

-NoValue-

rep.1 λ -definível Arithmetical Functions

lam:rep:arf:
sec
lam:rep:arf:
prop:succ-ld

Proposição rep.1. *The successor function succ is λ -definível.*

Prova. A term that λ -definem the successor function is

$$\text{Succ} \equiv \lambda a. \lambda f x. f(afx).$$

Given our conventions, this is short for

$$\text{Succ} \equiv \lambda a. \lambda f. \lambda x. (f((af)x)).$$

Succ is a function that accepts as argument a number a , and evaluates to another function, $\lambda f x. f(afx)$. That function is not itself a Church numeral. However, if the argument a is a Church numeral, it reduces to one. Consider:

$$(\lambda a. \lambda f x. f(afx)) \bar{n} \rightarrow \lambda f x. f(\bar{n}fx).$$

The embedded term $\bar{n}fx$ is a redex, since $\bar{n}$ is $\lambda f x. f^n x$. So $\bar{n}fx \rightarrow f^n x$ and so, for the entire term we have

$$\text{Succ } \bar{n} \rightarrow \lambda f x. f(f^n(x)),$$

O.s., $\overline{n+1}$.

□

Exemplo rep.2. Let's look at what happens when we apply Succ to $\bar{0}$, O.s., $\lambda f x. x$. We'll spell the terms out in full:

$$\begin{aligned} \text{Succ } \bar{0} &\equiv (\lambda a. \lambda f. \lambda x. (f((af)x)))(\lambda f. \lambda x. x) \\ &\rightarrow \lambda f. \lambda x. (f(((\lambda f. \lambda x. x)f)x)) \\ &\rightarrow \lambda f. \lambda x. (f((\lambda x. x)x)) \\ &\rightarrow \lambda f. \lambda x. (fx) \equiv \bar{1} \end{aligned}$$

Problema rep.1. The term

$$\text{Succ}' \equiv \lambda n. \lambda f x. nf(fx)$$

λ -definem the successor function. Explain why.

lam:rep:arf:
prop:add-ld

Proposição rep.3. *The addition function add is λ -definível.*

Prova. Addition is λ -defined by the terms

$$\text{Add} \equiv \lambda ab. \lambda fx. af(bfx)$$

or, alternatively,

$$\text{Add}' \equiv \lambda ab. a \text{Succ } b.$$

The first addition works as follows: Add first accept two numbers a and b . The result is a function that accepts f and x and returns $af(bfx)$. If a and b are Church numerals $\bar{n}$ and $\bar{m}$, this reduces to $f^{n+m}(x)$, which is identical to $f^n(f^m(x))$. Or, slowly:

$$\begin{aligned} (\lambda ab. \lambda fx. af(bfx)) \bar{n} \bar{m} &\rightarrow \lambda fx. \bar{n} f(\bar{m} fx) \\ &\rightarrow \lambda fx. \bar{n} f(f^m x) \\ &\rightarrow \lambda fx. f^n(f^m x) \equiv \overline{n+m}. \end{aligned}$$

The second representation of addition Add' works differently: Applied to two Church numerals $\bar{n}$ and $\bar{m}$,

$$\text{Add}' \bar{n} \bar{m} \rightarrow \bar{n} \text{Succ } \bar{m}.$$

But $\bar{n}fx$ always reduces to $f^n(x)$. So,

$$\bar{n} \text{Succ } \bar{m} \rightarrow \text{Succ}^n(\bar{m}).$$

And since Succ λ -defines the successor function, and the successor function applied n times to m gives $n + m$, this in turn reduces to $\overline{n+m}$. $\square$

Proposição rep.4. *Multiplication is λ -definível by the term*

*lam:rep:arf:
prop:mult-ld*

$$\text{Mult} \equiv \lambda ab. \lambda fx. a(bf)x$$

Prova. To see how this works, suppose we apply Mult to Church numerals $\bar{n}$ and $\bar{m}$: $\text{Mult } \bar{n} \bar{m}$ reduces to $\lambda fx. \bar{n}(\bar{m} f)x$. The term $\bar{m}f$ defines a function which applies f to its argument m times. Consequently, $\bar{n}(\bar{m}f)x$ applies the function “apply f m times” itself n times to x . In other words, we apply f to x , $n \cdot m$ times. But the resulting normal term is just the Church numeral $\overline{nm}$. $\square$

We can actually simplify this term further by η -reduction:

$$\text{Mult} \equiv \lambda ab. \lambda f. a(bf).$$

But then we first have to explain η -reduction.

Problema rep.2. Multiplication can be λ -defined by the term

$$\text{Mult}' \equiv \lambda ab. a(\text{Add } a)\bar{0}.$$

Explain why this works.

The definition of exponentiation as a λ -term is surprisingly simple:

$$\text{Exp} \equiv \lambda be. eb.$$

The first argument b is the base and the second e is the exponent. Intuitively, ef is f^e by our encoding of numbers. If you find it hard to understand, we can still define exponentiation also by iterated multiplication:

$$\text{Exp}' \equiv \lambda be. e(\text{Mult } b)\bar{1}.$$

Predecessor and subtraction on Church numeral is not as simple as we might think: it requires encoding of pairs.

Créditos das Imagens

Referências Bibliográficas