AN INTRODUCTION TO RECURSION THEORY

1. Informal computability

Recursion theory is the abstract theory of computations. All of the computations which we consider will be, at least in theory, performable in a finite length of time. This means that the objects with which we compute must be finite objects, i.e., objects which can be specified by a finite amount of information. For example, a natural number is a finite object, since it can be specified by giving the Arabic numeral designating that number. On the other hand, a real number is generally not a finite object, since to specify it we must, say, give each of its infinitely many decimal places.

A space is an infinite set X of finite objects such that, given a finite object x, we can decide whether or not x belongs to X. The most basic space is the set $N = \{0,1,2,\ldots\}$ of natural numbers.

For the rest of this section let X, Y be any spaces.

A partial function from X to Y is a mapping f such that $\mathrm{dom}\ f \subseteq X$ and $\mathrm{ran}\ f \subseteq Y$. A partial function f from X to Y is said to be total if $\mathrm{dom}\ f = X$.

An algorithm from X to Y is a rule which when applied to an input $x \in X$ goes through a sequence of computation steps, possibly terminating with an output $y \in Y$. The algorithm must specify how to obtain the next step from the steps already completed and the input. The process of obtaining new steps may go on forever without producing an output.

The set $\mathrm{Alg}(X,Y)$ of all algorithms from X to Y is a space. With each algorithm $I \in \mathrm{Alg}(X,Y)$ we associate a partial function $[I]$ from X to Y in the following way: $\mathrm{dom}\,[I]$ is the set of all inputs $x \in X$ for which I gives an output, in which case $[I](x)$ is this output.

A partial function f from X to Y is recursive if $f = [I]$ for some $I \in \mathrm{Alg}(X,Y)$. It has become standard usage to call such functions partial recursive functions rather than "recursive partial functions". A total recursive function (or just recursive function) is a partial recursive function which is total.

A set $A \subseteq X$ is recursive if its characteristic function $\chi_A : X \longrightarrow N$ is recursive, where

$$\chi_A(x) = \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{if } x \notin A. \end{cases}$$

Equivalently, A is recursive if there is an algorithm by which, given a member x of X, we can decide whether $x \in A$ or $x \notin A$.

2. Isomorphisms

In this section we continue to use X and Y for spaces.

An *isomorphism* from X onto Y is a bijection $f : X \longrightarrow Y$ such that both f and f^{-1} are recursive. If such an isomorphism exists, we say that X is *isomorphic* to Y.

The relation of being isomorphic is evidently an equivalence relation on the class of all spaces. Furthermore, it turns out that all spaces are isomorphic. To prove this result we need some lemmas.

If $A \subseteq X$, then a *listing* of A is a 1-1 recursive function $f : N \longrightarrow X$ such that ran f = A.

LEMMA 2.1. If $f : N \longrightarrow X$ is a listing of X, then f is an isomorphism from N onto X.

Proof. We need only show that f^{-1} is recursive. Given $x \in X$, we compute f(0), f(1), ... until we find an n such that f(n) = x. Then $f^{-1}(x) = n$. Q.E.D.

LEMMA 2.2. If A is an infinite subset of X, and $f : N \longrightarrow X$ is recursive with range A, then A has a listing.

Proof. Define $g : N \longrightarrow X$ inductively as follows: g(n) = f(m), where m is the smallest number such that f(m) is distinct from each of g(0), g(1), ..., g(n-1). Then g is a listing of A. Q.E.D.

LEMMA 2.3. If X has a listing, then any space Y included in X has a listing.

<u>Proof</u>. Let f be a listing of X. Pick $y_o \in Y$ and define $g : N \longrightarrow Y$ by

$$g(n) = \begin{cases} f(n) & \text{if } f(n) \in Y, \\ y_o & \text{otherwise.} \end{cases}$$

Then g is recursive and ran g = Y, so Y has a listing by Lemma 2.2. Q.E.D.

If x is a finite object, we may write down a complete description of it. We may suppose that the symbols used in this description are chosen from a finite set Γ independent of x. Let $Sq(\Gamma)$ be the set of all finite sequences of elements of Γ. Then $Sq(\Gamma)$ is a space and all of our descriptions belong to this space.

LEMMA 2.4. The space $Sq(\Gamma)$ has a listing.

<u>Proof</u>. Suppose that $\Gamma = \{x_1, x_2, \ldots, x_r\}$. Let $f(0) = f(1) = \emptyset$ (the empty sequence). If $n > 1$, let the prime power decomposition of n be $p_1^{n_1} p_2^{n_2} \ldots p_k^{n_k}$, where $p_1 < p_2 < \ldots < p_k$ and all of the exponents are positive. If all of the exponents are $\leq r$, let $f(n) = x_{n_1} x_{n_2} \ldots x_{n_k}$; otherwise, let $f(n) = \emptyset$. Then f is a recursive function from N onto $Sq(\Gamma)$; so $Sq(\Gamma)$ has a listing by Lemma 2.2.

Q.E.D.

THEOREM 2.5 (Isomorphism Theorem). Any two spaces are isomorphic.

<u>Proof</u>. It suffices to show that any space X is isomorphic to N. Let $Y \subseteq Sq(\Gamma)$ be the space of all descriptions of elements of X. By Lemmas 2.3 and 2.4 Y has a listing $g : N \longrightarrow Y$. Define $f : Y \longrightarrow X$ by letting $f(y)$ be the object described by y. Then f is a recursive function from Y onto X, hence the composition $f \circ g$ is a recursive function from N onto X. By Lemma 2.2 X has a listing, and this listing is an isomorphism by Lemma 2.1. Q.E.D.

3. Primitive recursive functions

The following functions are called <u>initial functions</u>:

(I) The <u>zero function</u>: $Z(x) = 0$.

(II) The <u>successor function</u>: $S(x) = x+1$.

(III) The <u>projection functions</u>: $P_i^n(x_1,\ldots,x_n) = x_i$ $(1 \leq i \leq n)$.

The following are rules for obtaining new functions from given functions:

(IV) <u>Substitution</u>:

$$f(x_1,\ldots,x_n) = g(h_1(x_1,\ldots,x_n),\ldots,h_m(x_1,\ldots,x_n)).$$

We say that f is obtained by substitution from the functions $g,h_1,\ldots,h_m$.

(V) <u>Recursion</u>:

$$f(x_1,\ldots,x_n,0) = g(x_1,\ldots,x_n),$$

$$f(x_1,\ldots,x_n,y+1) = h(x_1,\ldots,x_n,y,f(x_1,\ldots,x_n,y)).$$

When $n = 0$ these equations take the form

$$f(0) = k,$$

$$f(y+1) = h(y,f(y)),$$

where k is a fixed natural number. We shall say that f is obtained from g and h (or, in the case $n = 0$, from h alone) by recursion.

A function $f : N^n \longrightarrow N$, $n \geq 1$, is said to be <u>primitive recursive</u> if it can be obtained from the initial functions by any finite number of substitutions and recursions, i.e., if there is a finite sequence of functions $f_1, f_2, \ldots, f_k$, called a <u>derivation</u> of f, such that $f_k = f$, and, for each $i = 1, \ldots, k$, either f_i is an initial function or f_i is obtained from earlier functions in the sequence by substitution or recursion. For example, the function $f(x,y) = x+y$ has the following derivation:

$$f_1(x) = x \qquad \text{by (III)}$$

$$f_2(x) = x+1 \qquad \text{by (II)}$$

$$f_3(x,y,z) = z \qquad \text{by (III)}$$

$$f_4(x,y,z) = f_2(f_3(x,y,z)) \qquad \text{by (IV)}$$

$$f_5(x,0) = f_1(x)$$

$$f_5(x,y+1) = f_4(x,y,f_5(x,y)) \qquad \text{by (V).}$$

A relation (predicate) $R \subseteq N^n$, $n \geq 1$, is said to be <u>primitive recursive</u> if its characteristic function is primitive recursive.

Clearly all initial functions are recursive, and any function obtained from recursive functions by substitution or recursion is recursive. Hence all primitive recursive functions and relations are recursive. The converse is false:

THEOREM 3.1. There exists a function $f : N \longrightarrow N$ which is recursive but not primitive recursive.

Proof. Each derivation of a primitive recursive function has a description, which is a finite sequence of symbols chosen from a fixed finite alphabet. The set of all descriptions of (derivations of) unary primitive recursive functions is obviously a space. Fix a listing $\{D_n : n \in N\}$ of this space, and define f by

$$f(n) = d_n(n) + 1,$$

where d_n is the primitive recursive function determined by D_n. Clearly f is recursive; but since $f \neq d_n$ for all n, f cannot be primitive recursive. Q.E.D.

