

req.1 Simulating Primitive Recursion

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

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

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 β).

inc:req:pri:
lem:prim-rec **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_i^n , add, mult, $\chi_=$, using composition and regular minimization.*

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

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

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

$$\hat{h}(x, z) = \mu d (\beta(d, 0) = f(\vec{z}) \wedge \forall i < x \beta(d, i + 1) = g(i, \beta(d, i), \vec{z})).$$

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

$$h(x, \vec{z}) = \beta(\hat{h}(x, \vec{z}), x),$$

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

Photo Credits

Bibliography