Suppose a function $h$ is defined by primitive recursion

$$
\begin{align*}
  h(0, \vec{z}) &= f(\vec{z}) \\
  h(x + 1, \vec{z}) &= g(x, h(x, \vec{z}), \vec{z})
\end{align*}
$$

and suppose the functions $f$ and $g$ are computable. Then $h(0, \vec{z})$ can obviously be computed, since it is just $f(\vec{z})$ which we assume is computable. $h(1, \vec{z})$ can then also be computed, since $1 = 0 + 1$ and so $h(1, \vec{z})$ is just

$$
g(0, h(0, \vec{z}), \vec{z}) = g(0, f(\vec{z}), \vec{z}).
$$

We can go on in this way and compute

$$
\begin{align*}
  h(2, \vec{z}) &= g(1, g(0, f(\vec{z}), \vec{z}), \vec{z}) \\
  h(3, \vec{z}) &= g(2, g(1, g(0, f(\vec{z}), \vec{z}), \vec{z}), \vec{z}) \\
  h(4, \vec{z}) &= g(3, g(2, g(1, g(0, f(\vec{z}), \vec{z}), \vec{z}), \vec{z}), \vec{z}) \\
  &\vdots
\end{align*}
$$

Thus, to compute $h(x, \vec{z})$ in general, successively compute $h(0, \vec{z})$, $h(1, \vec{z})$, $\ldots$, until we reach $h(x, \vec{z})$.

Thus, primitive recursion yields a new computable function if the functions $f$ and $g$ are computable. Composition of functions also results in a computable function if the functions $f$ and $g$ are computable.

Since the basic functions zero, succ, and $P^n_i$ are computable, and composition and primitive recursion yield computable functions from computable functions, his means that every primitive recursive function is computable.

Photo Credits

Bibliography