The above method of constructing the function f is known as diagonalization.

We next describe ways of demonstrating primitive recursiveness, and collect examples of primitive recursive functions and predicates.

THEOREM 3.2. If $R, h_1, \ldots, h_m$ are primitive recursive, and Q is defined by

$$Q(x_1, \ldots, x_n) \iff R(h_1(x_1, \ldots, x_n), \ldots, h_m(x_1, \ldots, x_n)),$$

then Q is primitive recursive.

Proof. χ_Q is obtained by substitution from the functions $\chi_R, h_1, \ldots, h_m$:

$$\chi_Q(x_1, \ldots, x_n) = \chi_R(h_1(x_1, \ldots, x_n), \ldots, h_m(x_1, \ldots, x_n)).$$

Q.E.D.

THEOREM 3.3. Let g be a primitive recursive function, and let f be obtained from g by the identity

$$f(x_1,\ldots,x_n) = g(x_{i_1},\ldots,x_{i_m}),$$

where $1 \leq i_k \leq n$ for $k = 1,\ldots,m$ and the i_k need not be distinct. Then f is primitive recursive.

Similarly for predicates, i.e., if R is primitive recursive, and Q is defined by

$$Q(x_1,\ldots,x_n) \iff R(x_{i_1},\ldots,x_{i_m}),$$

then Q is primitive recursive.

Proof. Simply note that

$$f(x_1,\ldots,x_n) = g(P^n_{i_1}(x_1,\ldots,x_n),\ldots,P^n_{i_m}(x_1,\ldots,x_n))$$

and

$$Q(x_1,\ldots,x_n) \iff R(P^n_{i_1}(x_1,\ldots,x_n),\ldots,P^n_{i_m}(x_1,\ldots,x_n)).$$

Q.E.D.

There are three important special cases of Theorem 3.3:

(1) Permutation of variables; e.g.,

$$f(x,y,z) = g(y,z,x).$$

(2) Identification of variables; e.g.,

$$f(x,y) = g(x,y,x).$$

(3) Addition of dummy variables; e.g.,

$$f(x,y) = g(x).$$

THEOREM 3.4. The following functions and predicates are primitive recursive:

(a) The constant function $C^n_k(x_1,\ldots,x_n) = k$, where k is some fixed natural number.

(b) $x+y$.

(c) $x \cdot y$.

(d) x^y.

(e) $x!$.

(f) $$pd(x) = \begin{cases} x-1 & \text{if } x > 0, \\ 0 & \text{if } x = 0. \end{cases}$$

(g) $$x \dot{-} y = \begin{cases} x-y & \text{if } x \geq y, \\ 0 & \text{if } x < y. \end{cases}$$

(h) $|x-y|$.

(i) $\min(x,y)$.

(j) $\max(x,y)$.

(k) $$sg(x) = \begin{cases} 0 & \text{if } x = 0, \\ 1 & \text{if } x \neq 0. \end{cases}$$

(l) $$\overline{sg}(x) = \begin{cases} 1 & \text{if } x = 0, \\ 0 & \text{if } x \neq 0. \end{cases}$$

(m) $x < y$.

(n) $x \leq y$.

(o) $x = y$.

Proof. (a) By induction on k: $C_o^n(x_1,\ldots,x_n) = Z(x_1)$, $C_{k+1}^n(x_1,\ldots,x_n) = S(C_k^n(x_1,\ldots,x_n))$.

(b) Already noted.

(c) $x\cdot 0 = 0$, $x\cdot(y+1) = x\cdot y + x$.

(d) $x^o = 1$, $x^{y+1} = x^y\cdot x$.

(e) $0! = 1$, $(y+1)! = (y!)\cdot(y+1)$.

(f) $pd(0) = 0$, $pd(y+1) = y$.

(g) $x \dot{-} 0 = x$, $x \dot{-} (y+1) = pd(x \dot{-} y)$.

(h) $|x-y| = (x \dot{-} y)+(y \dot{-} x)$.

(i) $\min(x,y) = x \dot{-} (x \dot{-} y)$.

(j) $\max(x,y) = y+(x \dot{-} y)$.

(k) $sg(0) = 0$, $sg(y+1) = 1$.

(l) $\overline{sg}(x) = 1 \dot{-} sg(x)$.

(m) $\chi_<(x,y) = sg(y \dot{-} x)$.

(n) $\chi_\leq(x,y) = \overline{sg}(x \dot{-} y)$.

(o) $\chi_=(x,y) = \overline{sg}(|x-y|)$. Q.E.D.

THEOREM 3.5. Let $R_1, \ldots, R_k$ be primitive recursive predicates such that, for each $\overline{x}$, exactly one of $R_1(\overline{x}), \ldots$ $\ldots, R_k(\overline{x})$ holds. (We write $\overline{x}$ for $x_1, \ldots, x_n$.)

(a) If $f_1, \ldots, f_k$ are primitive recursive functions, and f is defined by

$$f(\overline{x}) = \begin{cases} f_1(\overline{x}) & \text{if } R_1(\overline{x}), \\ \vdots & \\ f_k(\overline{x}) & \text{if } R_k(\overline{x}), \end{cases}$$

then f is primitive recursive.

(b) If $Q_1, \ldots, Q_k$ are primitive recursive predicates, and Q is defined by

$$Q(\overline{x}) \iff \begin{cases} Q_1(\overline{x}) & \text{if } R_1(\overline{x}), \\ \vdots & \\ Q_k(\overline{x}) & \text{if } R_k(\overline{x}), \end{cases}$$

then Q is primitive recursive.

<u>Proof</u>. (a) By the identity

$$f(\overline{x}) = f_1(\overline{x}) \cdot \chi_{R_1}(\overline{x}) + \ldots . + f_k(\overline{x}) \cdot \chi_{R_k}(\overline{x}).$$

(b) Note that

$$\chi_Q(\overline{x}) = \begin{cases} \chi_{Q_1}(\overline{x}) & \text{if } R_1(\overline{x}), \\ \vdots & \\ \chi_{Q_k}(\overline{x}) & \text{if } R_k(\overline{x}), \end{cases}$$

and apply (a). Q.E.D.

THEOREM 3.6. If R and Q are primitive recursive predicates, then the predicates

$$\neg R, \quad R \wedge Q, \quad R \vee Q$$

are also primitive recursive.

<u>Proof</u>. The characteristic functions of these predicates are as follows:

$$\chi_{\neg R}(\bar{x}) = 1 \dot{-} \chi_R(\bar{x}),$$

$$\chi_{R \wedge Q}(\bar{x}) = \chi_R(\bar{x}) \cdot \chi_Q(\bar{x}),$$

$$\chi_{R \vee Q}(\bar{x}) = \max(\chi_R(\bar{x}), \chi_Q(\bar{x})). \quad \text{Q.E.D.}$$

THEOREM 3.7. If g is a primitive recursive function, then the functions

$$f_1(\bar{x},y) = \sum_{z<y} g(\bar{x},z), \qquad f_2(\bar{x},y) = \prod_{z<y} g(\bar{x},z)$$

are also primitive recursive.

<u>Proof</u>. By the identities

$$f_1(\bar{x},0) = 0, \quad f_1(\bar{x},y+1) = f_1(\bar{x},y) + g(\bar{x},y)$$

and

$$f_2(\bar{x},0) = 1, \quad f_2(\bar{x},y+1) = f_2(\bar{x},y) \cdot g(\bar{x},y). \quad \text{Q.E.D.}$$

The <u>bounded μ-operator</u> $\mu z<y(\ldots)$ is defined by

$$\mu z<y(\ldots) = \begin{cases} \text{the least } z<y \text{ such that } \ldots & \text{if } \exists z<y(\ldots), \\ y & \text{otherwise.} \end{cases}$$

THEOREM 3.8. Suppose R is a primitive recursive predicate. Then the following are primitive recursive:

(a) The predicate $Q_1(\overline{x},y) \iff \forall z<y\ R(\overline{x},z)$.

(b) The predicate $Q_2(\overline{x},y) \iff \exists z<y\ R(\overline{x},z)$.

(c) The function $f(\overline{x},y) = \mu z<y\ R(\overline{x},z)$.

Proof. (a) $\chi_{Q_1}(\overline{x},y) = \prod_{z<y} \chi_R(\overline{x},z)$.

