Now let $i_1,\dots,i_k,j,n_1,\dots,n_m$ be distinct. By changing register numbers in Q, we produce a program Q' with the following property. If Q' is executed with $x_1,\dots,x_k$ in $\mathfrak{r}i_1,\dots,\mathfrak{r}i_k$, then the machine eventually halts iff $F(x_1,\dots,x_k)$ is defined; and in this case, $F(x_1,\dots,x_k)$ is in $\mathfrak{r}j$, and the number in $\mathfrak{r}i$ is unchanged unless $i = j$ or i is one of $n_1,\dots n_m$. We write the macro of Q' as

$$F(\mathfrak{r}i_1,\dots,\mathfrak{r}i_k) \to \mathfrak{r}j \text{ USING } \mathfrak{r}n_1,\dots,\mathfrak{r}n_m.$$

As above, we generally omit USING $\mathfrak{r}n_1,\dots,\mathfrak{r}n_m$.

5. Closure Properties

We are going to show that the class of recursive functions has certain closure properties; i.e., that certain operations performed on members of the class lead to other members of the class. In later sections, we shall use these results to see that various functions are recursive.

If $1 \leq i \leq k$, we define the total k-ary function I_i^k by $I_i^k(x_1,\dots,x_k) = x_i$. Recall that every number is a 0-ary total function. The successor function Sc is defined by $Sc(x) = x + 1$. The function I_i^k, 0, and Sc are called the initial functions.

5.1. PROPOSITION. The initial functions are recursive.

Proof. The function I_i^k is computed by the program

0) MOVE $\mathfrak{r}i$ TO $\mathfrak{r}0$.

The function 0 is computed by the program

0) ZERO $\mathfrak{r}0$.

The function Sc is computed by the program

0) MOVE $\mathfrak{r}1$ TO $\mathfrak{r}0$,

1) INCREASE $\mathfrak{r}0$. □

Because our functions need not be total, we often meet expressions which may be undefined. Thus if F and G are unary, $F(G(x))$ is defined iff x is in the domain of G and $G(x)$ is in the domain of F. Suppose that X and Y are expressions which may be undefined, and which, if they are defined, represent numbers. Then $X \simeq Y$ means that either X and Y are both defined and represent the same number, or X and Y are both undefined. Note that the expression $X \simeq Y$ is always defined.

A class Φ of functions is <u>closed under composition</u> if whenever G, H_1, ..., H_n are in Φ, then so is the F defined by

$$F(\vec{x}) \simeq G(H_1(\vec{x}),\dots,H_n(\vec{x})).$$

5.2. PROPOSITION. The class of recursive functions is closed under composition.

Proof. Suppose that G, H_1, ..., H_n are recursive, and that F is defined as above, where $\vec{x}$ is $x_1,\dots,x_k$. Then F is computed by the program

$$\begin{array}{ll} 0) & H_1(\mathfrak{R}1,\dots,\mathfrak{R}k) \to \mathfrak{R}(k+1), \\ & \dots, \\ n{-}1) & H_n(\mathfrak{R}1,\dots,\mathfrak{R}k) \to \mathfrak{R}(k+n), \\ n) & G(\mathfrak{R}(k+1),\dots,\mathfrak{R}(k+n)) \to \mathfrak{R}0. \ \square \end{array}$$

If G and H are total functions, we may define a total function F by induction on y as follows:

$$F(0,\vec{x}) = G(\vec{x}),$$
$$F(y+1,\vec{x}) = H(F(y,\vec{x}),y,\vec{x}).$$

A class Φ of functions is <u>inductively closed</u> if whenever G and H are total functions in Φ and F is defined as above, then F is in Φ.

5.3. PROPOSITION. The class of recursive functions is inductively closed.

Proof. Let G and H be total recursive functions and let F be defined as above. To improve readability, assume that $\vec{x}$ is just x. Then F is computed by the program

0) $G(\mathfrak{R}2) \to \mathfrak{R}0$,

1) MOVE $\mathfrak{R}1$ TO $\mathfrak{R}3$,

2) ZERO $\mathfrak{R}1$,

3) GO TO 7,

4) $H(\mathfrak{r}0,\mathfrak{r}1,\mathfrak{r}2) \to \mathfrak{r}4$,

5) MOVE $\mathfrak{r}4$ to $\mathfrak{r}0$,

6) INCREASE $\mathfrak{r}1$,

7) DECREASE $\mathfrak{r}3$,4.

