) "rplfmhG,t" are algorithms, but they give no output for any number. Finally, 6) "If the input is 7, 15, 93, or 1079 give 8 as output" is an algorithm which gives output for only four different inputs.

Notice that the instructions (and hence the algorithms) are strings of symbols and that in order for the computer to understand the instruction there must be only a finite number of symbols choosen from a finite set, called the computer's vocabulary, to which the computer attaches meaning. Using this fact we can prove the following result due to Kleene and known as the Enumeration Theorem:

Theorem. There exists a function f that assigns a number to each algorithm. This function is 1-1 and onto. Both f and f^{-1} are effective (in that given an algorithm A, f(a) can be effectively computed and conversely given $n \in N$, A such that f(A)=n can be effectively found from n).

<u>Proof</u>: Assign each symbol in the computer's vocabulary (and the blank space) a unique number. The ordering of these numbers will be the ordering of the symbols. Let S_i be the i^{th} symbol, with n symbols in all. Then enumerate all the algorithms in the following order: $S_1, S_2, S_3, \ldots, S_n, S_1S_1, S_1S_2, \ldots, S_1S_n,$ $S_2S_1, S_2S_2, S_2S_3, \ldots, S_2S_n, \ldots S_nS_1, \ldots, S_nS_n, S_1S_1S_1,$ $S_1S_1S_2, \ldots, S_1S_1S_n, S_1S_2S_1, \ldots, S_1S_2S_n, \ldots, S_1S_nS_1, \ldots,$ $S_1S_nS_n, \ldots, S_nS_1S_1, \ldots, S_nS_nS_n, S_1S_1S_1S_1, \ldots,$ that is, all the one symbol algorithms, followed by all the two symbol algorithms, followed by the three, the four, etc. with each group of the same length ordered alphabetically. This listing is 1-1, onto N and is effective. The n^{th} algorithm in the above listing is denoted A_n and n is called the Gödel number of A_n.

To each algorithm A, there corresponds a function from a subset of N^k into N defined as follows: given x as input, $\varphi(x)$ is the output of the algorithm A with x as input. If there is no output for x, x is not in the domain of φ, hence we say $\varphi(x)$ is undefined. Thus the domain and range of φ may both be empty. The function φ corresponding to the algorithm with Godel number n is denoted φ_n. Notice that two functions φ_n and φ_m may be equal as functions but the algorithms A_n and A_m may be quite different, that is with $n \neq m$.

A function f is a <u>partial recursive function</u> iff $f = \varphi_n$ for some n. A <u>recursive function</u> is a partial recursive function whose domain is all of N^k. Now the set of partial recursive functions is countable by the Enumeration Theorem, but the set of all functions from subsets of N into N is not countable. Hence there exists a function which is not partial recursive.

A set A is <u>recursive enumerable</u> (is r.e.) iff A is the range of a partial recursive function. We denote the r.e. set which is the range of φ_n by E_n.

The characteristic function of a set A is the function X_A from N into $\{0,1\}$ defined by $X_A(x) =$ 1 if $X \in A$, 0 if $X \notin A$. The set A is <u>recursive</u> iff X_A is a recursive function.

Notice the following: A is recursive if and only if there is an effective procedure for deciding whether or not an arbitrary number is in A.

<u>Proof</u>: $\Longrightarrow$) Assume A is recursive. Then there is an algorithm for finding $X_A(x)$. If $X_A(x) = 1$, $x \in A$. If $X_A(x) = 0$, $x \notin A$.

$\Longleftarrow$) Assume there is an effective procedure for deciding whether or not $n \in A$ for any n. Then X_A is recursive. Then A is recursive.

We now prove the following lemma: the set of all ordered triples of numbers can be enumerated.

Proof: Let (a,b,c) be an ordered triple. The volume of the triple is a+b+c. The height of the triple is b+c. Then for each $n \in N$ the number of triples with volume n is finite and for any volume the number of triples with height less than or equal to that volume is finite. Thus we can effectively list the order triples by putting them in order by volume and within each volume by height. The ordering would go (0,0,0), (0,0,1), (0,1,0), (1,0,0), (0,0,2), (0,1,1), (1,1,0), (2,0,0), (0,0,3) etc. A corollary to this lemma is that the set of ordered pairs can be enumerated.

The r.e. sets are those whose elements can be effectively listed.

Proof: $\Rightarrow$) Let $A_{nj}(i)$ be the j^{th} act in computing $\varphi_n(i)$. Start computing $\varphi_n(i)$ for all $i \in N$ simultaneously. This means perform the acts in the order for pairs where (j,i) is the pair for $A_{nj}(i)$. Whenever $\varphi_n(i)$ is found, add this to the list of E_n.

$\Leftarrow$) Suppose x_0, $x_1, \ldots$ is an effective listing of a set. Define φ by $\varphi(i)= x_i$. Then φ is a partial recursive function since the listing is effective and it's range is $\{x_i : i \in N\}$. We will make use of the above remarks throughout that which follows.

The following are examples of recursive sets: 1) N 2) $\emptyset$ 3) the set of even numbers 4) the set of prime numbers.

The compliment of the set A(denoted A^c) is the set of all a in N such that a is not in A. Thus $X_{A^c}=1-X_A$, so if A is recursive, so is A^c.

If A is finite then A is recursive.

Proof: Let x be given. Look through the elements of A. If x appears $x \in A$ and $X_A(x)=1$. If x does not appear, $x \in A^c$ and $X_A(x)=0$. This is an effective procedure since A is finite.

Since the compliment of every recursive set is recursive, every cofinite set is recursive.

We now prove several theorems; first, if A is recursive, then A is r.e.

Proof: The following is an effective procedure for listing the elements of any recursive set: For each $i \in N$ find $X_A(i)$. If $X_A(i) = 1$, adjoin i to the list of A.

<u>Theorem</u>: A is recursive if and only if A and A^c are both r.e.

<u>Proof</u>: $\Rightarrow$) A recursive $\Rightarrow A^c$ recursive $\Rightarrow A^c$ is r.e. $\Big\}$ $\Rightarrow$ A and A^c are r.e

A recursive $\Rightarrow$ A is r.e.

$\Leftarrow$) Since A and A^c are r.e., each is the range of a partial recursive function. Let A = Range φ_n and A^c=Range of φ_m.

Then for a given x look at $\varphi_n(0)$, $\varphi_m(0)$, $\varphi_n(1)$, $\varphi_m(1)$, $\varphi_n(2)$, $\varphi_m(2)$, ... x must appear as one of these since $A \cup A^c = N$. If $x = \varphi_n(y)$ for some y, $x \in A$. If $x = \varphi_m(y)$ for some y, $x \in A^c$.

<u>Theorem</u>: If A and B are r.e. so are $A \cap B$ and $A \cup B$.

<u>Proof</u>: Let $a_1, a_2, a_3, \ldots$ and $b_1, b_2, b_3, \ldots$ be effective listings for A and B respectively Then $a_1, b_1, a_2, b_2, \ldots$ is an effective listing for $A \cup B$. Enumerate A and B. When x appears in both lists, adjoin x to the set $A \cap B$

A corollary to this theorem is: If A and B are recursive so are $A \cap B$ and $A \cup B$.

<u>Proof</u>: A and B recursive $\Rightarrow$ A and B are r.e. $\Rightarrow A \cup B$ is r.e.

A and B recursive $\Rightarrow A^c$ and B^c are recursive $\Rightarrow A^c$ and B^c are r.e.

$\Rightarrow A^c \cap B^c$ is r.e. $\Rightarrow (A \cup B)^c$ is r.e.

But $A \cup B$ and $(A \cup B)^c$ r.e. $\Rightarrow A \cup B$ is recursive.

Similarly for $A \cap B$.

Most of the early results in the theory of recursive functions relied on a diagonal argument. This type of argument is illustrated in the proof of the following theorem:

<u>Theorem</u>: The set of recursive functions is not r.e.

<u>Proof</u>: Assume it is r.e. Then we can write out the recursive functions in order $f_0, f_1, f_2, \ldots$. Define f such that $f(n) = f_n(n) + 1$.

f is clearly recursive. Then it must be in the list, that is, $f = f_n$ for some n. But $f_n(n) = f(n) = f_n(n) + 1$ which is a contradiction.

Corollary: The set of recursive functions is not recursive. This means there is no algorithm for telling whether or not a given algorithm defines a recursive function (rather than only a partial recursive function).

The Creative Set K.

We shall now define a set which is r.e. but not recursive. We let K be the set of all numbers n such that $n \in E_n$, i.e. $K = \{n : n \in E_n\}$. K is known as the creative set.

Theorem: K is r.e.

Proof: Let a_{ij} be the j^{th} act in the enumeration of E_i. Enumerate all the E_i's simultaneously. This means perform the acts in the order of the enumeration of ordered pairs with a_{ij} corresponding to (i,j). Whenever n appears in E_n place n in the list of elements of K. Thus K is effectively enumerated.

Theorem: K is not recursive.

Proof: Assume K^c is r.e. Then $K^c = E_n$ for some n. Thus $n \in K^c \Leftrightarrow n \in E_n \Leftrightarrow n \in K$. This is a contradiction. Hence K^c is not r.e. and thus K is not recursive.

Chapter 2: Turing Reducibility and Post's Problem

We wish now to discuss the concept of one set being recursive in (or reducible to) another. Intuitively, A is recursive in B means that there is an algorithm for computing X_A from X_B. Suppose we have a set B and an algorithm A_n with a finite number of instructions of the following type: "If $X_B(y)=0$, then do ..., if $X_B(y)=1$, then do ... ." Now if B is recursive our computer will have no trouble deciding whether $X_B(y)$ is 0 or 1. But if B is not recursive, then there may be no output. Now if our computer had some method of knowing $X_B(y)$ whether or not B was recursive, then there would be no problem. (Turing spoke of the computer having an oracle.) So we define A_n^B to be an algorithm with X_B attached. We then define φ_n^B to be the partial function defined by A_n^B.

We now make the definition that A is recursive in B, or A is Turing reducible to B (denoted $A \leq B$) if and only if there exists an n such that $\varphi_n^B = X_A$. We have the following propositions: 1) $A \leq A$ and 2) if $A \leq B$ and $B \leq C$, then $A \leq C$. Proofs: 1) Immediate since clearly X_A is recursive in A. 2) Our computer can evaluate $X_A(n)$ for any n by evaluating X_B for a finite number of inputs. It can evaluate X_B for each of these inputs by evaluating X_C for a finite number of of inputs. Thus we can find $X_A(n)$ by evaluating X_C for a finite of inputs. Thus $A \leq C$. Thus if we define $A \sim B$ iff $A \leq B$ and $B \leq A$, we have that $\sim$ is an equivalence relation.

We now define $\bar{A} = \{B : A \sim B\}$. $\bar{A}$ is called the Turing degree of A. The notion $\bar{A} \leq \bar{B}$ iff $A \leq B$ is well defined. Proof: Let $A \leq B$. Suppose $A' \in \bar{A}$ and $B' \in \bar{B}$. Then $A \sim A' \Rightarrow A' \leq A$ and $B \sim B' \Rightarrow B \leq B'$. Hence $A' \leq A \leq B \leq B'$. Hence $A' \leq B'$. Therefore we have a partial ordering of Turing degrees. The lowest Turing degree is the degree of a recursive set, for let A be recursive. Then A is recursive in any set B since we can find $X_A(n)$ without having to ask any questions about B. This lowest degree is denoted $\bar{\emptyset}$ since $\emptyset$ is recursive.