(b) $Q_2(\overline{x},y) \iff \neg\forall z<y\ \neg R(\overline{x},z)$.

(c) $f(\overline{x},y) = \sum_{w<y} \chi_Q(\overline{x},w)$, where Q is defined by $Q(\overline{x},w) \iff \neg\exists z\leq w\ R(\overline{x},z)$. Q.E.D.

THEOREM 3.9. The following functions and predicates are primitive recursive:

(a) $qt(x,y)$ = quotient when x is divided by y.

(b) $rm(x,y)$ = remainder when x is divided by y.

(c) $x|y$ (x divides y).

(d) x is prime.

(e) p_x = the x^{th} prime number (where $p_o = 2$, $p_1 = 3$, $p_2 = 5$, etc.)

(f) $(x)_y$ = the exponent of p_y in the prime factorization of x.

<u>Proof</u>. (a) $qt(x,y) = \mu z{<}x\ (y \cdot (z+1) > x)$.

(b) $rm(x,y) = x \dot{-} y \cdot qt(x,y)$.

(c) $x|y \iff rm(y,x) = 0$.

(d) x is prime $\iff x > 1 \wedge \forall y{<}x\ (y \leq 1 \vee y \nmid x)$.

(e) $p_o = 2$,

$p_{x+1} = \mu z \leq (p_x! + 1)(z > p_x \wedge z \text{ is prime})$.

(f) $(x)_y = \mu z{<}x\ (p_y^{z+1} \nmid x)$. Q.E.D.

4. The basic results

We return now to the study of partial recursive functions. We shall limit ourselves to numerical functions and predicates, so from now on:

(i) n-ary function will mean (total) function from N^n to N.

(ii) n-ary partial function will mean partial function from N^n to N.

(iii) n-ary predicate will mean subset of N^n.

We suppose that a listing

$$\{I_e : e \in N\}$$

of the space $\mathrm{Alg}(\bigcup_{n=1}^{\infty} N^n, N)$ is fixed once and for all. The number e is called the index or Gödel number of I_e. We now define:

(a) $\varphi_e^{(n)}$ = the n-ary partial function computed by I_e.

(b) $W_e^{(n)} = \mathrm{dom}\ \varphi_e^{(n)}$.

We shall be mainly concerned with unary partial recursive functions, so for convenience we omit the superscript (1) when it occurs, i.e., we write φ_e for $\varphi_e^{(1)}$, and W_e for $W_e^{(1)}$.

Note that

$$\varphi_0^{(n)}, \varphi_1^{(n)}, \varphi_2^{(n)}, \ldots$$

is an enumeration (with repetitions) of the set of all n-ary partial recursive functions. Thus, if f is an n-ary partial recursive function, then $f = \varphi_e^{(n)}$ for some $e \in N$. Any such e is called an <u>index</u> or <u>Gödel number</u> of f. Since there are many different algorithms that compute f, we cannot say that e is <u>the</u> index of f; in fact, each partial recursive function has infinitely many indices.

We adopt a convention for equality between expressions (such as $\varphi_e(x)$) which may be undefined. We let $u = v$ hold if both u and v are defined and they have the same value, or if both u and v are undefined. In all other cases, $u \neq v$.

The <u>μ-operator</u> $\mu z(\ldots)$ is given the following meaning:

$$\mu z(\ldots) = \begin{cases} \text{the least } z \text{ such that } \ldots & \text{if } \exists z(\ldots), \\ \text{undefined} & \text{otherwise.} \end{cases}$$

We are now ready to state and prove some basic results, due to Kleene, from which virtually the whole of recursion theory can be developed.

THEOREM 4.1 (Normal Form Theorem). There is a unary recursive function U and, for each $n \geq 1$, an $(n+2)$-ary recursive predicate T_n (called the <u>Kleene T-predicate</u>) such that

$$\varphi_e^{(n)}(\bar{x}) = U(\mu z\ T_n(e,\bar{x},z)).$$

<u>Proof</u>. Fix a listing $\{(a_z,b_z): z \in N\}$ of N^2, and define U, T_n by

$$U(z) = a_z,$$

$$T_n(e,\bar{x},z) \iff I_e \text{ with input } \bar{x} \text{ yields output } a_z \text{ in fewer than } b_z \text{ steps.}$$

Q.E.D.

THEOREM 4.2 (Enumeration Theorem). For each $n \geq 1$ there is an $(n+1)$-ary partial recursive function F_n such that

$$\varphi_e^{(n)}(\bar{x}) = F_n(e,\bar{x}).$$

(F_n is called the <u>universal partial function</u> for the n-ary partial recursive functions.)

<u>Proof</u>. The identity $F_n(e,\bar{x}) = \varphi_e^{(n)}(\bar{x})$ serves as a definition of F_n. By Theorem 4.1 we have $F_n(e,\bar{x}) = U(\mu z\ T_n(e,\bar{x},z))$, so F_n is recursive. Intuitively speaking, to compute $F_n(e,\bar{x})$ we effectively recover I_e and then apply it to input $\bar{x}$ until (if ever) an output is obtained.

Q.E.D.

THEOREM 4.3 (Parameter Theorem). For each $m \geq 1$ there is a 1-1 recursive (m+1)-ary function s^m such that

$$\varphi_e^{(m+n)}(\bar{x},\bar{y}) = \varphi_{s^m(e,\bar{x})}^{(n)}(\bar{y}).$$

(Here $\bar{x}$ are parameters being held fixed.)

Proof. The algorithm $I_{s^m(e,\bar{x})}$ on input $\bar{y}$ first obtains I_e and then applies it to input $(\bar{x},\bar{y})$. Q.E.D.

In a very rough sense the Enumeration and Parameter Theorems are inverses to one another, since the former "pushes indices up" as variables, while the latter "pulls variables down" as indices. (The Parameter Theorem is also known as the "s-m-n Theorem", for historical reasons.)

We have described the class of partial recursive functions in terms of the informal and rather vague notion of algorithm. During the 1930's and since then, several formal (i.e., mathematically exact) characterizations of this class have been proposed by Turing, Kleene, Church, Post, Markov, and others. All these characterizations turned out to be equivalent, in the sense that the same class $\mathcal{R}$ of partial functions is obtained in each case. Furthermore, a wide variety of particular partial recursive functions have been studied, and each has been demonstrated to belong to $\mathcal{R}$. On the basis of this evidence most mathematicians are led to accept Church's Thesis, which is the claim that $\mathcal{R}$ coincides with the class of partial recursive (i.e., intuitively computable) functions.

Once the class of partial recursive functions is identified with $\mathcal{R}$, Theorems 4.1 and 4.3 can be proved formally. This gives the additional information that U, T_n (of Theorem 4.1), and s^m (of Theorem 4.3) are in fact primitive recursive.

5. Unsolvable problems

With each n-ary predicate R we associate a *decision problem*:

> Find an algorithm by which, given any $\bar{x} \in N^n$, we can decide whether $R(\bar{x})$ holds or not.

If such an algorithm exists, i.e., if R is recursive, the decision problem for R is said to be *solvable*; otherwise, the decision problem for R is said to be *unsolvable*.

Let

$$K = \{x: \varphi_x(x) \text{ is defined}\} = \{x: x \in W_x\}.$$

The following theorem is a basic unsolvability result:

THEOREM 5.1. K is not recursive.

Proof. We use a diagonal argument. The function

$$f(x) = \begin{cases} \varphi_x(x) + 1 & \text{if } x \in K, \\ 0 & \text{if } x \notin K \end{cases}$$

cannot be recursive because $f \neq \varphi_e$ for each e. Consequently K is not recursive. Q.E.D.

COROLLARY 5.2. The predicate $\{(x,y): x \in W_y\}$ is not recursive.

Proof. Note that $x \in K \iff (x,x) \in R$, where $R = \{(x,y) : x \in W_y\}$. If R were recursive, then K would be recursive, contrary to Theorem 5.1. Q.E.D.

Corollary 5.2 is often described as the unsolvability of the halting problem.

THEOREM 5.3 (Rice's Theorem). Let $\mathcal{R}_1$ be the class of all unary partial recursive functions. Suppose that $\mathcal{C} \subseteq \mathcal{R}_1$, and $\mathcal{C} \neq \emptyset, \mathcal{R}_1$. Then the set $\{x : \varphi_x \in \mathcal{C}\}$ is not recursive.