For suppose that we start the machine with y in $\mathfrak{r}1$ and x in $\mathfrak{r}2$. After 0), 1), and 2) are executed, we have $F(0,x)$, 0, x, y in $\mathfrak{r}0$, $\mathfrak{r}1$, $\mathfrak{r}2$, $\mathfrak{r}3$. If $F(z,x)$, z, x are in $\mathfrak{r}0$, $\mathfrak{r}1$, $\mathfrak{r}2$ and we execute 4), 5), and 6), then $F(z+1,x)$, $z+1$, x are in these registers. This sequence of three steps is repeated y times because of 3) and 7). Hence we finish with $F(y,x)$, y, x in $\mathfrak{r}0$, $\mathfrak{r}1$, $\mathfrak{r}2$. □

Let $X(x)$ be a statement about the number x which is defined for all values of x. We use $\mu xX(x)$ to designate the least x such that $X(x)$. If there is no x such that $X(x)$, then $\mu xX(x)$ is undefined.

A class Φ of functions is *μ–closed* if whenever R is a relation in Φ (i.e., such that χ_R is in Φ), then the function F defined by $F(\vec{x}) \simeq \mu yR(y,\vec{x})$ is in Φ.

5.4. Proposition. The class of recursive functions is μ–closed.

Proof. If R is a recursive relation and F is defined as above, then F is computed by the program

0)$0 \to \mathfrak{r}0$,

1) GO TO 3,

2) INCREASE $\mathfrak{r}0$,

3) $\chi_R(\mathfrak{r}0,\mathfrak{r}1,\ldots,\mathfrak{r}k) \to \mathfrak{r}(k+1)$,

4) DECREASE $\mathfrak{r}(k+1)$,2. □

A class of functions is *recursively closed* if it contains the initial functions and is closed under composition, inductively closed, and μ–closed. The results of this section are summarized in the proposition:

5.5. Proposition. The class of recursive functions is recursively closed. □

In the next three sections, the only fact about the class of recursive functions that we use will be the fact that it is recursively closed. This will enable us to prove in §9 that the class of recursive functions is the smallest recursively closed class.

6. Definitions of Recursive Functions

We are now going to show that certain kind of definitions of functions and relations always lead to recursive functions and relations. The simplest kind of definition of a function has the form $F(\vec{x}) \simeq$ ____, where ____ is an expression which, if defined, represents a number and which contains only previously defined symbols and variables from the sequence $\vec{x}$. Such a definition is called an <u>explicit</u> definition of F <u>in</u> <u>terms</u> <u>of</u> the symbols which appear in ____ .

6.1. PROPOSITION. If F is defined explicitly in terms of variables and names of recursive functions, then F is recursive.

Proof. We suppose that F is defined by $F(\vec{x}) \simeq$ ____ and use induction on the number of symbols in ____ . If ____ consists of just an x_i, then F is an I_i^k and hence is recursive. Otherwise, ____ is $G(X_1,\dots,X_n)$ where G is recursive. By the induction hypothesis, we may define a recursive function H_i by $H_i(\vec{x}) \simeq X_i$. Then

$$F(\vec{x}) \simeq G(H_1(\vec{x}),\dots,H_n(\vec{x}));$$

so F is recursive by 5.2. □

The simplest type of definition of a relation has the form $R(\vec{x}) \leftrightarrow$ ____ where ____ is a statement containing only previously defined symbols and variables from the sequence $\vec{x}$. In order to make sure that this defines a relation, we insist that ____ be defined for all values of $\vec{x}$. We call such a definition an <u>explicit</u> definition of R <u>in</u> <u>terms</u> <u>of</u> whatever symbols appear in ____.

6.2. PROPOSITION. If R is defined explicitly in terms of variables and names of recursive functions and relations, then R is recursive.

Proof. The definition must be $R(\vec{x}) \leftrightarrow Q(X_1,\dots,X_n)$, where Q is a