The degree of $\bar{A}$ is a recursive enumerable Turing degree (r.e. degree) iff there exist $B \in \bar{A}$ such that B is r.e. Notice that $B \in \bar{A}$ and $\bar{A}$ is a r.e. degree does <u>not</u> necessarily mean B is r.e. For consider $\bar{K}$. K is r.e.; thus $\bar{K}$ is r.e. However K^c is not r.e. but clearly $K \sim K^c$, and hence $K^c \in \bar{K}$, by $X_K = 1 - X_{K^c}$. The highest r.e. degree is $\bar{K}$. To prove this it suffices to show if A is r.e. then $A \leq K$. Let A be r.e. Fix m. Define $f^m(x)$ by $f^m(x) = x$ if $m \in A$ and $f^m(x)$ is undefined otherwise. f^m is partial recursive. Proof: Generate A and when m appears in A give output x. This is an algorithm; thus each $f^m = \varphi_i$ for some i. Now this i can be found recursively from m by the Enumeration Theorem. Hence there is a recursive function g such that $g(m) = i$ where $\varphi_i = f^m$. Thus $E_{g(m)} = N$ if $m \in A$; $\emptyset$ if $m \in A$. Hence $m \in A \Leftrightarrow E_{g(m)} = N \Leftrightarrow g(m) \in E_{g(m)} \Leftrightarrow g(m) \in K$. Thus $A \leq K$.

The question arises whether the only r.e. degrees are $\bar{\emptyset}$

and $\bar{K}$. This problem was first posed by Emil Post in 1944 and became known as Post's Problem. It was finally solved in 1956 when Friedburg and Muchnik independently and almost simultaneously found the solution. This solution is presented in the next chapter as the first example of the priority argument.

Chapter 3: The Solution to Post's Problem

Difficulties involved in finding a Solution

Post's Problem remained unsolved for 12 years. This was due in large measure to the fact that to show that there are more than two r.e. degrees requires a nonconstructive approach. We discuss what this means below.

We say that A is r.e. in B iff A is the range of φ_n^B for some n. Then we have the following theorem: $A \leq B \Leftrightarrow A$ and A^c are both r.e. in B.

Proof: $\Leftarrow$) Let x be given. Let A be r.e. in B by φ_n^B and A^c by φ_m^B. Look at $\varphi_n^B(0), \varphi_m^B(0), \varphi_n^B(1), \varphi_m^B(1), \ldots$. x must appear in this list in a finite number of steps since $A \cup A^c = N$. If $\varphi_n^B(i)=x$, then $X_A(x)=1$; if $\varphi_m^B(i)=x$, then $X_A(x)=0$. This is an algorithm in B and thus $A \leq B$.

$\Rightarrow$) Suppose $X_A = \varphi_n^B$. Then simultaneously begin computing $\varphi_n^B(x)$ for all x. Whenever $\varphi_n^B(x)=1$ put x in the list of A. This is an effective listing by an algorithm in B. Let the first element listed be x_0, the second x_1, etc. Then φ^B such that $\varphi^B(i)=x_i$ is a function recursive in B with range A. Similarly for A^c.

To show there are more than two r.e. degrees we can try to find a r.e. set C such that $C \nleq \emptyset$ and $K \nleq C$. Now by the above

$A \nleq B$ for A and B r.e. is equivalent to for all x, $A \neq E_x^B$ or $A^c \neq E_x^B$. But we know that there is an x such that $A = E_x^B$. Thus for all x, $A^c \neq E_x^B$. This is equivalent to for each x there is a y such that $y \notin A^c$ if and only if $y \in E_x^B$. Now suppose this y can be found from x by a recursive function f. Then we would have there is a recursive f such that for all x, $f(x) \notin A^c \Leftrightarrow f(x) \in E_x^B$. We then say A is <u>constructively nonrecursive</u> in B, i.e. $(\exists \text{ recursive } f) \forall x (f(x) \in A \Leftrightarrow f(x) \in E_x^B)$. We may then want to try to find a set C such that C is r.e. and C is constructively nonrecursive in K. But if A and B are r.e. and if A is constructively nonrecursive in B, then B is recursive.

Proof: Let z be given. Let $B = E_{m_0}$. Consider the algorithm "Take x as input. If $X_B(z) = 0$ give x as output." This is an algorithm in B and hence is A_n^B for some n. Then we have a partial function in B φ_n^B such that $\varphi_n^B(x) = x$ if $z \notin B$ and is not defined if $z \in B$. Now n depends only on z. Given z we can write the algorithm above and then code it by the method of Chapter 1 to find n. Then there is an algorithm for finding n from z. Therefore there is a recursive function g such that $g(z) = n$. Now notice $E_n^B = N$ if $z \notin B$ and $\emptyset$ if $z \in B$. Thus we have shown there is a recursive g such that $E_{g(z)}^B = N$ if $z \notin B$ and $\emptyset$ if $z \in B$. Now we have a recursive f such that $\forall x (f(x) \in A \Leftrightarrow f(x) \in E_x^B)$. Consider $fg(z)$ for any z. $fg(z) \in A \Leftrightarrow f(g(z)) \in E_{g(z)}^B$ by our choice of f. $f(g(z)) \in E_{g(z)}^B \Leftrightarrow E_{g(z)}^B \neq N$ by our choice of g. But $E_{g(z)}^B = N \Leftrightarrow z \in B$; hence we have $fg(z) \in A \Leftrightarrow z \notin B \Leftrightarrow z \in B^c$.

Then to list B^c we enumerate A and compute $fg(z)$ for $z=0,1,2,\ldots$ Then if $fg(z)$ appears in our listing of A we put z in our listing of B^c. Thus B^c is r.e. We know B is r.e. Thus B is recursive. Thus by the above it is impossible to find a C that is nonrecursive in $\emptyset$ and such that K is constructively nonrecursive in C because K constructively nonrecursive in $C \Rightarrow C \sim \emptyset$.

The Solution to Post's Problem

Basically our idea will be to find two r.e. sets, neither of which is recursive in the other. This will give us at least four r.e. degrees. (We will later show there are at least a countable number of r.e. degrees.) This proof is due to Sacks [1966].

If $L=\{h_0, h_1, \ldots, h_m\}$, then $j(L)= 2^{h_0}+\ldots+2^{h_m}$. j is 1-1, onto from the class of all finite sets onto N if $j(\emptyset)= 0$. Both j and j^{-1} are effective (i.e., recursive).

Let $p_0, p_1, p_2, \ldots$ be 2, 3, 5, ... (i.e. the primes in their natural order). Then $a\in N \Rightarrow a=p_0^{a_0}p_1^{a_1}\ldots p_n^{a_n}\ldots$ with all but a finite number of the $a_i = 0$. Let $(a)_i=a_i$. Every function $(a)_i$ is clearly recursive and we let $(0)_i=0$ for all i.

Definition 1: A requirement is a set (perhaps empty) of ordered pairs of disjoint finite sets of numbers. We write $R=\{(F_i, H_i):i\in I\}$ where I is some set for enumerating the pairs and where $F_i \cap H_i=\emptyset$. The set T meets R if there is $i\in I$ such that $F_i\subseteq T$ and $H_i \cap T=\emptyset$.

Let $t: N \to N$. Then t enumerates requirements if for each $s \in N$, $j^{-1}((t(s))_0)$ and $j^{-1}((t(s))_1)$ are disjoint finite sets. The requirements enumerated by t are R_0, R_1, ... where $R_k = \{(j^{-1}((t(s))_0), j^{-1}((t(s))_0)) \mid (t(s))_2 = k\}$. We write F^s for $j^{-1}((t(s))_0)$ and H^s for $j^{-1}((t(s))_1)$ and $g(s)$ for $(t(s))_2$. Thus $R_k = \{(F^s, H^s) \mid g(s)=k\}$.

A function f is recursive in a function g iff the range of f is recursive in the range of g. A set A is recursive enumerable in f if A is the range of a function recursive in f.

Definition 2: If t enumerates requirements, we define the <u>priority set</u> T of t to be as follows:

$T = \bigcup_{s \geq 0} T_s$

$s=0$: Then $T_0 = \emptyset$.

s 0: $T_s = T_{s-1}$ if (a), (b), or (c) holds. Otherwise $T_s = T_{s-1} \cup F^s$.

(a): There is $r<s$ such that	(b): There is $r<s$ such that	(c):
1) $r>0$	1) $r>0$	$H^s \cap T_{s-1} \neq \emptyset$
2) $g(r) < g(s)$	2) $g(r)=g(s)$	
3) $F^r \not\subseteq T_{r-1}$	3) $F^r \not\subseteq T_{r-1}$	
4) $F^r \not\subseteq T_r$	4) $F^r \not\subseteq T_r$	
5) $H^r \cap T_{s-1} = \emptyset$	5) $H^r \cap T_{s-1} = \emptyset$	
6) $H^r \cap F^s \neq \emptyset$		

We shall see that T is r.e. in t and meets every member of a certain subclass of the class of requirements enumerated by t.

Definition 3: For each k, R_k is <u>met</u> at stage s if

1) $s>0$
2) $k = g(s)$
3) $F^s \not\subseteq T_{s-1}$
4) $F^s \subseteq T_s$

If R_k is met at stage s then $H^s \cap T_s = \emptyset$.

Proof: Suppose R_k is met at stage s. Then $F^s \not\subseteq T_{s-1}$ and $F^s \subseteq T_s$. Hence $T_s \neq T_{s-1}$. Then $T_s = T_{s-1} \cup F^s$ and neither (a), (b), or (c) holds. Since (c) does not hold, $H^s \cap T_{s-1} = \emptyset$. $F^s \cap H^s = \emptyset$ for all s. Hence $H^s \cap T_s = \emptyset$.

Now at a later stage s than the stage r when R_k was met it may turn out that we will want to put something in T_s that will ruin our chance of having met R_k at stage r. Thus we say R_k is <u>injured</u> at stage s if there is an $r < s$ such that

1) R_k was met at stage r

2) $H^r \cap T_{s-1} = \emptyset$

3) $H^r \cap T_s \neq \emptyset$

Now if R_k is met at stage r and not injured at any later stage then T meets R_k. Proof: We have $F^r \subseteq T_r$ and $H^r \cap T_s = \emptyset$ for all $s \geq r$. Then $F^r \subseteq T$ since $T_r \subseteq T$. Further $H^r \cap T = \emptyset$ for if $n \in H^r \cap T$, then $n \in H^r$ and $n \in T$ which means $n \in H^r$ and $n \in T_s$ for some s, i.e. $n \in H^r \cap T_s$ for some s. But $H^r \cap T_s = \emptyset$ for $s \geq r$ and also for $s<r$ since $s < r$ implies $T_s \subseteq T_r$.

We can now see where the difficulty lies. We must try to make T meet infinitely many R_k's. To do this we must put some things in T and keep other things out. Thus we run the risk of each time we put something in T to make it meet R_k we put something in T that injures R_j. Therefore we have defined T such that we have set up priorities for the R_k's so that T will meet enough R_k's to get the job done.

In Definition 2, (b) holds if and only if $R_{g(s)}$ was met at stage r and was not injured at stage r+1,..., s-1. Then $T_s = T_{s-1}$ to avoid injuring $R_{g(s)}$. Also note nothing would be gained by trying to meet $R_{g(s)}$ at stage s for stage r has already taken care of fulfilling the condition such that T will meet $R_{g(s)} = R_{g(r)}$. If (c) holds, there is no hope of meeting $R_{g(s)}$ at stage s, so we add nothing to avoid injuring anything. Condition (a) holds if and only if $g(r) < g(s)$ and $R_{g(r)}$ was met at stage $r < s$ and was not injured at stages r+1,..., s-1. However if F^s is adjoined to T_{s-1}, then $R_{g(r)}$ is injured at step s. We do not adjoin F^s to T_{s-1} to meet $R_{g(s)}$ at stage s at the cost of injuring $R_{g(r)}$ at stage s. Thus we assign a higher priority to $R_{g(r)}$ than to $R_{g(s)}$ when $g(r) < g(s)$.

If R_k is injured at stage s, then there is $i < k$ such that R_i is met at stage s. Proof: Suppose R_k is injured at stage s. Then there is $r < s$ such that R_k was met at stage r, $H^r \cap T_{s-1} = \emptyset$ and $H^r \cap T_s \neq \emptyset$. Thus $T_s \neq T_{s-1}$. Hence $0 < r < s$, $F^r \not\subseteq T_{r-1}$, $F^r \subseteq T_r$, $H^r \cap T_{s-1} = \emptyset$ and $H^r \cap F^s \neq \emptyset$, the latter because $H^r \cap T_s \neq \emptyset$ and $H^r \cap T_{s-1} = \emptyset$. If $g(r) \leq g(s)$, then either (a) or (b) holds at stage s and $T_{s-1} = T_s$. But $T_{s-1} \neq T_s$, hence $g(s) < g(r)$. Now $g(r) = k$. Let $i = g(s)$. Hence $i < k$. By the above R_i is met at stage s.