Proof. Let $A = \{x : \varphi_x \in \mathcal{C}\}$. Clearly $A \neq \emptyset, N$, and

$$(*) \qquad (x \in A \ \& \ \varphi_x = \varphi_y) \implies y \in A.$$

Choose e_o such that $\varphi_{e_o}(y)$ is undefined for all y. There are two cases to consider.

Case 1: $e_o \notin A$. Choose $e_1 \in A$ and define a binary partial recursive function f by

$$f(x,y) = \begin{cases} \varphi_{e_1}(y) & \text{if } x \in K, \\ \text{undefined} & \text{if } x \notin K. \end{cases}$$

By the Parameter Theorem there is a unary recursive function g such that $\varphi_{g(x)}(y) = f(x,y)$. Now

$$x \in K \implies \varphi_{g(x)} = \varphi_{e_1} \implies g(x) \in A,$$

$$x \notin K \implies \varphi_{g(x)} = \varphi_{e_o} \implies g(x) \notin A,$$

the last implication in each line being justified by (*). Thus $x \in K \iff g(x) \in A$, so by Theorem 5.1 A cannot be recursive.

Case 2: $e_0 \in A$. Choose $e_1 \in N \setminus A$ and define f and g as before. Then $x \in K \iff g(x) \notin A$, so again A cannot be recursive. Q.E.D.

COROLLARY 5.4. None of the sets defined below is recursive:

(a) $K_1 = \{x: W_x \neq \emptyset\}$.

(b) $Fin = \{x: W_x \text{ is finite}\}$.

(c) $Inf = N \setminus Fin = \{x: W_x \text{ is infinite}\}$.

(d) $Tot = \{x: \varphi_x \text{ is total}\} = \{x: W_x = N\}$.

(e) $Con = \{x: \varphi_x \text{ is total and constant}\}$.

(f) $Cof = \{x: W_x \text{ is cofinite}\}$.

(g) $Rec = \{x: W_x \text{ is recursive}\}$.

(h) $Ext = \{x: \varphi_x \text{ is extendible to a total recursive function}\}$.

Proof. These are all special cases of Rice's Theorem. Q.E.D.

6. Recursive enumerability

An n-ary predicate R is said to be <u>recursively enumerable</u> (abbreviated <u>r.e.</u>) if there exists an (n+1)-ary recursive predicate Q such that

$$R(\overline{x}) \iff \exists y\, Q(\overline{x},y)$$

for all $\overline{x}$.

As an example, the equivalence

$$x \in W_y \iff \exists n\, (I_y \text{ with input } x \text{ yields an output in } < n \text{ steps})$$

shows that $\{(x,y): x \in W_y\}$ is an r.e. predicate. Similarly, the set $K = \{x: x \in W_x\}$ is r.e.

Notice that every recursive predicate R is r.e.; for $R(\overline{x}) \iff \exists y\, Q(\overline{x},y)$, where Q is the recursive predicate defined by $Q(\overline{x},y) \iff R(\overline{x})$. The converse is false, as the above examples show.

Let R be an (n+1)-ary predicate, where $n \geq 1$. A <u>selector</u> for R is an n-ary partial function f such that

$$f(\overline{x}) \text{ is defined} \iff \exists y\, R(\overline{x},y),$$

and in this case $R(\overline{x},f(\overline{x}))$. Thus f selects a $y = f(\overline{x})$ such that $R(\overline{x},y)$, provided that such a y exists.

THEOREM 6.1 (Selection Theorem). If R is an $(n+1)$-ary r.e. predicate ($n \geq 1$), then there is a recursive selector for R.

Proof. There is a recursive predicate Q such that

$$R(\overline{x},y) \iff \exists z\ Q(\overline{x},y,z).$$

Let $\{(a_m,b_m): m \in N\}$ be a listing of N^2. We define an n-ary partial recursive function f by $f(\overline{x}) = a_m$, where m is the first number such that $Q(\overline{x},a_m,b_m)$. (If no such m exists, then $f(\overline{x})$ is undefined.) It is clear that f is a selector for R. Q.E.D.

COROLLARY 6.2. An n-ary predicate R is r.e. if and only if $R = W_e^{(n)}$ for some e. (Any such e is called an *index* of R.)

Proof. $\Longrightarrow$: Suppose R is r.e., so that $R(\overline{x}) \iff \exists y\ Q(\overline{x},y)$ for some recursive predicate Q. By Theorem 6.1 there is a recursive selector $\varphi_e^{(n)}$ for Q, and it follows that $R = \operatorname{dom} \varphi_e^{(n)} = W_e^{(n)}$.

$\Longleftarrow$: If $R = W_e^{(n)}$, then R is r.e. since

$$R(\overline{x}) \iff \exists m\ (I_e \text{ with input } \overline{x} \text{ yields an output in } < m \text{ steps}).$$

Q.E.D.

COROLLARY 6.3. If Q is r.e. and R is defined by

$$R(\overline{x}) \iff \exists y\ Q(\overline{x},y),$$

then R is r.e.

Proof. Let $\varphi_e^{(n)}$ be a recursive selector for Q. Then $R = W_e^{(n)}$, so R is r.e. by Corollary 6.2. Q.E.D.

COROLLARY 6.4. If R is an n-ary r.e. predicate, then there exists an (n+1)-ary primitive recursive predicate Q such that

$$R(\overline{x}) \iff \exists y\, Q(\overline{x}, y).$$

Proof. By Corollary 6.2 $R = W_e^{(n)}$ for some e. By the Normal Form Theorem we have

$$R(\overline{x}) \iff \varphi_e^{(n)}(\overline{x}) \text{ is defined } \iff \exists y\, T_n(e, \overline{x}, y),$$

where T_n (the Kleene T-predicate) is primitive recursive. Define Q by

$$Q(\overline{x}, y) \iff T_n(e, \overline{x}, y).$$

Then Q is primitive recursive and $R(\overline{x}) \iff \exists y\, Q(\overline{x}, y)$.

Q.E.D.

The graph of an n-ary partial function f is the (n+1)-ary predicate R defined by

$$R(\overline{x}, y) \iff f(\overline{x}) = y.$$

THEOREM 6.5 (Graph Theorem). A partial function is recursive if and only if its graph is r.e.

Proof. The graph of $\varphi_e^{(n)}$ is r.e. by virtue of the equivalence

$$\varphi_e^{(n)}(\overline{x}) = y \iff \exists m\ (I_e \text{ with input } \overline{x} \text{ yields output } y \text{ in } < m \text{ steps}).$$

The converse follows from the Selection Theorem, since the only selector for the graph of f is f itself. Q.E.D.

Theorem 6.5 gives a characterization of partial recursive functions in terms of r.e. predicates. A related characterization of recursive predicates is as follows:

THEOREM 6.6 (Negation Theorem). A predicate R is recursive if and only if both R and $\neg R$ are r.e.

Proof. $\Longrightarrow$: If R is recursive, then $\neg R$ is also recursive; so R and $\neg R$ are r.e.

$\Longleftarrow$: Suppose that both R and $\neg R$ are r.e.; say $R(\overline{x}) \iff \exists y\, Q_1(\overline{x},y)$ and $\neg R(\overline{x}) \iff \exists y\, Q_2(\overline{x},y)$, where Q_1 and Q_2 are recursive. For each $\overline{x}$, either $R(\overline{x})$ or $\neg R(\overline{x})$; so there is a y such that either $Q_1(\overline{x},y)$ or $Q_2(\overline{x},y)$. Hence we may define a recursive function f by

$$f(\overline{x}) = \mu y\ (Q_1(\overline{x},y) \vee Q_2(\overline{x},y)).$$

Then $R(\overline{x}) \iff Q_1(\overline{x},f(\overline{x}))$; so R is recursive. Q.E.D.

For the rest of this section we confine our attention to unary predicates, i.e., subsets of $\mathbb{N}$.

THEOREM 6.7. Let $A \subseteq N$. Then the following are equivalent:

(a) A is r.e.

(b) $A = \emptyset$ or A is the range of a unary primitive recursive function.

(c) A is the range of a partial recursive function.

Proof. (a) $\Longrightarrow$ (b): Suppose that A is r.e. and $A \neq \emptyset$. By Corollary 6.4 there is a binary primitive recursive predicate Q such that $x \in A \iff \exists y\, Q(x,y)$. Choose $a \in A$ and define f by

$$f(z) = \begin{cases} (z)_1 & \text{if } Q((z)_1,(z)_2), \\ a & \text{otherwise,} \end{cases}$$

where $(z)_t$ = the exponent of p_t in the prime factorization of z (cf. Theorem 3.9). Clearly f is primitive recursive and ran $f = A$.

