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

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

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

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^k_i by $I^k_i(x_1,...,x_k) = x_i$. Recall that every number is a 0–ary total function. The successor function Sc is defined by $Sc(x) = x + 1$. The function I^k_i, 0, and Sc are called the initial functions.

5.1. PROPOSITION. The initial functions are recursive.

Proof. The function I^k_i 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