Lemma 1: Suppose $r < s$ and R_k is met at stage r and at stage s. Then there is a u such that $r < u < s$ and R_k is injured at stage u.

Proof: We have $g(r)=g(s)=k$, $F^r \not\subseteq T_{r-1}$, $F^r \subseteq T_r$. Then $H^r \cap T_{s-1} \neq \emptyset$

for otherwise (b) holds and R_k would not be met at stage s. Since R_k is met at stage r, $H^r \cap T_r = \emptyset$. Hence there is a u such that $r < u < s$ and $H^r \cap T_{u-1} = \emptyset$ and $H^r \cap T_u \neq \emptyset$. But this means R_k is injured at stage u.

Lemma 2: For each k, the number of times R_k is injured is finite.

Proof: Induction on k. R_0 is never injured since there is no $i < 0$ and no R_i can be met when R_0 is injured.
Assume $R_0, R_1, R_2, \ldots R_{k-1}$ are injured only a finite number of times and consider R_k. Suppose for contradiction that R_k is injured infinitely often. Then each time R_k is injured some R_i such that $i < k$ is met. Since there are only finitely many $i < k$ one of these R_i's must be met infinitely often. Then by lemma 1, this R_i is injured infinitely often, a contradiction.

Corollary: For each k, the number of times R_k is met is finite.

Proof: Suppose for contradiction that R_k is met infinitely often. Then by lemma 1, R_k is injured infinitely often, contradicting lemma 2.

We now make the following definition: For each k, R_k is <u>t-dense</u> if for each finite set L, there is an $s > 0$ such that

1) $g(s)=k$
2) $F^s \not\subseteq T_{s-1}$
3) $H^s \cap T_{s-1} = \emptyset$
4) $L \cap F^s = \emptyset$

<u>Theorem 1</u>: If t enumerates requirements, then the priority set T of t is r.e. in t and meets every t-dense requirement.

Proof: Clearly T_s as a function of s is recursive in t. Hence t is r.e. in t. Fix k and suppose R_k is t-dense. If we can find an r such that R_k is met at stage r and is not injured at any stage afterwards, then T meets R_k. Let $i \leq k$. By lemma 2 and its corollary, there is a u_i such that R_i is neither injured or met at any stage $s \geq u_i$. Let $u = Max(u_0, u_1, \ldots, u_k)$. Then $i \leq k$ and $s \geq u$ means R_i is neither met nor injured at stage s. Let $L = \bigcup_{w \leq u}(H^w \cup F^w)$. Since R_k is t-dense, there is an $s > 0$ such that $g(s)=k$, $F^s \not\subseteq T_{s-1}$, $H^s \cap T_{s-1} = \emptyset$ and $L \cap F^s = \emptyset$. $s > u$ for if $s \leq u$ then $F^s \subseteq L$ which implies $L \cap F^s \neq \emptyset$ which contradicts R_k being t-dense. Hense R_k is not met at stage s. Since $k = g(s)$, $T_s = T_{s-1}$. Hence F^s was not adjoined at stage s, for otherwise $T_{s-1} \subset T_s$ since $F^s \not\subseteq T_{s-1.}$ hence either (a), (b), or (c) holds at stage s. (c) does not hold since $H^s \quad T_{s-1} = \emptyset$. Suppose (a) holds. Then there is an $r < s$ such that $g(r) < g(s) = k$, $R_{g(r)}$ is met at stage r and $H^r \cap F^s \neq \emptyset$. Since $g(r) < k$, $R_{g(r)}$ is met for the last time before stage u. Hence $r < u$. Then $H^r \subseteq L$. Since $L \cap F^s = \emptyset$, $H^r \cap F^s = \emptyset$. But in (a) $H^r \cap F^s \neq \emptyset$. Thus (a) does not hold, hence (b) holds at stage s. Therefore there is $r < s$ such that $g(r) = g(s) = k$, $r > 0$, R_k is met at stage r and is not injured after r and before s. Now $s > u$ by the definition of u. R_k is never injured after u. Hence T meets R_k since R_k is met at stage r and not injured at any later stage.

Finally before we give the solution to Post's Problem we have the following lemma: Let A_k be a subset of A such that for all n, $n \leq k$ and $n \in A$ implies $n \in A_k$. If $\varphi_n^{A_k}$ computes a value for x with computation by algorithm taking y steps with $y \leq n$, then φ_n^A computes the same value for x.

We make the following notational definition: $\exists x$ is read "there exists x" and $\exists x$([x satisfies some condition]) is "there exists x such that x satisfies the given condition"

<u>Theorem 2</u> (Friedberg, Muchnik): There exist two r.e. sets such that neither is recursive in the other (and hence their Turing degrees are incomparable).

Proof: We define T_s, F^s, H^s, $g(s)$ and t simultaneously by induction:

s=0: Then $T_0=F^0=H^0=\emptyset$ and $g(0)=0$.

s>0: Let $A_s^0=\{n:2n \in T_{s-1}\}$

and $A_s^1=\{n:2n+1 \in T_{s-1}\}$

Let $i = \begin{cases} 1 \text{ if } (s)_0 \neq 0 & \text{, i.e. if s is odd} \\ 0 \text{ otherwise} & \text{, i.e. if s is even} \end{cases}$

Let $e = (s)_1$ i.e. the power of 3 in the prime factorization of s.

Case 1: $\exists m \leq s$ and $\exists y \leq s$ such that

(I) $\varphi_e^{A_s^i}(p_e^m) = 0$ with computation by algorithm taking y steps

Let r be the greatest $m \leq s$ such that (I) holds.

Let $F^s=\{2p_e^r+1-i\}$

and $H^s=\{2n+i: n \leq s \text{ and } n \notin A_s^i\}$

and $g(s) = 2e+i+1$

Case 2: Otherwise let $F^s= H^s=\emptyset$ and $g(s) = 0$

Now let $t(s) = 2^{j(F^s)}3^{j(H^s)}5^{g(s)}$

Then t enumerates requirements. Proof: We must show $F^s \cap H^s=\emptyset$.

In Case 1, if i=0, F^s contains an odd number while H^s contains only even numbers,

if i=1, F^s contains an even number while H^s contains only odd numbers.

Thus in case 1, $F^s \cap H^s=\emptyset$. In Case 2 $H^s=\emptyset=F^s$, thus $F^s \cap H^s=\emptyset$.

Define T_s at each stage according to definition 2 so as to make $T = \bigcup_{s \geq o} T_s$ the priority set of t. Note that each step is effective: t is recursive, T is r.e. and condition (I) is effective because there are only finitely many $m \leq s$ and $y \leq s$. Note also that $H^s \cap T_{s-1} = \emptyset$ for all s. Proof: $H^s =$

$$\{2n+i: n \leq s \text{ and } n \notin A_s^i\} = \{2n+i: n\_s \text{ and } 2n+i \notin T_{s-1}\}.$$

This insures that condition (c) of definition 2 will not be met. Now let $A^o = \{n: 2n \in T\}$

and $A^1 = \{n: 2n+1 \in T\}$. Clearly A^o and A^1 are r.e. and also $A^i = \bigcup_{s \geq o} A_s^i$ for i=1,2. Hence for each n $\exists s(n \in A^i \Leftrightarrow n \in A_s^i)$.

Now we show A^1 is not recursive in A^o. Let $e \in N$. We show $X_{A'} \neq \varphi_e^{A^o}$. Assume $\varphi_e^{A^o}$ is a characteristic function else there is nothing to prove.

Case I: R_{2e+1} is t-dense. By theorem 1, T meets R_{2e+1}.

Hence $\exists s(g(s)=2e+1, F^s \subseteq T$, and $H^s \cap T = \emptyset$).

Case 1 holds at stage s for otherwise $g(s)=0 \neq 2e+1$ for all e.

Hence $g(s) = 2e+1 = 2e^* + i + 1$. Thus $i = 0$ and $e = e^* = (s)_1$.

Hence $\exists r$ and $\exists y \leq s$ such that $\varphi_e^{A_s^o}(p_e^r) = 0$ with the computation taking y steps, $F^s \subseteq T$, $2p_e^r + 1 \in T$ and hence $p_e^r \in A^1$.

Since $H^s \cap T = \emptyset$, $n \leq s$ and $n \notin A_s^o \Rightarrow 2n \notin T \Rightarrow n \notin A^o$ and hence $n \leq s$ and $n \in A^o \Rightarrow n \in A_s^o$.

Since $y \leq s$, $\varphi_e^{A^o}(p_e^r) = 0$ by the lemma just preceeding this theorem. So $\varphi_e^{A^o} \neq X_{A'}$ for $\varphi_e^{A^o}(p_e^r) = 0 \neq 1 = X_A(p_e^r)$.

Case II: R_{2e+1} is not t-dense. Then $\exists L$ a finite set such that for all $s > 0$ either $g(s) \neq 2e+1$ or $F^s \subseteq T_{s-1}$ or $L \cap F^s \neq \emptyset$.

(Recall that $H^s \cap T_{s-1} = \emptyset$ for all s.) We show that if m>0

$2p_e^m+1$ is greater than every $l \in L$, then

i) $p_e^m \in A^1$ and ii) $\varphi_e^{A^o}(p_e^m) \neq 0$. This shows $X_A \neq \varphi_e^{A^o}$.

Let $m > 0$ be such that $2p_e^m+1$ is greater than every $l \in L$.

Suppose for contradiction that $p_e^m \in A^1$. Then $2p_e^m+1 \in T$.

Let $s>0$ such that $2p_e^m \in T_s - T_{s-1}$. Then $F^s = \{2p_e^m+1\}$ and $T_s = T_{s-1} \cup F^s$. Hence $g(s) = 2e+1$, $F^s \not\subseteq T_{s-1}$, and $L \cap F^s = \emptyset$; but this contradicts the fact that R_{2e+1} is not t-dense.

Now for ii). Suppose for contradiction that $\varphi_e^{A^o}(p_e^m)=0$.

Let y be the number of steps in the computation. Now choose s large enough so that $m \leq s$ and $y \leq s$ and $e = (s)_1$ and $i = 0$. Then case 1 holds at stage s. Then $F^s = \{2p_e^r+1\}$ where $r \geq m$ and $g(s) = 2e+1$. Since $r \geq m$ we have $F^s \not\subseteq T_{s-1}$ by i). For if $F^s \subseteq T_{s-1}$, $2p_e^r+1 \in T_{s-1}$, $p_e^r \in A_s^1 \subseteq A^1$, but if $2p_e^m+1$ is greater than $l \in L$ for each l, and $r \geq m$, then $2p_e^r+1$ is greater than $l \in L$ for each l and hence by i) $p_e^r \notin A^1$, a contradiction. Thus $g(s) = 2e+1$, $F^s \not\subseteq T_{s-1}$ and $L \cap F^s = \emptyset$. But all this is impossible.

Thus $A^1 \not\leq A^o$. Similarly $A^o \not\leq A^1$. This concludes the proof.

<u>Corollary</u> (Solution to Post's Problem): There exist more than two r.e. degrees.

Proof: We now have $\bar{\emptyset}$, $\bar{A}^o$, $\bar{A}^1$, and $\bar{K}$, all distinct since $\bar{\emptyset}$ and $\bar{K}$ are upper and lower bounds.