(b) $\Longrightarrow$ (c): Trivial.

(c) $\Longrightarrow$ (a): Suppose that A is the range of an n-ary partial recursive function g. Then

$$x \in A \iff \exists y_1 \ldots \exists y_n\, (g(y_1,\ldots,y_n) = x).$$

The predicate in brackets on the right is r.e. by the Graph Theorem; so A is r.e. by Corollary 6.3 (applied repeatedly).

Q.E.D.

THEOREM 6.8 (Listing Theorem). **An infinite set** $A \subseteq N$ **is r.e. if and only if it has a listing, i.e.,** A **is the range of a 1-1 recursive function** $f : N \longrightarrow N$.

Proof. By Theorem 6.7 and Lemma 2.2. Q.E.D.

There is a similar characterization of infinite recursive sets:

THEOREM 6.9. An infinite set $A \subseteq N$ is recursive if and only if it is the range of a strictly increasing recursive function $f : N \longrightarrow N$.

Proof. $\Longrightarrow$: Suppose that A is recursive and infinite. Define $f : N \longrightarrow N$ by

$$f(0) = \mu x\ (x \in A),$$

$$f(n+1) = \mu x\ (x \in A \text{ and } x > f(n)).$$

Then f is a strictly increasing recursive function with range A.

$\Longleftarrow$: Suppose that $A = \text{ran } f$, where $f : N \longrightarrow N$ is strictly increasing and recursive. It is clear that if $f(n) = x$ then $n \leq x$. Hence

$$x \in A \iff \exists n \leq x\ (f(n) = x);$$

so A is recursive. Q.E.D.

The above theorem may be applied to prove

THEOREM 6.10. Every infinite r.e. set $A \subseteq N$ has an infinite recursive subset.

Proof. Let $f : N \longrightarrow N$ be a recursive function with range A. Define $g : N \longrightarrow N$ by

$$g(0) = f(0),$$

$$g(n+1) = f(\mu x[f(x) > g(n)]).$$

Clearly g is recursive and strictly increasing; so by dint of Theorem 6.9 ran g is an infinite recursive subset of A.

Q.E.D.

Since K is r.e. but nonrecursive, it follows from Theorem 6.6 that the set $\neg K = N \setminus K$ is not r.e. Using this result we now prove:

THEOREM 6.11 (Rice-Shapiro). Let $\mathcal{C} \subseteq \mathcal{R}_1$ = the class of all unary partial recursive functions. Assume that the set $\{x : \varphi_x \in \mathcal{C}\}$ is r.e. Then for any $f \in \mathcal{R}_1$,

$$f \in \mathcal{C} \iff \exists g \in \mathcal{C}\ (g \text{ is a finite function and } g \subseteq f).$$

(Note. By a finite function we mean a function whose domain is finite.)

Proof. Let $A = \{x : \varphi_x \in \mathcal{C}\}$. We are given that A is r.e. We shall show that if either implication in the statement of the theorem is false for some $f \in \mathcal{R}_1$, then there is a unary recursive function s such that $x \in \neg K \iff s(x) \in A$. This would show that $\neg K$ is r.e., a contradiction.

<u>Case 1</u>: $f \in \mathcal{C}$ but $g \notin \mathcal{C}$ for all finite $g \subseteq f$. Choose an e such that $K = W_e$, and define a binary partial recursive function h by

$$h(x,y) = \begin{cases} \text{undefined} & \text{if } I_e \text{ on input } x \text{ yields} \\ & \text{an output in } < y \text{ steps,} \\ f(y) & \text{otherwise.} \end{cases}$$

The Parameter Theorem provides a recursive function s such that $h(x,y) = \varphi_{s(x)}(y)$. Then $x \in \neg K \iff s(x) \in A$, showing that $\neg K$ is r.e., a contradiction.

<u>Case 2</u>: there is a finite function $g \in \mathcal{C}$ with $g \subseteq f$, but $f \notin \mathcal{C}$. Define h by

$$h(x,y) = \begin{cases} f(y) & \text{if } x \in K \text{ or } y \in \text{dom } g, \\ \text{undefined} & \text{otherwise,} \end{cases}$$

and proceed as in Case 1. Q.E.D.

We now have a method which can be used to give quick proofs of non-recursive enumerability:

> If the conclusion of the Rice-Shapiro Theorem is false for a certain $\mathcal{C} \subseteq \mathcal{R}_1$, then the set $\{x\colon \varphi_x \in \mathcal{C}\}$ is not r.e.

We illustrate the method with

COROLLARY 6.12. None of the following sets (cf. Corollary 5.4) is r.e.:

(a) Fin = $\{x\colon W_x$ is finite$\}$.

(b) Inf = $\neg$Fin = $\{x\colon W_x$ is infinite$\}$.

(c) Tot = $\{x\colon \varphi_x$ is total$\}$.

(d) $\neg$Tot = $\{x\colon \varphi_x$ is not total$\}$.

Proof. Each of these sets is of the form $\{x\colon \varphi_x \in \mathcal{C}\}$ for some $\mathcal{C} \subseteq \mathcal{R}_1$ for which the conclusion of the Rice-Shapiro Theorem is false. Q.E.D.

Rice's Theorem follows easily from the Rice-Shapiro Theorem (so the latter is a generalization of the former). We omit the details.

7. Diophantine predicates and Matiyasevich's Theorem

In what follows polynomial will mean polynomial with integer (possibly negative) coefficients.

A predicate R is said to be diophantine if there exists a polynomial $p(\overline{x},y_1,\ldots,y_m)$ such that

$$R(\overline{x}) \iff \exists y_1 \ldots \exists y_m \, (p(\overline{x},y_1,\ldots,y_m) = 0)$$

for all $\overline{x}$. (Here $m \geq 0$ and the quantifiers $\exists y_1$, ..., $\exists y_m$ are taken as ranging over N.)

THEOREM 7.1. Every diophantine predicate is r.e.

Proof. By Corollary 6.3. Q.E.D.

In 1970 Matiyasevich (following earlier work of Davis, Robinson and Putnam) proved the converse of Theorem 7.1, namely

THEOREM 7.2 (Matiyasevich's Theorem). Every r.e. predicate is diophantine.

The proof of this result is far too long to present here. The major part of the proof consists in showing that diophantine predicates are closed under bounded universal quantification; i.e., if $R(\overline{x},y)$ is diophantine then so is the predicate $\forall z<y \; R(\overline{x},z)$. (It is an easy exercise to show that r.e. predicates are closed under bounded universal quantification.)

A surprising consequence of Theorem 7.2 is

THEOREM 7.3. There is a <u>universal polynomial</u> which generates all r.e. sets; i.e., a polynomial $p(z,x,y_1,\ldots,y_m)$ such that for all $e \in N$,

$$x \in W_e \iff \exists y_1 \ldots \exists y_m \, (p(e,x,y_1,\ldots,y_m) = 0).$$

<u>Proof</u>. The predicate "$x \in W_z$" is r.e., so by Matiyasevich's Theorem there exists a polynomial $p(z,x,y_1,\ldots,y_m)$ such that

$$x \in W_z \iff \exists y_1 \ldots \exists y_m \, (p(z,x,y_1,\ldots,y_m) = 0)$$

for all x,z. Q.E.D.

A <u>diophantine equation</u> is an equation of the form

$$p(x_1,\ldots,x_n) = 0,$$

where $p(x_1,\ldots,x_n)$ is a polynomial.

<u>Hilbert's tenth problem</u> asks:

(1) Is there an algorithm by which, given any diophantine equation, we can decide whether or not that equation has a solution in the integers?

A related question is:

(2) Is there an algorithm by which, given any diophantine equation, we can decide whether or not that equation has a solution in the natural numbers?

We shall see that a negative solution to Hilbert's tenth problem is easily derived from Matiyasevich's Theorem. First a lemma:

LEMMA 7.4. Questions (1) and (2) are equivalent, i.e., the answer to (1) is "yes" if and only if the answer to (2) is "yes".

Proof. $\Rightarrow$: By a well-known theorem of Lagrange, every natural number is the sum of four squares. So a given diophantine equation

$$(*) \qquad p(x_1,\ldots,x_n) = 0$$

has a solution in natural numbers if and only if the diophantine equation

$$p(s_1^2+t_1^2+u_1^2+v_1^2,\ldots,s_n^2+t_n^2+u_n^2+v_n^2) = 0$$

has a solution in integers.

