One advantage to having the precise inductive description of the primitive recursive functions is that we can be systematic in describing them. For example, we can assign a “notation” to each such function, as follows. Use symbols zero, succ, and \( P^n_i \) for zero, successor, and the projections. Now suppose \( h \) is defined by composition from a \( k \)-place function \( f \) and \( n \)-place functions \( g_0, \ldots, g_{k-1} \), and we have assigned notations \( F, G_0, \ldots, G_{k-1} \) to the latter functions. Then, using a new symbol \( \text{Comp}_{k,n} \), we can denote the function \( h \) by \( \text{Comp}_{k,n}[F, G_0, \ldots, G_{k-1}] \).

For functions defined by primitive recursion, we can use analogous notations. Suppose the \((k+1)\)-ary function \( h \) is defined by primitive recursion from the \( k \)-ary function \( f \) and the \((k+2)\)-ary function \( g \), and the notations assigned to \( f \) and \( g \) are \( F \) and \( G \), respectively. Then the notation assigned to \( h \) is \( \text{Rec}_{k}[F, G] \).

Recall that the addition function is defined by primitive recursion as

\[
\begin{align*}
\text{add}(x_0, 0) &= P^1_0(x_0) = x_0 \\
\text{add}(x_0, y + 1) &= \text{succ}(P^2_3(x_0, y, \text{add}(x_0, y))) = \text{add}(x_0, y) + 1
\end{align*}
\]

Here the role of \( f \) is played by \( P^1_0 \), and the role of \( g \) is played by \( \text{succ}(P^2_3(x_0, y, z)) \), which is assigned the notation \( \text{Comp}_{1,3}[\text{succ}, P^3_2] \) as it is the result of defining a function by composition from the 1-ary function succ and the 3-ary function \( P^3_2 \). With this setup, we can denote the addition function by

\[
\text{Rec}_1[P^1_0, \text{Comp}_{1,3}[\text{succ}, P^3_2]].
\]

Having these notations sometimes proves useful, e.g., when enumerating primitive recursive functions.

**Problem rec.1.** Give the complete primitive recursive notation for mult.

---

**Photo Credits**

**Bibliography**