Now let us take an intuitive look at what we have done. What we want to do is define two sets A^0 and A^1, neither of which is recursive in the other. To do this we proceed by stages. At stage s, if s is even, we work on A^1 to prevent it from being recursive in A^0. If s is odd, we work on A^0 to prevent it from being recursive in A^1. Let us assume s is even. Now we find the power of 3 in the prime factorization of s and call this number e. We now want to put something in A^1 or A^{1^c} that will prevent $\varphi_e^{A^0}$ from being the characteristic function of A^1. We use $\varphi_e^{A^0}$ since e will take on every value in N infinitely many times and hence $\varphi_e^{A^0}$ will be worked on infinitely often. But we don't know what A^0 is. However, we do have the lemma just preceding the theorem. So we have to construct A^0 when we are working on it, in accordance with the hypothesis of this lemma. Then if A_s^0 is what we have constructed of A^0 at stage s, $\varphi_e^{A^0}$ and $\varphi_e^{A_s^0}$ will compute the same value for a given x, provided the computation takes no more than s steps. So now we want to put something in A^1, call it x, that will prevent $\varphi_e^{A_s^0}$ from being X_{A^1}. We have to come up with an effective procedure for trying to find this x. We do this by looking at the e <u>th</u> prime, p_e, and seeing if there is an $m \leq s$ such that $\varphi_e^{A_s^0}(p_e^m) = 0$ in not more than s steps. Since there are only finitely many $m \leq s$, this procedure is effective. The reason for using this procedure will become clear later on. Now say we find an $m \leq s$ such that $\varphi_e^{A_s^0}(p_e^m) = 0$. Then if we put p_e^m in A^1, $X_{A^1}(p_e^m) = 1 \neq 0 = \varphi_e^{A_s^0}(p_e^m)$. But we can't just throw numbers into A^1 whenever we feel like it. We want A^1 to be constructed in accordance with the lemma, so that the odd stages will be effective as well. So we pick the largest m such that $\varphi_e^{A_s^0}(p_e^m) = 0$ in not more than s steps and call it r.

We put $2p_e^r+1$ in F^s and in H^s we put twice all the numbers less than or equal to s that are not already in A_s^0. We then make (F^s, H^s) an element of the requirement R_{2e+1}. Now if there is no m such that $\varphi_e^{A_s^0}(p_e^m)= 0$ in not more than s steps, we make $F^s= H^s= \emptyset$ and put (F^s, H^s) in the requirement R_0. Then we define a function t such that t enumerates these requirements. Then by using the priority set T of t, we can effectively decide what goes into A^1 and A^0 so that our t-dense requirements are met. Now notice if a requirement R_{2e+1} is met, there is a p_e^r in A^0 such that $\varphi_e^{A_s^0}(p_e^r)= 0$ for some s (by means of $F^s \subseteq T$) <u>and</u> A^0 is constructed such that $\varphi_e^{A_s^0}(p_e^r)=\varphi_e^{A^0}(p_e^r)$ (by $H^s \cap T= \emptyset$). (This is what Case I: R_{2e+1} is t-dense, proves). Hence $\varphi_e^{A^0} \neq X_{A^1}$. If a requirement R_{2e+1} is not met, then we want to have put something in A^{1^c} so that $\varphi_e^{A^0}$ is not X_{A^1}. Now R_{2e+1} is not t-dense if it is not met. So we have a finite set L such that at each stage s either we are not working with $\varphi_e^{A_s^0}$ (i.e., $g(s)\neq 2e+1$) or $F^s \subseteq T_{s-1}$ or $L \cap F^s \neq \emptyset$. We can find m such that $2p_e^m+ 1$ is greater than any $l \in L$, and hence if we can show $F^s= \{2p_e^m+ 1\}$ for some s, $L \cap F^s= \emptyset$. This is the reason for using numbers of the form $2p_e^m+ 1$ in connection with $\varphi_e^{A_s^0}$. The exponent m enables us to find a large enough number to show that if R_{2e+1} is not t-dense then $g(s)= 2e+ 1$ implies $F^s \subseteq T_{s-1}$ since $L \cap F^s= \emptyset$. By assuming $p_e^m \in A^1$ we show that $F^s \not\subseteq T_{s-1}$ and $g(s)= 2e+ 1$ for some s. Similarly we show that $\varphi_e^{A^0}(p_e^m)= 0$ implies $g(s)= 2e+ 1$ and $F^s \not\subseteq T_{s-1}$. These two contradictions give us $p_e^m \notin A^1$ and $\varphi_e^{A^0}(p_e^m)\neq 0$. Then we are done.

Chapter 4: Further examples of the Priority Argument

First we make the following definitions: Q_n is the set $\{0,1,\ldots,n-1\}$. r_n is the function from N onto Q_n such that $r_n(s)$ is the remainder when s is divided by n.

Corollary to the Friedberg-Muchnik Theorem: There exists a countably infinite family of r.e. degrees such that any two members of this family are incomparable with respect to $\leq$.

Proof: Assume no such family has more than h-1 elements. To show there is such a family with h elements.

We define T_s, F^s, H^s, g(s) and t simultaneously by induction:

s = 0: Then $T_o = F^o = H^o = \emptyset$ and $g(0) = 0$.

s > 0: Let $A_s^i = \{n : hn+i \in T_{s-1}\}$ for each $i \in Q_h$.

Let $e = (s)_1$

Case 1: $\exists m \leq s$ and $\exists y \leq s$ such that

(I) $\varphi_e^{A_s^{r(s)}}(p_e^m) = 0$ with the computation taking y steps.

Let r be the greatest $m \leq s$ such that (I) holds.

Let $F^s = \{hp_e^r + i : i \in Q_h \text{ and } i \neq r_h(s)\}$

and $H^s = \{hn + r_h(s) : n \leq s \text{ and } n \notin A_s^{r_h(s)}\}$

and $g(s) = he + r_h(s) + 1$.

Case 2: Otherwise let $F^s = H^s = \emptyset$ and $g(s) = 0$.

Now let $t(s) = 2^{j(F^s)} 3^{j(H^s)} 5^{g(s)}$. Then as before t enumerates requirements. Again define T_s at each stage so as to make $T = \cup_{s \geq o} T_s$ the priority set of t. Each step is effective and also $H^s \cap T_{s-1} = \emptyset$ for all s. Let $A^i = \{n : hn+i \in T\} = \cup_{s>o} A_s^i$, for all $i \in Q_h$. A^i is r.e. as before. To show A^i is not recursive in A^j we need only notational changes from theorem 2.

The following theorem is a theorem of Sacks [1966]:

Theorem 3: Let D be r.e. but not recursive. Then there exist two sets D_0 and D_1 such that $D_0 \cup D_1 = D$, $D_0 \cap D_1 = \emptyset$, $D \not\leq D_i$ for i=1,2.

Proof: Since D is r.e., $D = E_n$ for some n. Define f to be such that $f(x) = \begin{cases} \varphi_n(x) & \text{if there is no } y < x \text{ such that } f(y) = \varphi_n(x) \\ \text{undefined otherwise.} \end{cases}$

Clearly f is recursive and enumerates D without repitions.

Let $d(s,n) = \begin{cases} 1 & \text{if there is } k \leq s \text{ such that } f(k) = n \\ 0 & \text{otherwise} \end{cases}$

Then $\lim_{s \to \infty} d(s,n) = \chi_D(n)$.

Before we proceed further we make the following definition:

$\varphi_e^A(x) = z$ has computation number y if y = Max(the number of steps in the computation, the largest number that appears in doing the computation+1). This notion is well-defined since we arrive at the output z in a finite number of steps and each step has only a finite number of symbols (and hence numbers) in it.

We will now define six (6) recursive functions simultaneously by induction. At each stage s of the induction we will define $d(i,s,n)$, $t(i,s)$, $y(i,s,n,e)$, $P(i,s,n,e)$, $m(i,s,e)$ and $K(i,s,e)$ for all $i<2$ and all n and all e. The purpose of stage s is to put $f(s)$ in either D_0 or D_1, but not in both. Thus at stage s we will set either $d(0,s,f(s))$ or $d(1,s,f(s))$ equal to 1 and the other to 0.

Stage s=0: We set $d(1,0,n) = 0$ for all $n \neq f(s)$ and $d(1,0,f(0)) = 1$. We set $P(i,0,n,e) = 2$, $t(i,0) = y(i,0,n,e) = 1$ and $d(0,0,n) = m(i,0,e) = K(i,0,e) = 0$ for all $i<2$, all e and n.

Stage $s > 0$: For each i 2, let

$$t(i,s) = \begin{cases} \text{the least } e \leq s \text{ such that } f(s) < K(i,s-1,e) & \text{if such an } e \text{ exists} \\ s+1 & \text{otherwise} \end{cases}$$

Let $z(s) = 1$ if $t(1,s) \geq t(0,s)$ and let $z(s) = 0$ otherwise.

Thus $t(z(s),s) \geq t(1-z(s),s)$. We set

$$d(i,s,n) = \begin{cases} 1 & \text{if } i = z(s) \text{ and } n = f(s) \\ d(i,s-1,n) & \text{otherwise} \end{cases}$$

for all $i<2$ and all n.

We make D_i^s the set such that its characteristic function is $d(i,s,n)$.

We now define $y(i,s,n,e)$ and $P(i,s,n,e)$ for $i<2$ and for all n and e:

If $\exists y \leq s$ such that

(I) $f(y) = r$ and $\varphi_e^{D_i^r}(n)$ is defined and has computation number y,

we let $y(i,s,n,e)$ be the least y such that (I) holds and let $P(i,s,n,e) = \varphi_e^{D_i^r}(n)$ where $r=f(y(i,s,n,e))$.

Otherwise let $y(i,s,n,e) = s+1$ and $P(i,s,n,e) = s+2$.

Before we define $m(i,s,e)$ we note that for all $i<2$ and for all e

$$\exists t \leq s (d(s,t) \neq P(i,s,t,e)).$$

This is clear since $d(s,t)$ is 0 or 1 for all t and $P(0,s,s,e) = P(1,s,s,e) = s+2$ for all e because s appears in the computation of $\varphi_e^{D_i^r}(s)$ and $s \geq y$. Therefore we define $m(i,s,e)$ to be the least t such that $d(s,t) \neq P(i,s,t,e)$ for $i<2$ and for all e.

We define $K(i,s,e)$ for $i<2$ and all e such that

$$K(i,s,e) = \begin{cases} K(i,s-1,e) & \text{if for all } n,\ n < m(i,s,e) \text{ implies } y(i,s,n,e) \leq K(i,s,e) \\ \text{otherwise the least } t & \text{such that for all } n,\ n<m(i,s,e) \text{ implies } y(i,s,n,e) < t \end{cases}$$

We now list two remarks which will be needed in lemmas 2 and 4:

R1: For all $i<2$ for all s and e, $K(i,s+1,e) \geq K(i,s,e)$

R2: For all $i<2$ and for all s and e
$n < m(i,s,e)$ implies $y(i,s,n,e) \leq K(i,s,e)$

Both of these remarks are clearly true from the definition of K.

The six functions we have defined above are recursive.

Before we proceed with the rigorous, mathematical argument, let us consider the intutuve content of the above construction. Since D is r.e. our method is to enumerate D without repition by means of a recursive function f. At stage s we enumerate the s^{th} member of D and place it in just one of the sets D_o or D_1; we choose between D_o and D_1 according to a criterion based solely on our desire that D not be recursive in D_i for $i<2$. The functions t, y, P, m, and K serve to establish this criterion. The value of $z(s)$ at $s>0$ is 0 or 1 according to whether we put the s^{th} member of D in D_o or D_1.

In order that D_i be recursive enumerable the above functions must be recursive. Two obstacles separate us from this objective. First D is not recursive so when we try to make some estimate at stage s of our progress towards our goal of defining D_i such that $D \not\leq D_i$, we must make this estimate without any perfect knowledge of the membership of D. But since D is r.e., we can define a function $d(s,n)$ which is recursive and approaches X_D as s increases without bound.

The second obstacle consists of our inability to know at stage s if we have "met" some "requirements" once and for all. (The words in quotation marks are used in their intuitive sense rather than the strict mathematical sense of Chapter 3.) Our "requirements" are: $X_D \neq \varphi_e^{D_i}$ for $i<2$ and for all $e \geq 0$. It may appear that a "requirement" is "met" at stage s, but at a later stage, a change in the membership of D_i or in our approximation of the membership of D may alter things completely. Thus there are conflicting "requirements" that must be "met" and we are forced to make repeated attempts to "meet" them with the hope that eventually each of tham will be "met."