$\Leftarrow$: Every integer is the difference of two natural numbers, so (*) has a solution in integers if and only if the diophantine equation

$$p(y_1-z_1,\ldots,y_n-z_n) = 0$$

has a solution in natural numbers. Q.E.D.

THEOREM 7.5. Hilbert's tenth problem is unsolvable, i.e., there is no algorithm for testing diophantine equations for possession of a solution in integers.

Proof. In view of Lemma 7.4 it suffices to show that there is no algorithm for testing diophantine equations for solvability in N. If such an algorithm existed, then every diophantine set would be recursive; so by Matiyasevich's Theorem every r.e. set would be recursive, a contradiction.

Q.E.D.

8. The Recursion Theorem

The following theorem of Kleene is one of the most elegant and important results in the subject. In spirit it is a strong extension of rule (V) for the primitive recursive functions because it allows us to conclude that certain implicitly defined functions are actually recursive. The proof, which uses only the Parameter Theorem, is best visualized as a diagonal argument that fails.

THEOREM 8.1 (Recursion Theorem). For every unary recursive function f there exists an n such that

$$\varphi_n = \varphi_{f(n)}.$$

(We call n a <u>fixed point</u> of f; for this reason the theorem is also known as the "Fixed Point Theorem".)

<u>Proof</u>. By the Parameter Theorem there is a recursive function d such that

$$\varphi_{d(x)}(y) = \varphi_{\varphi_x(x)}(y).$$

Note that the sequence $\{\varphi_{d(x)}: x \in N\}$ is the diagonal of the matrix

$$\{\varphi_{\varphi_x(y)}: x,y \in N\},$$

where it is understood that $\varphi_{\varphi_x(y)}$ denotes the totally undefined function if $\varphi_x(y)$ is undefined.

Given f, we "operate" on $\{\varphi_{d(x)}: x \in N\}$ to form the sequence $\{\varphi_{f(d(x))}: x \in N\}$. This sequence is one of the rows of the matrix, namely $\{\varphi_{\varphi_e(x)}: x \in N\}$, where $\varphi_e = f \circ d$. Then

$$\varphi_{f(d(e))} = \varphi_{\varphi_e(e)} = \varphi_{d(e)},$$

so $n = d(e)$ is a fixed point of f. Q.E.D.

The following corollary summarizes the way that the Recursion Theorem is often applied, in conjunction with the Parameter Theorem.

COROLLARY 8.2. Let g be a binary partial recursive function. Then there exists an n such that for all y,

$$\varphi_n(y) = g(n,y).$$

Proof. Use the Parameter Theorem to obtain a unary recursive function s such that $\varphi_{s(x)}(y) = g(x,y)$; now apply the Recursion Theorem, taking n to be a fixed point of s.

Q.E.D.

A typical application of Corollary 8.2 is that there exists an n such that $W_n = \{n\}$. (Consider a binary partial recursive function g such that $g(x,y)$ is defined if and only if $y = x$.)

We can use the Recursion Theorem to give a short proof of Rice's Theorem:

Alternative proof of Rice's Theorem. Let $\mathcal{C} \subseteq \mathcal{R}_1$ with $\mathcal{C} \neq \emptyset, \mathcal{R}_1$. We have to show that the set $A = \{x: \varphi_x \in \mathcal{C}\}$ is not recursive. Choose $a \in A$, $b \notin A$, and define

$$f(x) = \begin{cases} a & \text{if } x \notin A, \\ b & \text{if } x \in A. \end{cases}$$

Clearly f has no fixed points, so by the Recursion Theorem f cannot be recursive. Hence A is not recursive. Q.E.D.

We now prove some generalizations of the Recursion Theorem.

THEOREM 8.3. There exists a 1-1 unary recursive function n such that for any z, if φ_z is total, then

$$\varphi_{n(z)} = \varphi_{\varphi_z(n(z))}.$$

Proof. The result is already implicit in the proof of the Recursion Theorem. We first use the Parameter Theorem to get a 1-1 recursive function d such that

$$\varphi_{d(x)}(y) = \varphi_{\varphi_x(x)}(y).$$

A second application of the Parameter Theorem provides a 1-1 recursive function s such that

$$\varphi_{s(z)}(x) = \varphi_z(d(x)).$$

Then $n = d \circ s$ is our desired function. Q.E.D.

THEOREM 8.4 (Recursion Theorem with Parameters). Let f be a (k+1)-ary recursive function. Then there exists a 1-1 k-ary recursive function n_f such that for all $\overline{z}$,

$$\varphi_{n_f(\overline{z})} = \varphi_{f(n_f(\overline{z}),\overline{z})}.$$

<u>Proof</u>. By the Parameter Theorem there is a 1-1 recursive function s such that $\varphi_{s(\overline{z})}(x) = f(x,\overline{z})$. Let $n_f = n \circ s$, where n is as in Theorem 8.3. Evidently n_f possesses the desired properties. Q.E.D.

THEOREM 8.5. For every unary recursive function f there is an infinite r.e. set of fixed points of f.

<u>Proof</u>. Define a binary recursive function g by $g(x,z) = f(x)$. By Theorem 8.4 there exists a 1-1 recursive function n_g such that

$$\varphi_{n_g(z)} = \varphi_{g(n_g(z),z)} = \varphi_{f(n_g(z))}.$$

Hence $\operatorname{ran} n_g$ is an infinite r.e. set of fixed points of f.

Q.E.D.

9. Many-one reducibility and one-one reducibility

From now on, unless otherwise indicated, the word *set* will mean subset of N. We shall use A, B, C, ... to denote sets.

Given two sets A and B, we shall say that A is *many-one reducible* (*m-reducible*) to B (written $A \leq_m B$) if there is a unary recursive function f such that for all x,

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

If such an f exists and if in addition f is 1-1, then we shall say that A is *one-one reducible* (*1-reducible*) to B ($A \leq_1 B$).

The following theorem gives some simple and basic facts about $\leq_m$ and $\leq_1$.

THEOREM 9.1. (a) $\leq_m$ and $\leq_1$ are reflexive and transitive.

(b) $A \leq_m B \iff \neg A \leq_m \neg B$.

(c) $A \leq_1 B \iff \neg A \leq_1 \neg B$.

(d) If $A \leq_m B$ and B is recursive (resp. r.e.), then A is recursive (resp. r.e.).

(e) If A is recursive and $B \neq \emptyset, N$, then $A \leq_m B$.

Proof. (a)-(d) are immediate.

(e) Choose $b \in B$, $c \notin B$, and define

$$f(x) = \begin{cases} b & \text{if } x \in A, \\ c & \text{if } x \notin A. \end{cases}$$

Then f is recursive and $x \in A \iff f(x) \in B$. Q.E.D.

COROLLARY 9.2. If A is a nonrecursive r.e. set, then $\neg A \not\leq_m A$ and $A \not\leq_m \neg A$.

Proof. If $\neg A \leq_m A$, then by Theorem 9.1(d) $\neg A$ is r.e., a contradiction. For $A \not\leq_m \neg A$, use Theorem 9.1(b).

Q.E.D.

We now prove a theorem which shows again the key role played by the r.e. set K.

THEOREM 9.3. The following are equivalent:

(a) A is r.e.

(b) $A \leq_1 K$.

(c) $A \leq_m K$.

Proof. (a) $\Rightarrow$ (b): Suppose that A is r.e. Define a binary partial recursive function f by

$$f(x,y) = \begin{cases} 1 & \text{if } x \in A, \\ \text{undefined} & \text{if } x \notin A. \end{cases}$$

The Parameter Theorem gives a 1-1 recursive function s such that $\varphi_{s(x)}(y) = f(x,y)$. It is clear from the definition of f that $x \in A \iff s(x) \in K$, so $A \leq_1 K$.

(b) $\Rightarrow$ (c): Trivial.

(c) $\Rightarrow$ (a): By Theorem 9.1(d). Q.E.D.

The next result is extracted from the proof of the Rice-Shapiro Theorem.

THEOREM 9.4. If the conclusion of the Rice-Shapiro Theorem is false for a certain $\mathcal{C} \subseteq \mathcal{R}_1$, then

$$\neg K \leq_1 \{x: \varphi_x \in \mathcal{C}\}.$$

COROLLARY 9.5. If $A = K_1$, Fin, Inf, Tot, or $\neg$Tot (cf. Corollary 5.4), then $K \leq_1 A$.

