

df.1 λ -Definable Functions are Recursive

lam:df: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:df:ldr:
thm:lambda-computable **Theorem df.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., systematically assign Gödel numbers to λ -terms as using the usual power-of-primes coding of sequences. Then we define a partial recursive function $\text{normalize}(t)$ 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 lambda term F that λ -defines f . To compute $f(n_1, \dots, n_k)$, first obtain the Gödel numbers of the corresponding Church numerals using $\text{toChurch}(n_i)$, append these to $\#F\#$ to obtain the Gödel number of the term $F\overline{n_1} \dots \overline{n_k}$. Now use normalize on this Gödel number. If $f(n_1, \dots, n_k)$ is defined, $F\overline{n_1} \dots \overline{n_k}$ has a normal form (which must be a Church numeral), and otherwise it has no normal form (and so

$$\text{normalize}(\#F\overline{n_1} \dots \overline{n_k}\#)$$

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

Photo Credits

Bibliography