For each $i<2$ and $e \geq 0$, let R_{2e+i} denote the "requirement" that $X_D \neq \varphi_e^{D_i}$. Our system of priorities is the same as that of Chapter 3, i.e. R_n has higher priority than R_m if $n<m$. At stage s we examine all deductions whose computation numbers are not greater than s in order to determine as far as possible the result of applying the e^{th} partial function recursive in D_i. The function P (as a function of n) is merely an approximation of $\varphi_e^{D_i}$ at stage s. We compare this approximation with d(s,n) in order to define an initial segment of natural numbers on which the functions $\varphi_e^{D_i}$ and X_D appear to agree at stage s; the length of this segment is m(i,s,e). The value of K(i,s,e) is greater than or equal to the computation number of the deduction needed to establish the apparent equality of $\varphi_e^{D_i}$ and X_D for all $n < m(i,s,e)$.

When we choose between D_0 and D_1 at stage s, our choice is motivated by a desire to preserve as far as possible the apparent equality between $\varphi_e^{D_i}$ and X_D noted at stage s-1. If

$f(s) \geq K(i, s-1, e)$, then the apparent equality between $\varphi_e^{D_i}(n)$ and $X_D(n)$ for all $n < m(i, s-1, e)$ will not be disturbed if $f(s)$ is added to D_i. This is so because $f(s)$ is greater than the computation number of any deduction relevant to this apparent equality, and hence by the definition of the computation number, the addition of $f(s)$ to D_i will not affect any such deduction. If $f(s)$ is smaller than $K(i, s-1, e)$, then we arbitrarily regard the addition of $f(s)$ to D_i at stage s as an "injury" to "requirement" R_{2e+1}. The value of $t(i,s)$ is the least e less than or equal to s such that R_{2e+1} will be "injured" if $f(s)$ is put in D_i; if no such e exists, then $t(i, s)=s+1$. Thus, $R_{2t(i, s)+1}$ is the highest priority "requirement" that will be "injured" (if any) when $f(s)$ is added to D_i. The function z is defined in such a manner that when we are faced with the alternatives of "injuring" $R_{2t(0, s)}$ or $R_{2t(1, s)+1}$, we choose to "injure" the "requirement" of lower priority.

Thus it follows that we will never "injure" R_0, that we will "injure" R_1 only to avoid "injuring" R_0, and in general we will "injure" R_m at stage s only to avoid "injuring" R_n for some $n < m$. R_{2e+i} will be "met" if the set $\{K(i, s, e) : s \geq 0\}$ is finite, because if the latter set is finite, then $X_0(n)$ and $\varphi_e^{D_i}(n)$ will appear to be equal for only finitely many n during the course of the construction, and hence at the end of the construction there will be an n such that either $\varphi_e^{D_i}(n)$ is undefined or is not equal to $X_D(n)$.

Thus, what we have left to show is that for each $e \geq 0$

(a) R_{2e+i} is "injured" only infinitely often, and (b) the set $\{K(i, s, e)\ \ s \geq 0\}$ is finite (i.e. R_{2e+i} is met). We will do this by means of induction (in the form of an infinite descent).

Then we let $X_{D_i}(n) = \lim_{s \to \infty} d(i, s, n)$ and we are done, since at each stage s, either $d(0,s,f(n))$ or $d(1,s,f(s))$ was set equal to 1, but not both; hence, $D_0 \cup D_1 = D$ and $D_0 \cap D_1 = \emptyset$.

For the sake of contradiction we assume there is an $i<2$ such that (b) above is false. Let e^* be the least e such that the set $\{K(i,s,e) : i<2 \text{ and } s \geq 0\}$ is infinite. Let i^* be the least i such that $\{K(i,s,e^*) : s \geq 0\}$ is infinite. Our objective is to show this assumption leads to D recursive. First we show $R_{2e^*+i^*}$ is injured only finitely often.

Lemma 1: There is an s' such that for all $s \geq s'$ either $z(s) = 1-i^*$ or $e^* < t(i^*,s)$.

Proof: Suppose there are infitely many s such that $z(s) = i^*$ and $e^* \geq t(i^*,s)$. Let S be an infinite set such that for all $s \in S$, $s \geq e^*$, $z(s) = i^*$ and $e^* \geq t(i^*,s)$. Recall that $t(z(s),s) \geq t(1-z(s),s)$ for all $s>0$. Then for each $s \in S$, $e^* \geq t(i^*,s) \geq t(1-i^*,s) \geq 0$. Since S is infinite, there must be an infinite subset R of S and an e^{**} such that $t(i^*,s) = e^{**} \geq 0$ for all $s \in R$. Thus we have $s \geq e^* \geq t(i^*,s) \geq t(1-i^*,s) = e^{**}$ for all $s \in R$. Thus $f(s) < K(1-i^*,s-1,e^{**})$. But then $\{K(1-i^*,s-1,e^{**}): s \in R\}$ is infinite since the set $\{f(s): s \in R\}$ is infinite (recall f is 1-1). It follows from the definition of e^* and the fact that $e^{**} \leq e^*$, that $e^{**} = e^*$. Thus the set $\{K(1-i^*,s,e^*): s \geq 0\}$ is infinite since $s \in R$ implies $s \geq 0$. This means $1-i^* > i^*$ and thus $i^* = 0$.

But then for any $s \in R \subseteq S$ we have $e^* = t(0,s) = t(1,s) = e^{**}$ and $z(s) = i^* = 0$. This last is impossible since $z(s) = 1$ if $t(1,s) = t(0,s)$ and $s>0$.

Now let s^* be the least s' satisfying lemma 1. Then we are ready to prove the following lemma:

Lemma 2: If $s \geq s^*$ and $n < m(i^*,s,e^*)$, then $d(i^*,s,j) = d(i^*,s',j)$ for all j and s' such that $j < y(i^*,s,n,e^*)$ and $s' \geq s$.

Proof: Let s and n be such that $s \geq s^*$ and $n < m(i^*,s,e^*)$. Our proof proceeds by induction on $s' \geq s$. The induction hypothesis is $s' \geq s$ and $d(i^*,s,j) = d(i^*,s',j)$ for all $j < y(i^*,s,n,e^*)$. Since $s' \geq s^*$ we have either $z(s'+1) = 1-i^*$ or $e^* < t(i^*,s'+1)$ by lemma 1. If $z(s'+1) = 1-i^*$, then $d(i^*,s'+1,j) = d(i^*,s',j)$ for all j. Suppose then that $e^* < t(i^*,s'+1)$. Then $e^* < t(i^*,s'+1) =$ [the least $e \leq s'+1$ such that $f(s'+1) < K(i^*,s',e)$]. Thus $f(s'+1) \geq K(i^*,s',e^*)$ since $e^* < s^* < s'+1$. By remark R1, $K(i^*,s',e^*) \geq K(i,s,e^*)$ since $s' \geq s$, and by remark R2, $K(i^*,s,e^*) \geq y(i^*,s,n,e^*)$ since $n < m(i^*,s,n,e^*)$. But then $f(s'+1) \geq y(i^*,s,n,e^*)$. Then by the definition of d, $d(i^*,s'+1,j) = d(i^*,s',j)$ for all j such that $j < y(i^*,s,n,e^*)$

Lemma 3: If $s \geq s^*$ and $n < m(i,s,e^*)$, then $y(i^*,s,n,e^*) = y(i^*,s',n,e^*) \leq s$ for all $s' \geq s$.

Proof: Let $s,n,$ and s' be such that $s' \geq s \geq s^*$ and $n < m(i^*,s,e^*)$. Then $d(s,n) = P(i^*,s,n,e^*)$ by definition of P. Since $d(s,n) < s+2$, we have $P(i^*,s,n,e^*) = \varphi_{e^*}^{D_{i^*}^{f(y(i^*,s,n,e^*))}}(n)$ and $y(i^*,s,n,e^*) \leq s$. But then $y(i^*,s,n,e) =$ the least $y \leq s$ such that

(II) $\varphi_{e^*}^{D_{i^*}^{f(y)}}(n)$ is defined and has computation number y.

By lemma 2 $d(i^*,s,j) = d(i^*,s',j)$ for all $j < y(i^*,s,n,e^*)$; hence, $D_{i^*}^{f(y)}$ is the same for all y such that $y \leq s'$, hence the least $y \leq s'$ such that (II) is the same as the least $y \leq s$ such that (II), thus $y(i^*,s,n,e^*) = y(i^*,s',n,e^*)$.

Lemma 4: $\{m(i^*,s,e^*) : s \geq 0\}$ is infinite.

Proof: Suppose this set is finite. Let m' be the largest member of it. Let s' be such that $s' \geq s^*$ and $d(s,n) = X_D(n)$ for all s and n such that $s \geq s'$ and $n \leq m'$. Let m'' be the greatest member of $\{m(i^*,s,e^*) : s \geq s^*\}$. Let s'' be such that $s'' \geq s'$ and $m(i^*,s'',e^*) = m'' \leq m'$. We now show by induction on $s \geq s''$ that $m(i^*,s,e^*) = m''$ and $K(i^*,s,e^*) = K(i^*,s'',e^*)$ for all $s \geq s''$. Suppose then that $s \geq s''$, $m(i^*,s,e^*) = m''$ and $K(i^*,s,e^*) = K(i^*,s'',e^*)$. By lemma 3, $y(i^*,s,n,e^*) = y(i^*,s,n,e^*)$, since $s \geq s'' \geq s' \geq s^*$. This means that $P(i^*,s,n,e^*) = P(i^*,s+1,n,e^*) = d(s,n)$ for all $n < m(i^*,s,e^*)$. But $d(s+1,n) = d(s,n) = X_D(n)$ for all $n < m'' \leq m'$, since $s \geq s'$. Hence $P(i^*,s+1,n,e^*) = d(s+1,n)$ for all $n < m'' = m(i^*,s,e^*)$, and thus $m'' \leq m(i^*,s+1,e^*)$. It follows from the definition of m'' that $m(i^*,s+1,e^*) = m''$. But then by remark R2, $y(i^*,s,n,e^*) = y(i^*,s+1,n,e^*) \leq K(i^*,s,e^*)$ for all $n < m(i^*,s+1,e^*) = m'' = m(i^*,s,e^*)$. From this last and the definition of K, it is clear that $K(i^*,s,e^*) = K(i^*,s'',e^*)$ for all $s \geq s''$; but this is a contradiction since by definition of e^* and i^* $\{K(i^*,s,e^*): s \geq 0\}$ is infinite.

Lemma 5: If $s \geq s^*$ and $n < m(i^*,s,e^*)$, then $X_D(n) = P(i^*,s,n,e^*)$.

Proof: Let $s \geq s^*$ and $n < m(i^*,s,e^*)$. By lemma 3, $y(i^*,s,n,e^*) = y(i^*,s',n,e^*) \leq s$ for all $s' \geq s$. By lemma 4, there is $s'' \geq s$ such that $m(i*,s'',e*) \geq m(i*,s,e*)$ and $d(s'',n) = \lim_{s\to\infty} d(s'',n) = X_D(n)$. But then $d(s'',n) = P(i*,s'',n,e*) = \varphi_{e*}^{D_{i*}^{f(y(i*,s'',n,e*))}}(n)$, since since $n < m(i*,s'',e*)$ and $y(i*,s'',n,e*) \leq s \leq s''$. Thus $X_D(n) = \varphi_{e*}^{D_{i*}^{f(y(i*,s'',n,e*))}}(n) = \varphi_{e*}^{D_{i*}^{f(y(i*,s,n,e*))}}(n) =$ $P(i*,s,n,e*)$ since $y(i*,s'',n,e*) = y(i*,s,n,e*) \leq s$.

Let $t(n)$ be the least s such that $s \geq s*$ and $n < m(i*,s,e*)$; by lemma 4, $t(n)$ is defined for all n. By lemma 5, $X_D(n) =$ $P(i*,t(n),n,e*)$ for all n. Since $m(i*,s,e)$ and $P(i*,s,n,e*)$ are both recursive, t is recursive; hence, X_D is recursive which is a contradiction. Thus $\{K(i,s,e) : s\geq 0\}$ is finite for $i<2$ and all e.

