dfl.1 λ-Definable Functions are Recursive

lam:dfl:ldr:sec

Not only are all partial recursive functions λ-definable, the converse is true, too. That is, all λ-definable functions are partial recursive.

lam:dfl:ldr:thm:lambda-computable

Theorem dfl.1. *If a partial function f is λ-definable, it is partial recursive.*

Proof. We only sketch the proof. First, we arithmetize λ-terms, i.e., systematially assign Gödel numbers to λ-terms, using the usual power-of-primes coding of sequences. Then we define a partial recursive function normalize(t) operating on the Gödel number t of a lambda term as argument, and which returns the Gödel number of the normal form if it has one, or is undefined otherwise. Then define two partial recursive functions toChurch and fromChurch that maps natural numbers to and from the Gödel numbers of the corresponding Church numeral.

Using these recursive functions, we can define the function f as a partial recursive function. There is a λ-term F that λ-defines f. To compute $f(n_1, \ldots, n_k)$, first obtain the Gödel numbers of the corresponding Church numerals using toChurch(n_i), append these to ${}^{\ulcorner}F^{\urcorner}$ to obtain the Gödel number of the term $F\overline{n_1}\ldots\overline{n_k}$. Now use normalize on this Gödel number. If $f(n_1, \ldots, n_k)$ is defined, $F\overline{n_1}\ldots\overline{n_k}$ has a normal form (which must be a Church numeral), and otherwise it has no normal form (and so

$$\text{normalize}({}^{\ulcorner}F\overline{n_1}\ldots\overline{n_k}{}^{\urcorner})$$

is undefined). Finally, use fromChurch on the Gödel number of the normalized term. □

Photo Credits

Bibliography