Proof. By Theorems 9.4 and 9.1(c). Q.E.D.

By Theorem 9.1(a) $\leq_m$ and $\leq_1$ induce the following equivalence relations:

$$A \equiv_m B \iff (A \leq_m B \text{ and } B \leq_m A),$$

$$A \equiv_1 B \iff (A \leq_1 B \text{ and } B \leq_1 A).$$

The equivalence classes under $\equiv_m$ and $\equiv_1$ are called the <u>m-degrees</u> and <u>1-degrees</u> respectively, and are denoted by

$$\deg_m(A) = \{B: A \equiv_m B\},$$

$$\deg_1(A) = \{B: A \equiv_1 B\}.$$

For example, by Theorem 9.3 and Corollary 9.5 we have $K_1 \equiv_1 K$ and hence $\deg_1(K_1) = \deg_1(K)$.

A <u>recursive permutation</u> is a recursive bijection from N onto N. We say that A is <u>recursively isomorphic</u> to B (written $A \equiv B$) if there is a recursive permutation f such that $f(A) = B$. Clearly $\equiv$ is an equivalence relation.

The following result, due to Myhill, is an effective analogue of the classical Schröder-Bernstein Theorem for cardinal numbers.

THEOREM 9.6 (Myhill Isomorphism Theorem). The relations $\equiv$ and $\equiv_1$ coincide, i.e.,

$$A \equiv B \iff A \equiv_1 B.$$

<u>Proof</u>. $\Longrightarrow$: Trivial.

$\Longleftarrow$: Let $A \leq_1 B$ via f and $B \leq_1 A$ via g. We define a recursive permutation $h = \bigcup_{i=0}^{\infty} h_i$ by stages (where h_i

is defined at stage i) so that $h(A) = B$. Each h_i will be a finite 1-1 function satisfying

$$x \in A \iff h_i(x) \in B$$

for all $x \in \operatorname{dom} h_i$; also, $h_i \subseteq h_{i+1}$.

<u>Stage $i = 0$</u>. Let $h_o = \emptyset$.

<u>Stage $i+1 = 2x+1$</u>. If $x \in \operatorname{dom} h_i$, let $h_{i+1} = h_i$. Otherwise compute $f(x) = y_o$, $f(h_i^{-1}(y_o)) = y_1$, $f(h_i^{-1}(y_1)) = y_2$, ... until a value $f(h_i^{-1}(y_{n-1})) = y_n$ not in $\operatorname{ran} h_i$ is obtained. Then let $h_{i+1} = h_i \cup \{(x,y_n)\}$.

<u>Stage $i+1 = 2x+2$</u>. If $x \in \operatorname{ran} h_i$, let $h_{i+1} = h_i$. Otherwise compute $g(x) = z_o$, $g(h_i(z_o)) = z_1$, $g(h_i(z_1)) = z_2$, ... until a value $g(h_i(z_{m-1})) = z_m$ not in $\operatorname{dom} h_i$ is obtained. Then let $h_{i+1} = h_i \cup \{(z_m,x)\}$. Q.E.D.

The rest of this section is devoted to a brief study of m-degrees. We shall use $\underline{a}$, $\underline{b}$, $\underline{c}$, etc. for m-degrees.

The relation $\leq_m$ on sets induces a relation on m-degrees, also denoted by $\leq_m$, as follows:

$$\deg_m(A) \leq_m \deg_m(B) \iff A \leq_m B.$$

This is a legitimate definition: if $A \equiv_m A_1$ and $B \equiv_m B_1$, then $A \leq_m B \iff A_1 \leq_m B_1$. It is clear that

(1) $\underline{a} \leq_m \underline{a}$,

(2) $(\underline{a} \leq_m \underline{b}$ & $\underline{b} \leq_m \underline{c}) \implies \underline{a} \leq_m \underline{c}$,

(3) $(\underline{a} \leq_m \underline{b}$ & $\underline{b} \leq_m \underline{a}) \implies \underline{a} = \underline{b}$,

i.e., $\leq_m$ is a partial ordering of m-degrees. As usual, we write $\underline{a} <_m \underline{b}$ for $\underline{a} \leq_m \underline{b}$ & $\underline{a} \neq \underline{b}$.

THEOREM 9.7. Any two m-degrees have a least upper bound.

Proof. Let

$$A \oplus B = \{2x: x \in A\} \cup \{2x+1: x \in B\}.$$

Then $\deg_m(A \oplus B)$ is the least upper bound of $\deg_m(A)$ and $\deg_m(B)$. Q.E.D.

A partially ordered set in which any two elements have a least upper bound is called an upper semi-lattice. Theorem 9.7 thus says that the set of m-degrees forms an upper semi-lattice under $\leq_m$. It can be shown that the set of 1-degrees (which is partially ordered by $\deg_1(A) \leq_1 \deg_1(B) \iff A \leq_1 B$) does not form an upper semi-lattice.

The name <u>recursive m-degree</u> is given to any m-degree that contains a recursive set; similarly, an <u>r.e. m-degree</u> is one that contains an r.e. set. Note that any recursive (resp. r.e.) m-degree consists only of recursive (resp. r.e.) sets.

THEOREM 9.8. (a) There are exactly three recursive m-degrees, namely $\deg_m(\emptyset) = \{\emptyset\}$, $\deg_m(N) = \{N\}$, and $\{A\colon A$ is recursive and $A \neq \emptyset, N\}$. These three degrees are denoted by $\underline{o}$, $\underline{n}$, and $\underline{0}_m$, respectively.

(b) $\underline{o} \not\leq_m \underline{n}$ and $\underline{n} \not\leq_m \underline{o}$.

(c) $\underline{o} <_m \underline{0}_m$ and $\underline{n} <_m \underline{0}_m$.

(d) If $\underline{a} \neq \underline{o}, \underline{n}$, then $\underline{0}_m \leq_m \underline{a}$.

(e) If $\underline{a} \leq_m \underline{b}$ and $\underline{b}$ is r.e., then $\underline{a}$ is r.e.

(f) There is a maximum r.e. m-degree, namely $\deg_m(K)$. This degree is denoted by $\underline{0}'_m$.

<u>Proof</u>. By Theorems 9.1 and 9.3. Q.E.D.

10. Productive and creative sets

A set A is productive if there is a unary partial recursive function f, called a productive function for A, such that for all x,

$$W_x \subseteq A \implies (f(x) \text{ is defined } \& \ f(x) \in A \setminus W_x).$$

Thus, a productive set is a set which fails to be r.e. in a very strong way (its failure to be r.e. is "effectively demonstrated").

A set A is said to be creative if A is r.e. and $\neg A$ is productive. Informally, a creative set is an "effectively nonrecursive" r.e. set.

For example, the set K is creative since $\neg K$ is productive via the identity function $f(x) = x$.

Many examples of productive sets are obtained from the following theorem.

THEOREM 10.1. If A is productive and $A \leq_m B$, then B is productive.

Proof. Let A be productive via f, and let $A \leq_m B$ via g. By the Parameter Theorem there is a recursive function s such that $W_{s(x)} = g^{-1}(W_x)$. Then $h(x) = g(f(s(x)))$ is a productive function for B. Q.E.D.

COROLLARY 10.2. The sets $\neg K_1$, Fin, Inf, Tot, and $\neg$Tot are productive.

Proof. By Corollary 9.5 and Theorem 10.1. Q.E.D.

THEOREM 10.3. Any productive set has a 1-1 total productive function.

Proof. Let A be productive via f. First we obtain a total productive function q for A as follows. By the Parameter Theorem there is a recursive function s such that

$$W_{s(x)} = \begin{cases} W_x & \text{if } f(x) \text{ is defined,} \\ \emptyset & \text{otherwise.} \end{cases}$$

Define $q(x)$ to be either $f(x)$ or $f(s(x))$, whichever is computed first.

Now we convert q to a 1-1 total productive function p. The Parameter Theorem gives a recursive function t such that $W_{t(x)} = W_x \cup \{q(x)\}$. Note that

$$(*) \qquad W_x \subseteq A \implies W_{t(x)} \subseteq A.$$

Define $p(0) = q(0)$. To compute $p(x+1)$, enumerate the set $\{q(x+1), q(t(x+1)), q(t^2(x+1)), \ldots\}$ until either some y not in $\{p(0),\ldots,p(x)\}$ is found; or a repetition occurs. In the former case set $p(x+1) = y$. In the latter case, $W_{x+1} \not\subseteq A$ by $(*)$, and we can set $p(x+1) = \mu y\, [y \notin \{p(0),\ldots,p(x)\}]$. Q.E.D.