Now fix i and e and let m be the largest member of $\{K(i,s,e) : s\geq 0\}$. We wish to show there is an $n\leq m$ such that $\varphi_e^{D_i}(n) \neq X_D(n)$. Suppose the contrary. Then let $y(n)$ be the computation number of $\varphi_e^{D_i}(n)$. y is recursive and $\varphi_e^{D_i}(n)=X_D(n)$ for all $n\leq m$. Let y be the largest member of $\{y(n): n\leq m\}$. Let s be such that $y\leq s$, $d(s,n) = X_D(n)$ and $d(i,s,j) = X_{D_i}(j)$ for all $n\leq m$ and $j\leq y$. Then $y(i,s,n,e) = y(n) \leq s$ for all $n\leq m$. Thus $P(i,s,n,e) = \varphi_e^{D_i^{f(y(i,s,n,e))}}(n) = \varphi_e^{D_i^{f(y(n))}}(n) = \varphi_e^{D_i}(n) = X_D(n) =$ $d(s,n)$ for all $n\leq m$. This means $m < m(i,s,e)$ by the definition of the function m. By remark R2, $y(m) = y(i,s,m,e) \leq K(i,s,e)$. But $K(i,s,e) \leq m$ by the definition of m. Hence, $y(m) \leq m$. But this is absurd since $y(m) =$ the computation number of $\varphi_e^{D_i}(m)$, and it must be greater than m.

This concludes the proof of theorem 3.

Corollary 1: Let D be r.e. but not recursive. Then there are two sets D_0 and D_1 such that $D_0 \cup D_1 = D$, $D_0 \cap D_1 = \emptyset$ and $D_i \not\leq D_{1-i}$ for $i<2$.

Proof: Suppose the contrary. By using Theorem 3, we get $D_i \leq D_{1-i}$ for some $i=0,1$. We may assume $D_0 \leq D_1$. Since $D=D_0 \cup D_1$, $D \leq D_0 \cup D_1$. Now $D_0 \cup D_1 \leq D_1$. Proof: We want to know if $n \in D_0 \cup D_1$. We ask if $n \in D_0$. We can find this out from D_1 since $D_0 \leq D_1$. We also ask if $n \in D_1$. We can find this out from D_1. If $n \in D_0$ or $n \in D_1$, then $n \in D_0 \cup D_1$. Hence $D_0 \cup D_1 \leq D_1$. Therefore $D \leq D_0 \cup D_1 \leq D_1$ which contradicts theorem 3. Hence $D_i \nleq D_{1-i}$ for $i < 2$.

Theorem 3 and corollary 1 may be stated together as follows: Each non-recursive, recursively enumerable set D is the union of two disjoint, recursively enumerable sets whose degrees are less than the degree of D and whose degrees are incomparable.

Corollary 2: There is no least non-recursive r.e. degree.

Proof: Assume that there is one and that it is $\bar{A}$. But A is the union of two sets whose degrees are less than that of A.

We conclude with another theorem of Sacks [1964]. The proof of this theorem has the interesting feature that each "requirement" may be "injured" infinitely often. Nonetheless we still manage to "meet" each requirement. The difference is the consequence of the fact that before, we only dealt with finite, initial segments of functions, while here we are forced to deal directly with entire functions.

We are given a non-recursive r.e. set B, and we wish to find an r.e. set D such that $D \nleq B$ but $B \leq D$. Since B is to be recursive in D, we are forced to add many members of B to D in complete disregard of all priorities. Thus, even the highest priority "requirement" may be "injured" infinitely often.

<u>Theorem</u> 4: If $\underline{b}$ and $\underline{c}$ are r.e. degrees such that $\underline{b} < \underline{c}$, then there exists a recursively enumerable degree $\underline{d}$ such that $\underline{b} < \underline{d} < \underline{c}$.

Proof: Let f and g be 1-1 recursive functions which enumerate the sets B and C without repitions such that $\overline{B}=\underline{b}$ and $\overline{C}=\underline{c}$. We define

$$b(s,n) = \begin{cases} 0 & \text{if } \exists t<s(f(t) = n) \\ 1 & \text{otherwise} \end{cases}$$

$$c(s,n) = \begin{cases} 2 & \text{if } \exists t<s(g(t) = n) \\ 1 & \text{otherwise} \end{cases}$$

Then $\lim_{s\to\infty} b(s,n)$ exists for all n and is $1-X_B(n)$. Similarly $\lim_{s\to\infty} c(s,n) = 1+X_C(n)$. Let k be a recursive function such that, for each s, $s< k(s)< k(s+1)$. We define three recursive functions:

1) $$y_b(s,n,e) = \begin{cases} \text{the least } y<s \text{ such that } \varphi_e^{B[s,y]}(n) \text{ has computation number } y, \text{ if one exists} \\ k(s) \quad \text{otherwise,} \end{cases}$$

where $B[s,y] = \{j<y : b(s,j) = 0\}$

2) $h(0,n,e) = 0$ and $h(s+1,n,e) = h(s,n,e)+sg(|y_b(s+1,n,e)-y_b(s,n,e)|)$

3) $t(s,i,e) = \sum_{j\leq i} h(s,j,e)$.

We define four recursive functions $y(s,n,e)$, $m(s,e)$, $r(s,n,e)$ and $d(s,n)$ simultaneously by induction on s. The desired degree $\underline{d}$ will be the degree of the set $\{n :$ there is an s such that $d(s,n)=0\}$.

$s = 0$: We set $y(0,n,e) = m(0,e) = 0$ and $r(0,n,e) = d(0,n) = 1$ for all n and e, except that $d(0,p(f(0),1)) = 0$ where $p(i,m) = p_i^{2+m}$. (Recall p_i is the i^{th} prime.)

$s > 0$: We set

$$y(s,n,e) = \begin{cases} \text{the least } y<s \text{ such that } \varphi_e^{D[s-1,y]}(n) \text{ has} \\ \qquad \text{computation number } y, \text{ if one exists} \\ 0 \quad \text{otherwise.} \end{cases}$$

We define $U(y(s,n,e)) = \begin{cases} \varphi_e^{D[s-1,y]}(n) & \text{if } y(s,n,e) \neq 0 \\ \text{undefined otherwise.} \end{cases}$

The definition of $m(s,e)$ has two cases:

Case m1: $\exists n < m(s-1,e)$ such that

$c(s,n) \neq U(y(s,n,e))$ and $y(s,n,e) \neq y(s-1,n,e)$.

Then we set $m(s,e)$ equal to the least such n.

Case m2: Otherwise. Then we set $m(s,e)$ equal to the

least n such that $m(s-1,e) \leq n < 2m(s-1,e)+s$

and $\exists t \leq n(c(s,t) \neq U(y(s,t,e))$, if such an n

exists, otherwise $m(s,e) = m(s-1,e)+s$.

Let $p(e,i,m) = p(e,p(i,m))$. We define $r(s,n,e)$ and $d(s,p(e,i,m))$ for all e,n,i,and m by a simultaneous induction on e. First we set $d(s,p(f(s),1)) = 0$.

$$r(s,n,e) = \begin{cases} 0 & \text{if } \exists i<e \exists t \leq n \exists m[p(i,m) < y(s,t,e) \text{ and} \\ & \qquad d(s,p(i,m)) \neq d(s-1,p(i,m))] \\ 0 & \text{if } \exists t \leq n[p(f(s),1) < y(s,t,e)] \\ 1 & \text{otherwise.} \end{cases}$$

We have three casas for $d(s,p(e,i,m))$:

Case d1: $t(s,i,e) > m$. We set

$$d(s,p(e,i,m)) = \begin{cases} d(s-1,p(e,i,m)) \text{ if} \\ \quad \exists u \leq e \exists v<m(s,u)[r(s,v,u) = 1 \\ \quad \text{and } p(e,i,m) < y(s,v,u)] \\ 0 \quad \text{otherwise.} \end{cases}$$

Case d2: $t(s,i,e) = m$. We set

$$d(s,p(e,i,m)) = \begin{cases} 0 & \text{if } c(s,i) = 2 \\ & \quad \text{and for all } j \; i \;\; d(s-1,j) = \\ & \qquad U(y_b(s,j,e)) \\ & \quad \text{and it is not the case} \\ & \qquad \text{that } \exists u\_e \exists v \; m(s,u) \text{ such} \\ & \qquad \text{that } r(s,v,u) = 1 \text{ and} \\ & \qquad\quad p(e,i,m) \;\; y(s,v,u) \\ d(s-1,p(e,i,m)) & \text{otherwise.} \end{cases}$$

Case d3: $t(s,i,e) < m$. We set

$$d(s,p(e,i,m)) = d(s-1,p(e,i,m)).$$

The construction is concluded by setting $d(s,w) = d(s-1,w)$ for all w not equal to $p(f(s),1)$ or $p(e,i,m)$ for some e, i and m. Let $d(n) = \lim_{s \to \infty} d(s,n)$ for each n. Let D be the set whose characteristic function is $1-d$; D is recursively enumerable since $n \in D$ if and only if $\exists s(d(s,n) = 0)$. We make two remarks which will be needed later:

N1: For all s, n and e, $r(s,n,e) = 0$ implies $r(s,n+1,e) = 0$

N2: For all s, n and e, $y(s,n,e) = 0$ implies $n \geq m(s,e))$.

(N2 can be seen by considering cases in the definition of $m(s,e)$.)

Lemma 1: Let $y(s,n,e) > 0$, $m(s,e) > n$ and $p(f(s),1) \geq y(s,t,e)$ for all $t \leq n$. Let $d(s,p(i,m)) = d(s-1,p(i,m))$ for all i, t and m such that $i < e$, $t \leq n$ and $p(i,m) < y(s,t,e)$. Then $y(s,n,e) = y(s+1,n,e)$ and $m(s+1,e) > n$.

Proof: Since $y(s,n,e) > 0$, we have that it is the computation number of $\varphi_e^{D[s,y(s,n,e)]}(n)$. If $d(s,j) = d(s-1,j)$ for all $j < y(s,n,e)$, then $y(s+1,n,e) = y(s,n,e)$. The hypothesis of our lemma tells us that $r(s,n,e) = 1$. But then it follows from the definition of $d(s,j)$ that $d(s,j) = d(s-1,j)$ for all $j < y(s,n,e)$, since $m(s,e) > n$.

Note that N1 makes clear that the above argument also works for any $t < n$. Thus we have for all $t \leq n$ that $y(s,t,e) = y(s+1,t,e)$. Suppose $m(s+1,e) \leq n$. Then $m(s+1,e) < m(s,e)$, and Case m1 of the definition of $m(s+1,e)$ holds. But then we have a t (namely, $m(s+1,e)$) such that $t \leq n$ and $y(s,t,e) \neq y(s+1,t,e)$.

For each e, we say that e is stable if $\lim_{s\to\infty} y(s,n,e)$ exists and is postive for all n. Since there are infinitely many e that are not numbers of systems of equations, there are infinitely many unstable e. Let $\{e_0 < e_1 < e_2 \ldots\}$ be the set of all nonstable e. For each j, let n_j be the least n such that $\lim_{s\to\infty} y(s,n,e_j)$ either does not exist or is zero. Lemma 2 is our basic combinatorial principle.

Lemma 2: For each k and v, there is an $s \geq v$ such that for all $j < k$, $m(s,e_j) \leq n_j$ or $r(s,n_j,e_j) = 0$ or $y(s,n_j,e_j) = 0$.

Proof: Fix k and v. We suppose there is no s with the desired properties and then define an infinite descending sequence of natural numbers.

We propose the following system of equations as a means of defining two functions, $S(t)$ and $M(t)$, simultaneously by induction:

$$S(0) = v$$

$$M(t) = \text{the least } y < k \text{ such that } n_j < m(S(t),e_j) \text{ and } r(S(t),n_j,e_j) = 1 \text{ and } y(S(t),n_j,e_j) > 0$$

$$S(t+1) = \text{the least } s \text{ such that } \exists m(s \geq S(t) \text{ and } m < y(S(t),n_{M(t)},e_{M(t)}) \text{ and } d(s,m) \neq d(S(t)-1,m)).$$

Clearly $S(0) \geq v$. Suppose $t \geq 0$, $S(t)$ is well-defined and $S(t) \geq v$. Then $M(t) < k$, since we have supposed the lemma to be false. Thus $y(S(t), n_{M(t)}, e_{M(t)}) > 0$ and $\lim_{s \to \infty} y(s, n_{M(t)}, e_{M(t)})$ does not exist or does equal 0. Then there must be an $s > S(t)$ such that $m < y(S(t), n_{M(t)}, e_{M(t)})$ and $d(s-1,m) \neq d(S(t)-1,m)$. But then $S(t+1)$ is well-defined and $S(t+1) \geq v$.

