Simulating Primitive Recursion

Now we can show that definition by primitive recursion can be “simulated” by regular minimization using the beta function. Suppose we have \(f(z)\) and \(g(u,v,z)\). Then the function \(h(x,z)\) defined from \(f\) and \(g\) by primitive recursion is

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

We need to show that \(h\) can be defined from \(f\) and \(g\) using just composition and regular minimization, using the basic functions and functions defined from them using composition and regular minimization (such as \(\beta\)).

Lemma req.1. If \(h\) can be defined from \(f\) and \(g\) using primitive recursion, it can be defined from \(f\), \(g\), the functions zero, succ, \(P^n\), add, mult, \(\chi=\), using composition and regular minimization.

Proof. First, define an auxiliary function \(\hat{h}(x,z)\) which returns the least number \(d\) such that \(d\) codes a sequence which satisfies

1. \((d)_0 = f(z)\), and
2. for each \(i < x\), \((d)_{i+1} = g(i,(d)_i,z)\),

where now \((d)_i\) is short for \(\beta(d,i)\). In other words, \(\hat{h}\) returns the sequence \(\langle h(0,z), h(1,z), \ldots, h(x,z) \rangle\). We can write \(\hat{h}\) as

\[
\hat{h}(x,z) = \mu d \ (\beta(d,0) = f(z) \land \forall i < x \beta(d, i + 1) = g(i, \beta(d, i), z)).
\]

Note: no primitive recursion is needed here, just minimization. The function we minimize is regular because of the beta function lemma ??.

But now we have

\[
h(x,z) = \beta(\hat{h}(x,z), x),
\]

so \(h\) can be defined from the basic functions using just composition and regular minimization.

Photo Credits

Bibliography