THEOREM 10.4. Every productive set contains an infinite r.e. subset.

Proof. Suppose A is productive. By Theorem 10.3 A has a 1-1 total productive function p. Let $W_n = \emptyset$, and let (by the Parameter Theorem) s be a 1-1 recursive function such that $W_{s(x)} = W_x \cup \{p(x)\}$. Then

$$B = \{p(n),\ p(s(n)),\ p(s^2(n)),\ \ldots\}$$

is an infinite r.e. subset of A. Q.E.D.

COROLLARY 10.5. Every productive set contains an infinite recursive subset.

Proof. By Theorems 10.4 and 6.10. Q.E.D.

A set A is completely productive if there is a (total) recursive function f, called a completely productive function for A, such that

$$\forall x\ (f(x) \in W_x \smallsetminus A \quad \text{or} \quad f(x) \in A \smallsetminus W_x).$$

It is immediate that a completely productive set is productive. The converse is a nontrivial theorem:

THEOREM 10.6 (Myhill). Every productive set is completely productive.

Proof. Suppose A is productive via f, where f is total (Theorem 10.3). By the Parameter Theorem there is a recursive function s such that $W_{s(x,y)} = W_y \cap \{f(x)\}$. We now apply the Recursion Theorem with Parameters to obtain a recursive function n satisfying

$$W_{n(y)} = W_{s(n(y),y)} = W_y \cap \{f(n(y))\}.$$

It is easy to check that $f \circ n$ is a completely productive function for A. Q.E.D.

The next theorem gives a characterization of productiveness in terms of the set K.

THEOREM 10.7 (Myhill). The following are equivalent:

(a) A is productive.

(b) $\neg K \leq_1 A$.

(c) $\neg K \leq_m A$.

Proof. (a) $\Rightarrow$ (b): Let p be a 1-1 total productive function for A, and let (by the Parameter Theorem) s be a recursive function satisfying

$$W_{s(x,y)} = \begin{cases} \{p(x)\} & \text{if } y \in K, \\ \emptyset & \text{if } y \notin K. \end{cases}$$

The Recursion Theorem with Parameters provides a 1-1 recursive function n such that

$$W_{n(y)} = W_{s(n(y),y)} = \begin{cases} \{p(n(y))\} & \text{if } y \in K, \\ \emptyset & \text{if } y \notin K. \end{cases}$$

Then $\neg K \leq_1 A$ via $p \circ n$.

(b) $\Longrightarrow$ (c): Trivial.

(c) $\Longrightarrow$ (a): By Theorem 10.1. Q.E.D.

A set is <u>m-complete</u> (resp. <u>1-complete</u>) if it is r.e. and any r.e. set is m-reducible (resp. 1-reducible) to it.

Combining Theorems 9.3 and 10.7 we obtain

THEOREM 10.8 (Myhill). The following are equivalent:

(a) A is creative.

(b) A is 1-complete.

(c) A is m-complete.

(d) $A \equiv_1 K$.

(e) $A \equiv_m K$.

Two sets A and B are said to be <u>recursively inseparable</u> if there is no recursive set C such that $A \subseteq C$ and $B \subseteq \neg C$. An equivalent statement is

$$\forall x \forall y \, [(A \subseteq W_x \;\&\; B \subseteq W_y \;\&\; W_x \cap W_y = \emptyset) \implies \exists z \, (z \notin W_x \cup W_y)].$$

A case of special interest is when z can be obtained from x and y in an effective way. This suggests the following definition: A and B are effectively inseparable if there exists a binary partial recursive function f, called a productive function for the pair A,B, such that

$$\forall x \forall y \, [(A \subseteq W_x \;\&\; B \subseteq W_y \;\&\; W_x \cap W_y = \emptyset) \implies (f(x,y) \text{ is defined } \;\&\; f(x,y) \notin W_x \cup W_y)].$$

Clearly, if A and B are effectively inseparable, they are also recursively inseparable.

Recursive inseparability for a pair of disjoint r.e. sets corresponds to nonrecursiveness for a single r.e. set. A pair of effectively inseparable disjoint r.e. sets is then the counterpart to a single creative set.

THEOREM 10.9. There exists a pair of effectively inseparable disjoint r.e. sets.

Proof. Let

$$A_0 = \{x : \varphi_x(x) = 0\}, \quad A_1 = \{x : \varphi_x(x) = 1\}.$$

A_0 and A_1 are evidently disjoint r.e. sets, and we shall show that they are also effectively inseparable.

Define a ternary partial recursive function h as follows: given x,y,z, start simultaneous computations of $\varphi_x(z)$ and $\varphi_y(z)$; if at some stage the value of $\varphi_x(z)$ (resp. $\varphi_y(z)$) is obtained, let $h(x,y,z) = 1$ (resp. $h(x,y,z) = 0$). By the Parameter Theorem there is a recursive function f such that $h(x,y,z) = \varphi_{f(x,y)}(z)$. Then A_0 and A_1 are effectively inseparable via f. Q.E.D.

THEOREM 10.10. If A and B are effectively inseparable disjoint r.e. sets, then A and B are both creative.

<u>Proof</u>. Let f be a productive function for the pair A,B. By the Parameter Theorem there is a recursive function s such that $W_{s(x)} = B \cup W_x$. Let $g(x) = f(a,s(x))$, where a is any index of A, i.e., $A = W_a$. Then $\neg A$ is productive via g, so A is creative. Similarly for B. Q.E.D.

THEOREM 10.11. Any pair of effectively inseparable disjoint r.e. sets has a total productive function.

<u>Proof</u>. Suppose A and B are effectively inseparable disjoint r.e. sets with productive function f. Using the Parameter Theorem we obtain two recursive functions r and s such that

$$W_{r(x,y)} = \begin{cases} A \cup W_x & \text{if } f(x,y) \text{ is defined,} \\ A & \text{otherwise,} \end{cases}$$

$$W_{s(x,y)} = \begin{cases} B \cup W_y & \text{if } f(x,y) \text{ is defined,} \\ B & \text{otherwise.} \end{cases}$$

Define $g(x,y)$ to be either $f(x,y)$ or $f(r(x,y),s(x,y))$, whichever is computed first. Then g is a total productive function for A,B. Q.E.D.

From a given pair of effectively inseparable sets, new pairs may be obtained using the following:

THEOREM 10.12. Let A and B be effectively inseparable, and let g be a unary recursive function. Then $g(A)$ and $g(B)$ are effectively inseparable.

Proof. Let f be a productive function for the pair A,B. By the Parameter Theorem there is a recursive function s such that $W_{s(z)} = g^{-1}(W_z)$. Then $g(A)$ and $g(B)$ are effectively inseparable via h, where $h(x,y) = g(f(s(x),s(y)))$.

Q.E.D.

Remark. Although the definitions of productiveness and effective inseparability mention the sets W_x, they do not depend on the particular listing $\{I_x: x \in N\}$ of algorithms chosen in advance. (Hint: if $\{J_x: x \in N\}$ is another listing of the space of algorithms, then there is a recursive permutation r such that $J_x = I_{r(x)}$ for all x.)

Our final task is to show that there are sets satisfying the following definition and hence to establish that not all nonrecursive r.e. sets are creative.

A set A is <u>simple</u> if

(a) A is r.e.,

(b) $\neg A$ is infinite,

(c) $\neg A$ contains no infinite r.e. subset.

The idea in (b),(c) of this definition is to pinpoint some features of a set that are <u>not</u> possessed by any recursive or creative set (cf. Theorem 10.4). Thus a simple set is neither recursive nor creative; and if $\underline{a}$ is the m-degree of any simple set, then $\underline{0}_m <_m \underline{a} <_m \underline{0}_m'$.

THEOREM 10.13 (Post). Simple sets exist.

<u>Proof</u>. Define a unary partial recursive function f as follows: given x, compute $\varphi_x(0)$, $\varphi_x(1)$, ... in succession (do not proceed to the computation of $\varphi_x(y+1)$ unless and until $\varphi_x(y)$ has been computed); stop if and only if a number z is found such that $\varphi_x(z) > 2x$; in that case put $f(x) = \varphi_x(z)$.

Now let A = ran f. Then A is r.e.; A intersects every infinite r.e. set; and for each n, at most n of the first 2n+1 natural numbers can occur in A. Thus A is a simple set. Q.E.D.