For each $t \geq 0$, let $u(t+1)$ be the least m such that $d(S(t)-1,m) \neq d(S(t+1),m)$. Fix $t > 0$. We show $u(t+1) < u(t)$. Since we know $u(t+1) < y(S(t), n_{M(t)}, e_{M(t)})$, it sufices to show $y(S(t), n_{M(t)}, e_{M(t)}) \leq u(t)$. It follows from the definition of S that $d(w,m) \neq d(S(t-1)-1,m)$ whenever $S(t) > w \geq S(t-1)$ and $m < y(S(t-1), n_{M(t-1)}, e_{M(t-1)})$. Consequently, $d(S(t),u(t)) \neq d(S(t)-1,u(t))$. First suppose $u(t) = p(f(S(t)),1)$. Then $y(S(t), n_{M(t)}, e_{M(t)}) \leq u(t)$, since $r(S(t), n_{M(t)}, e_{M(t)}) = 1$. Now suppose $u(t) = p(i,f,m)$ for some i, f, and m. Let $s = S(t)$, $n = n_{M(t)}$ and $e = e_{M(t)}$. If $i \geq e$, then the definition of r tells us that $y(s,n,e) \leq u(t)$, since $r(s,n,e) = 1$. If $i \geq e$, the definition of d tells ua that $y(s,n,e) \leq u(t)$, since $n < m(s,e)$, $r(s,n,e)=1$ and $d(s,u(t)) \neq d(s-1,u(t))$.

Let us consider the following sentences:

$A(e)$: If e is stable, then the set $\{m(s,e)\colon s \geq 0\}$ is finite.

$B(e)$: The set $\{m\colon d(p_e^m) = 0\}$ is recursive in B.

We will prove $A(e)$ and $B(e)$ are true for all e by means of a simultaneous induction on e. From $A(e)$ true for all e it will follow that $\underline{c} \not\leq \underline{d}$; from $B(e)$, that $\underline{d} \not\leq \underline{b}$.

Lemma 3: For all $i < e$ $B(i)$ true implies that $A(e)$ is true.

Proof: We know $\underline{c} \not\leq \underline{b}$. We suppose that $B(i)$ is true for each $i<e$ and that $A(e)$ is false, and we show $\underline{c} \leq \underline{b}$. Thus the set $\{m(s,e): s \geq 0\}$ is infinite, and for each n, $\lim_{s\to\infty} y(s,n,e)$ exists and is positive. Let $R(n,s) = 1$ if $m(s,e)$ n ,and for all i, t and m, $p(i,m) < y(s,t,e)$ and $i<e$ and $t \leq n$ implies that $d(p(i,m)) = d(s-1,p(i,m))$, and for all z and t, $z \in B$ and $p(z,1) < y(s,t,e)$ and $t \leq n$ implies that $d(s-1,p(z,1)) = 0$; and equal 0 otherwise. (Recall that $z \in B$ if and only if $d(p(z,1)) = 0$.) Since $B(i)$ is true for all $i<e$, it follows that R is recursive in B. Since the set $\{m(s,e): s \geq 0\}$ is infinite, and since $\lim_{s\to\infty} y(s,n,e)$ exists and is positive for each n, it follows that for all n there is an s such that $R(n,s) = 1$. Let $w(n)$ be the least s such that $R(n,s) = 1$. Then w is recursive in B, and for each n, $w(n+1) \geq w(n)$. Fix n. We show $\lim_{s\to\infty} y(s,n,e) = y(w(n),n,e)$. Let s be such that $s \geq w(n)$ and $y(w(n),n,e) = y(s,n,e)$ and $R(n,s) = 1$. Since $m(s,e) > n$, it follows from remark N2 that $y(s,t,e) > 0$ for all $t \leq n$. We know f enumerates B without repitions. It follows from the definition of R that $p(f(s),1) \geq y(s,t,e)$ for all $t \leq n$. But then lemma 1 tells us that $y(s+1,t,e) = y(s,t,e)$ for all $t \leq n$, and that $m(s+1,e) > n$. Thus $y(w(n),n,e) = y(s+1,n,e)$ and $R(n,s+1) = 1$. It follows that $\lim_{s\to\infty} y(s,n,e) = y(w(n),n,e)$. Finally we show $\lim_{s\to\infty} c(s,n) = U(y(w(n),n,e))$ for all n. If this last is true, then C is recursive in B, since $[\lim_{s\to\infty} c(s,n)]-1$ is the characteristic function of C; and since w is recursive in B. Fix n and suppose $\lim_{s\to\infty} c(s,n) \neq U(y(w(n),n,e))$. There must be an s^* such that for all $s \geq s^*$, $c(s,n) = \lim_{s\to\infty} c(s,n) \neq U(y(w(n),n,e)) =$

$U(y(s,n,e))$. Fix $s > s^*$ and suppose $m(s-1,e) \leq m(s^*,e)+n$. If Case m1 holds, then $m(s,e) \leq m(s^*,e)+n$. If Case m2 holds and $n < 2m(s-1,e)+s$, then $m(s,e) \leq m(s^*,e)+n$, since $c(s,n) \neq U(y(s,n,e))$. If $n \geq 2m(s-1,e)+s$, then $m(s,e) \geq n$. Thus $m(s,e) \leq m(s^*,e)+n$ for all $s \geq s^*$. This last is impossible, since the set $\{m(s,e): s \geq 0\}$ is infinite.

Suppose $\varphi_e^B(n)$ is defined for all n. Then $\lim_{s\to\infty} y_b(s,n,e)$ exists for each n. and $\lim_{s\to\infty} t(s,i,e)$ exists for each i. In addition, $\lim_{s\to\infty} t(s,i,e)$ (regarded as a function of i) is recursive in B. All this is clear from the definition of y_b and t.

Lemma 4: If $\varphi_e^B(n)$ is defined for all n and for all $u \leq e$, $A(u)$ is true, then $d(p(e,i,\lim_{s} t(s,i,e))) = 0$ for only finitely many i.

Proof: We suppose the lemma is false and show C is recursive in B. Our first claim is that $\varphi_e^B = d$. Fix j. Let s be so large that $d(j) = d(w-1,j)$ and $U(y_b(w,j,e)) = \varphi_e^B(j)$ for all $w \geq s$. Let $t(i) = \lim_{s\to\infty} t(s,i,e)$ for all i. Let w and i be such that $j < i$, $w \geq s$, $c(w,i) \neq 2$, $0 = d(w,p(e,i,t(i))) \neq d(w-1,p(e,i,t(i)))$ and $t(i) = t(w,i,e)$. It follows from Case 2 of the definition of d that $d(w-1,j) = U(y_b(s,j,e))$, since $j < i$. But then $d(j) = \varphi_e^B(j)$, since $w \geq s$, and our first claim is proved.

Our second claim is that for all sufficiently large i, $d(p(e,i,t(i))) = \lim_{s\to\infty} c(s,i)$. Our first and second claims, together with the fact that t is recursive in B, imply that C is recursive in B. Our second claim is a conquence of lemma 2 and the fact that for all $u \leq e$, $A(u)$ is true. If $u \leq e$ and u is stable, then

$A(u)$ tells ua that the set $\{ m(s,u) : s \geq 0\}$ is finite. If $u \leq e$ and u is stable, let $m(u)$ be the largest member of $\{ m(s,u) : s \geq 0\}$. If $u \leq e$ and u is non-stable, then $u = e_i$ for some i; let $m(u) = n_i$. Recall that n_i is the least witness to the fact that e_i is unstable. Thus $\lim_{s\to\infty} y(s,v,u)$ exists if $u \leq e$ and $v < m(u)$. Let y be so large that $y(s,v,u) \leq y$ if $s \geq 0$, $u \leq e$ and $v < m(u)$. Fix $i \geq y$. We show $d(p(e,i,t(i))) = \lim_{s\to\infty} c(s,i)$. If $\lim_{s\to\infty} c(s,i) = 1$, then $c(s,i) = 1$ for all s, and $d(s,p(e,i,t(i)))=1$ for all s. Suppose $\lim_{s\to\infty} c(s,i) = 2$. Let w be so large that $c(s,i) = 2$, $t(s,i,e) = t(i)$, and for all $j < i$ $d(s-1,j) = U(y_b(s,j,e))$ for all $s \geq w$. The existence of w follows from our first claim. By lemma 2, there is an $s \geq w$ such that for all $u \leq e$, if u is non-stable (hence equal to e_i for some i), then either $m(s,u) < n_i$ or $r(s,n_i,u) = 0$ or $y(s,n_i,u) = 0$. We show $d(s,p(e,i,t(i))) = 0$. Case d2 is such that we need only show it is not the case that $\exists u \leq e \exists v < m(s,u)[r(s,v,u) = 1$ and $p(e,i,t(i)) < y(s,v,u)]$. Fix $u \leq e$ and $v < m(s,u)$. Suppose u is stable. Then $y(s,v,u) \leq y \leq i \leq p(e,i,t(i))$. Suppose u is non-stable. Let $u = e_i$. If $v < n_i$, then $v < m(u)$ and $y(s,v,u) \leq y \leq p(e,i,t(i))$. Suppose $v \geq n_i$. Then $m(s,u) > n_i$, and consequently, $r(s,n_i,u) = 0$ or $y(s,n_i,u) = 0$. If $r(s,n_i,u) = 0$, then $r(s,v,u) = 0$ by remark N1, since $n_i \geq v$. Suppose $y(s,n_i,u) = 0$. Then $n_i \geq m(s,u)$ by remark N2. But $m(s,u) > n_i$, since we have supposed $v \geq n_i$.

Lemma 5: If for all $u \leq e$ $A(u)$ is true, then $B(e)$ is true.
Proof: Our first claim is (L) and its proof is identical to the proof of the second claim of lemma 2. (L): $\exists y$ such that for all m,x and i, $m < t(x,i,e)$ and $p(e,i,m) \geq y$ implies $d(p(e,i,m) = 0$.

We are ready to prove $B(e)$. We will use only lemma 4 and (L). First we suppose that $\varphi_e^B(n)$ is undefined for some n; let N be the least such n. It follows from the definition of y_b that $\lim_{s\to\infty} t(s,i,e)=\infty$ for all $i\geq N$. Let y have the property described by (L). Then for all i and m, $i\geq N$ and $p(e,i,m)\geq y$ implies $d(p(e,i,m))=0$. If $i<N$, then $\lim_{s\to\infty} t(s,i,e)$ is finite; consequently, the set $\{p(e,i,m): d(p(e,i,m))=0 \text{ and } i<N\}$ is finite by Case d3 of the definition of d. Now $d(p_e^n)=0$ only if $p_e^n=p(e,i,m)$ for some i and m or $p_e^n=p(f(s),1)$. It follows that the set $\{n: d(p_e^n)=0\}$ is recursive, hence recursive in B. Now suppose $\varphi_e^B(n)$ is defined for all n. Then $\lim_{s\to\infty} t(s,i,e)$ is finite for all i. Let this limit be $t(i)$. Recall that $t(s+1,i,e)\geq t(s,i,e)$ for all s and i. It follows from (L) that for all i and m, $m<t(i)$ and $p(e,i,m)\geq y$ implies $d(p(e,i,m))=0$. By lemma 4 the set $\{p(e,i,t(i)): d(p(e,i,t(i)))=0\}$ is finite. By Case d3, $d(p(e,i,m))=1$ if $m>t(i)$. Since $t(i)$ is recursive in B, it follows that the set $\{n: d(p_e^n)=0\}$ is recursive in B.

Lemmas 3 and 5 constitute a proof that $A(e)$ and $B(e)$ are true for all e.

Lemma 6: C is not recursive in D.

Proof: Suppose $\lim_{s\to\infty} c(s,n)=\varphi_e^D(n)$ for all n. Then e is stable, and by $A(e)$, the set $\{m(s,e): s\geq 0\}$ has a greatest member, say M. Let w be so large that $\lim_{s\to\infty} c(s,n)=c(s,n)=U(y(s,n,e))$ when $s\geq w$ and $n\leq M$. If $s\geq w$, then either

(a): $c(s,n)\neq U(y(s,n,e))$ for some $n\leq m(s,e)$
or
(b): $m(s,e) = m(s-1,e)+s$.

