ldf.1 Minimization

lam:ldf:min:sec The general recursive functions are those that can be obtained from the basic functions zero, succ, P^n_i by composition, primitive recursion, and regular minimization. To show that all general recursive functions are λ-definable we have to show that any function defined by regular minimization from a λ-definable function is itself λ-definable.

lam:ldf:min:lem:min **Lemma ldf.1.** *If $f(x_1, \dots, x_k, y)$ is regular and λ-definable, then g defined by*

$$g(x_1, \dots, x_k) = \mu y \; f(x_1, \dots, x_k, y) = 0$$

is also λ-definable.

Proof. Suppose the lambda term F λ-defines the regular function $f(\vec{x}, y)$. To λ-define h we use a search function and a fixpoint combinator.

$$\begin{aligned} H &\equiv \lambda \vec{x}.\, (Y \, \text{Search}) F \, \vec{x} \, \overline{0} \\ \text{Search} &\equiv \lambda g.\, \lambda f \, \vec{x} \, y.\, \text{IsZero}(f \, \vec{x} \, y) \, y \, (g \, \vec{x} (\text{Succ} \, y) \end{aligned}$$

where Y is any fixpoint combinator. Informally speaking, Search is a self-referencing function: starting with y, test whether $f \, \vec{x} \, y$ is zero: if so, return y, otherwise call itself with $\text{Succ}\, y$. Thus $(Y \, \text{Search}) F \overline{n_1} \dots \overline{n_k} \, \overline{0}$ returns the least m for which $f(n_1, \dots, n_k, m) = 0$.

Specifically, observe that

$$(Y \, \text{Search}) F \overline{n_1} \dots \overline{n_k} \, \overline{m} \twoheadrightarrow \overline{m}$$

if $f(n_1, \dots, n_k, m) = 0$, or

$$\twoheadrightarrow (Y \, \text{Search}) F \, \overline{n_1} \dots \overline{n_k} \, \overline{m+1}$$

otherwise. Since f is regular, $f(n_1, \dots, n_k, y) = 0$ for some y, and so

$$(Y \, \text{Search}) F \overline{n_1} \dots \overline{n_k} \, \overline{0} \twoheadrightarrow \overline{h(n_1, \dots, n_k)}.$$

□

Proposition ldf.2. *Every general recursive function is λ-definable.*

Proof. By **??**, all basic functions are λ-definable, and by **??**, **??**, and Lemma ldf.1, the λ-definable functions are closed under composition, primitive recursion, and regular minimization. □

Photo Credits

Bibliography