If \( f \) and \( g \) are two one-place functions of natural numbers, we can compose them: \( h(x) = g(f(x)) \). The new function \( h(x) \) is then defined by \textit{composition} from the functions \( f \) and \( g \). We’d like to generalize this to functions of more than one argument.

Here’s one way of doing this: suppose \( f \) is a \( k \)-place function, and \( g_0, \ldots, g_{k-1} \) are \( k \) functions which are all \( n \)-place. Then we can define a new \( n \)-place function \( h \) as follows:

\[
h(x_0, \ldots, x_{n-1}) = f(g_0(x_0, \ldots, x_{n-1}), \ldots, g_{k-1}(x_0, \ldots, x_{n-1}))
\]

If \( f \) and all \( g_i \) are computable, so is \( h \): To compute \( h(x_0, \ldots, x_{n-1}) \), first compute the values \( y_i = g_i(x_0, \ldots, x_{n-1}) \) for each \( i = 0, \ldots, k-1 \). Then feed these values into \( f \) to compute \( h(x_0, \ldots, x_{k-1}) = f(y_0, \ldots, y_{k-1}) \).

This may seem like an overly restrictive characterization of what happens when we compute a new function using some existing ones. For one thing, sometimes we do not use all the arguments of a function, as when we defined \( g(x, y, z) = \text{succ}(z) \) for use in the primitive recursive definition of \text{add}. Suppose we are allowed use of the following functions:

\[
P_i^n(x_0, \ldots, x_{n-1}) = x_i
\]

The functions \( P_i^n \) are called \textit{projection} functions: \( P_i^n \) is an \( n \)-place function. Then \( g \) can be defined by

\[
g(x, y, z) = \text{succ}(P_3^2(x, y, z)).
\]

Here the role of \( f \) is played by the 1-place function \text{succ}, so \( k = 1 \). And we have one 3-place function \( P_3^2 \) which plays the role of \( g_0 \). The result is a 3-place function that returns the successor of the third argument.

The projection functions also allow us to define new functions by reordering or identifying arguments. For instance, the function \( h(x) = \text{add}(x, x) \) can be defined by

\[
h(x_0) = \text{add}(P_0^1(x_0), P_0^1(x_0)).
\]

Here \( k = 2, n = 1 \), the role of \( f(y_0, y_1) \) is played by \text{add}, and the roles of \( g_0(x_0) \) and \( g_1(x_0) \) are both played by \( P_0^1(x_0) \), the one-place projection function (aka the identity function).

If \( f(y_0, y_1) \) is a function we already have, we can define the function \( h(x_0, x_1) = f(x_1, x_0) \) by

\[
h(x_0, x_1) = f(P_1^2(x_0, x_1), P_0^2(x_0, x_1)).
\]

Here \( k = 2, n = 2 \), and the roles of \( g_0 \) and \( g_1 \) are played by \( P_2^2 \) and \( P_0^2 \), respectively.

You may also worry that \( g_0, \ldots, g_{k-1} \) are all required to have the same \( \text{arity} \). (Remember that the \text{arity} of a function is the number of arguments; an \( n \)-place function has arity \( n \).) But adding the projection functions provides...
the desired flexibility. For example, suppose $f$ and $g$ are 3-place functions and $h$ is the 2-place function defined by

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

The definition of $h$ can be rewritten with the projection functions, as

$$h(x, y) = f(P_0^2(x, y), g(P_0^2(x, y), P_0^2(x, y), P_1^2(x, y)), P_1^2(x, y)).$$

Then $h$ is the composition of $f$ with $P_0^2$, $l$, and $P_1^2$, where

$$l(x, y) = g(P_0^2(x, y), P_0^2(x, y), P_1^2(x, y)),$$

i.e., $l$ is the composition of $g$ with $P_0^2$, $P_0^2$, and $P_1^2$.

**Photo Credits**

**Bibliography**