If (a) holds, then $M<m(s,e)$. Thus (b) must hold for infinitely

many s. But then $M < m(s,e)$ for some s, which contradicts the definition of M.

Lemma 7: D is not recursive in B.

Proof: Suppose $d(n) = \varphi_e^B(n)$ for all n. Then $\lim_{s\to\infty} t(s,i,e)$ is finite for each i. Let $t(i) = \lim_{s\to\infty} t(s,i,e)$. By lemma 4, there is an n such that $d(p(e,i,t(i))) = 1$ for each $i \geq n$. We shall find an i such that $i \geq n$ and $d(p(e,i,t(i))) = 0$. We proceed as in lemma 4. Define y as in lemma 4. Fix $i \geq y+n$. Define w and s as in lemma 4. Then the final argument of lemma 4 tells that $d(s,p(e,i,t(i))) = 0$.

Let $\underline{d}$ be the degree of D. It follows from lemmas 6 and 7 that $\underline{c} \nleq \underline{d}$ and $\underline{d} \nleq \underline{b}$. We have $\underline{b} \leq \underline{d}$, since $d(p(m,1)) = 0$ if and only if $m \in B$. It remains only to show that D is recursive in C. We give an intutive description of a procedure for computing D from C. Our description is such that the translation of E into a system of equation that define D recursively in C is not diffuclt. We need a recursive function Q such that

$$Q(u,v,s,e,i,m) = \begin{cases} 1 & \text{if } u \leq e \\ & \quad v < m(s,u) \\ & \quad r(s,v,u) = 1 \\ & \quad p(e,i,m) < y(s,v,u) \\ 0 & \text{otherwise.} \end{cases}$$

We also need a function R such that

$$R(s,e,n) = \begin{cases} 1 & \text{if for all } u,i,m \text{ and } t,\ u < e \text{ and } t \leq n \text{ and} \\ & \qquad p(u,i,m) < y(s,t,e) \text{ implies that} \\ & \qquad d(s-1,p(u,i,m)) = d(p(u,i,m)), \\ & \text{and for all } i \text{ and } t,\ i \in B \text{ and } t \leq n \text{ and} \\ & \qquad p(i,1) < y(s,t,e) \text{ implies that} \\ & \qquad d(s-1,p(i,1) = d(p(i,1)); \\ 0 & \text{otherwise.} \end{cases}$$

We need the next two lemmas to describe E:

Lemma 8: $Q(u,v,s,e,i,m) = R(s,u,v) = 1$ implies that for all $w \geq s, Q(u,v,w,e,i,m) = 1$.

Proof: We proceed with an induction on $w \geq s$. Fix $w \geq s$ and suppose Q and R hold for that w. Recall that f, the recursive function whose range is B, is 1-1. It follows from lemma 1 that $v < m(w+1,u)$ and that $y(w,t,u) = y(w+1,t,u)$ for all $t \leq v$. But then $R(w+1,u,v)$ is 1 and $r(w+1,v,u) = 1$, and consequently, $Q(u,v,w+1,e,i,m) = 1$.

Lemma 9: For all w there is an $s \geq w$ such that for all u and v, $Q(u,v,s,e,i,m) = 1$ implies $R(s,u,v) = 1$.

Proof: Fix w,e,i and m. For each $u \leq e$, define $m(u)$ as in the second half of the proof of lemma 4. Define y as in lemma 4. Thus $y(s,v,u) \leq y$ if $s > 0$, $u \leq e$ and $v < m(u)$. It is safe to suppose that w is so large that for all $s > w$ and $v \leq y$, $d(s-1,v) = d(v)$. By lemma i there is an $s > w$ such that for any non-stable $u \leq e$, we have either $m(s,u) \leq n_j$ or $r(s,n_j,u) = 0$ or $y(s,n_j,u) = 0$, where $u = e_j$; recall that if u is non-stable, then $m(u) = n_j$. Fix u and v and suppose $Q(u,v,s,e,i,m) = 1$. We show $R(s,u,v) = 1$. First we suppose u is stable. Then $v < m(s,u) \leq m(u)$ since Q is 1. Consequently for all n and t, $t \leq v$ and $n < y(s,t,u)$ implies that $d(s-1,n) = d(n)$, since $s > w$. But then $R(s,u,v) = 1$. Now we suppose u is non-stable. Let $u = e_j$. Then $m(u) = n(j)$. If $m(s,u) \leq m(u)$, then $v < m(u)$, and as above, $R(s,u,v) = 1$. Thus $r(s,m(u),u) = 0$ or $y(s,m(u),u) = 0$. If the former, then $v < m(u)$, since $r(s,v,u) = 1$; this last follows from N1. Suppose the latter is zero; then it follows from N2 that $m(u) \geq m(s,u)$. But then $v < m(u)$.

We return to the description of procedure E. Fix z. We compute d(z) with the help of C. If z is not of the form p(b,1) or p(e,i,m), then $d(z) = 1$. If $z = p(b,1)$, then $d(z) = 0$ if and only if $b \in B$. Suppose $z = p(e,i,m)$. Then the set $\{s: t(s,i,e) = j\}$ is recursive in B, since $t(s,i,e) \leq t(s+1,i,e)$ forall s. We consider three cases:

(a): for all s, $t(s,i,e) < m$;

(b): there exists an s such that $t(s,i,e) > m$;

(c): $\lim_{s \to \infty} t(s,i,e) = m$.

With the help of B, we can decide which of the cases holds; note that the monotonicity of t(s,i,e), regarded as a function of s, is vital. If (a) holds, then case d3 tells us that $d(z) = 1$. Suppose (b) holds. Let w^* be such that $t(s,i,e) > m$ for all $s \geq w^*$. We can determine w^* with the help of B. Suppose $d(s,z) = 1$ for all $s < w^*$. Then $d(z) = 0$ if and only if there exists an s such that for all u and v, $s \geq w^*$ and $Q(u,v,s,e,i,m) = 0$, since case d1 applies when $s \geq w^*$. By lemma 9, there is an $s \geq w^*$ such that for all u and v, $Q(u,v,s,e,i,m) = 1$ implies $R(s,u,v) = 1$. We can find s if we know the value of d(x) for some of the following x: $x = p(u,j,k)$, where $u < e$; $x = p(j,1)$, where $j \in B$. (Recall that $d(p(j,1)) = 0$ if and only if $j \in B$.) If $Q(u,v,s,e,i,m) = 0$ for all u and v, then $d(z) = 0$; we only have to check $u \leq e$ and $v < m(s,u)$. Suppose for some u and v, $Q(u,v,s,e,i,m) = 1$. Then we have $R(s,u,v) = 1$ by definition of s. By lemma 8 it follows that $Q(u,v,w,e,i,m) = 1$ for all $w \geq s$. But then $d(z) = 0$ if and only if $d(w,z) = 0$ for some $w < s$. Finally, we suppose (c) holds.

Let w' be such that $t(s,i,e) = m$ for all $s \geq w'$. We can determine w' with the help of B. suppose $d(s,z) = 1$ for all $s < w'$. Then $d(z) = 0$ if and only if there exists an $s \geq w'$ such that $c(s,i) = 2$ and for all u and v $Q(u,v,s,e,i,m) = 0$ and for all $j < i$ $d(s-1,j) = U(y_b(s,j,e)))$, since Case d2 applies when $s \geq w'$. If $j < i$, then $y_b(s,j,e) = \lim_{s\to\infty} y_b(s,j,e)$ for all $s \leq w'$, since $t(w',i,e) = \lim_{s\to\infty} t(s,i,e) = m$. Let $v^* \geq w'$ be such that for all $s \geq v^*$, $c(s,i) = \lim_{s\to\infty} c(s,i)$ and for all $j < i$, $d(s-1,j) = d(j)$. We can determine v^* with the help of C and the values of $d(x)$ for $x \quad z$, since $j < i$ implies that $j < p(e,i,m)$. If $\lim_{s\to\infty} c(s,i) = 1$ or $d(j) \neq U(\lim_{s\to\infty} y_b(s,j,e)$ for some $j < i$, then $d(z) = 1$. Suppose this last hypothesis is false. Suppose also that $d(s,z) = 1$ for all $s < v^*$. Then $d(z) = 0$ if and only if there exists an $s \geq v^*$ such that for all u and v, $Q(u,v,s,e,i,m) = 0$, since Case d2 applies when $s \geq v^*$. We now continue as in case (b). With the help of lemmas 8 and 9, $d(z)$ is easily determined.

That completes our computation of $d(z)$ from C. To find $d(z)$, we had to know the value of $d(x)$ for infinitely many x: $x<z$; $x=p(u, j, k)$, where $u < e$; $x = p(j, 1)$ where $j \in B$. We used very heavily the fact that B is recursive in C.

We combined this principle with a further combinatorial principle expressed by Lemma 4 in order to show the recursively enumerable degrees are dense. Lemma 4 was needed to show D is not recursive in B. The prethinking which inspired Lemma 4 may be described as follows. C is not recursive in B. So let us keep planting members of C in D until D looks enough like C to guarantee that D is not recursive in B. But let us not plant members of C in D with utter abandon because we wish to have C not recursive in D. At the same time let us plant B in D with utter abandon so that B will be recursive in D. We plant a member of C in D when we set $d(s, p(e, i, t(s, i, e))) = 0$; this happens only if $c(s, i) = 2$. (Recall $c(s, i) = 2$ only if $i \in C$.) In order to prevent us from planting too much of C in D, we must have a method of unplanting members of C already planted in D. Suppose we have planted $i \in C$ in D; that is, we set $d(s, p(e, i, t(s, i, e))) = 0$. If for some $w > s, t(w,i,e) > t(s,i,e)$, then Case d1 or d2 may give us a chance of setting $d(w, p(e, i, 1 + t(s, i, e))) = 0$; if this last happens, we have unplanted i. We have the opportunity to unplant i if and only if $\lim_{s\to\infty} t(s, i, e) > t(s, i, e)$. This happens if and only if $\varphi_e^B(i)$ is undefined or unequal to $y_b(s, i, e)$. If for some i, $\varphi_e^B(i)$ is undefined, there is no need to plant any member of C in D. If $\varphi_e^B(i)$ is defined for all i, then Lemma 4 tells us that we do not permanently plant infinitely much of C in D. Of course the proof of Lemma 4 turns on the fact that C is not recursive in B.

Many of the more recent results in the theory of recursive functions have made extensive use of the infinite-injury priority method. There are still a large number of open questions in this theory and the solution of them will probably use the priority method a great deal. Among the open questions are the following: 1) Is the theory of ordering of r.e. degrees decideable? 2) Is the ordering of r.e. degrees greater than a given degree isomorphic to the ordering of r.e. degrees? 3) Is the theory of ordering of r.e. degrees equivalent in some manner to the theory of the ordering of degrees of arithmetical sets?

BIBLIOGRAPHY

Davis, Martin. _Computability and Unsolvability_. New York: McGraw-Hill Book Company, Inc., 1958.

----------, editor. _The Undecidable_. Hewlett, New York: Raven Press Books, Ltd., 1965.

Friedberg, Richard M. "Two Recursively Enumerable Sets of Incomparable Degrees of Unsolvability," _Proceedings of the National Academy of Science_, XLIII (1957), 236-238.

Kleene, Stephen Cole. _Introduction to Metamathematics_. Princeton, New Jersey: D. Van Nostrand Company, Inc., 1952.

Mendelson, Elliott. _Introduction to Mathematical Logic_. Princeton, New Jersey: D. Van Nostrand Company, Inc., 1964.

Rogers, Hartley, Jr. _Theory of Recursive Functions and Effective Computability_. New York: McGraw-Hill Book Company, Inc., 1967.

Sacks, Gerald E. _Degrees of Unsolvability_. Princeton, New Jersey: Princeton University Press, 1966.

---------- "The Recursively Enumerable Degrees are Dense," _Annals of Mathematics_, LXXX (1964), 300-312.