rep.1 λ-Definable Arithmetical Functions

lam:rep:arf:sec

lam:rep:arf:prop:succ-ld

Proposition rep.1. *The successor function* succ *is λ-definable.*

Proof. A term that λ-defines the successor function is

$$\mathrm{Succ} \equiv \lambda a.\, \lambda f x.\, f(a f x).$$

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

$$(\lambda a.\, \lambda f x.\, f(a f x))\, \overline{n} \to \lambda f x.\, f(\overline{n} f x).$$

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

$$\mathrm{Succ}\, \overline{n} \twoheadrightarrow \lambda f x.\, f(f^n(x)),$$

i.e., $\overline{n+1}$. □

Problem rep.1. The term

$$\mathrm{Succ}' \equiv \lambda n.\, \lambda f x.\, n f(f x)$$

λ-defines the successor function. Explain why.

lam:rep:arf:prop:add-ld

Proposition rep.2. *The addition function* add *is λ-definable.*

Proof. Addition is λ-defined by the terms

$$\mathrm{Add} \equiv \lambda ab.\, \lambda f x.\, a f(b f x)$$

or, alternatively,

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

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 $a f(b f x)$. If a and b are Church numerals $\overline{n}$ and $\overline{m}$, this reduces to $f^{n+m}(x)$, which is identical to $f^n(f^m(x))$. Or, slowly:

$$\begin{aligned}
(\lambda ab.\, \lambda f x.\, a f(b f x))\overline{n}\,\overline{m} &\to \lambda f x.\, \overline{n}\, f(\overline{m}\, f x) \\
&\to \lambda f x.\, \overline{n}\, f(f^m x) \\
&\to \lambda f x.\, f^n(f^m x) \equiv \overline{n+m}.
\end{aligned}$$

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

$$\mathrm{Add}' \overline{n}\,\overline{m} \to \overline{n}\,\mathrm{Succ}\,\overline{m}.$$

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

$$\overline{n}\,\mathrm{Succ}\,\overline{m} \twoheadrightarrow \mathrm{Succ}^n(\overline{m}).$$

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 $\overline{n+m}$. □

Proposition rep.3. *Multiplication is λ-definable by the term* lam:rep:arf:prop:mult-ld

$$\mathrm{Mult} \equiv \lambda ab.\, \lambda f x.\, a(bf)x$$

Proof. To see how this works, suppose we apply Mult to Church numerals $\overline{n}$ and $\overline{m}$: $\mathrm{Mult}\,\overline{n}\,\overline{m}$ reduces to $\lambda f x.\, \overline{n}(\overline{m}\, f)x$. The term $\overline{m} f$ defines a function which applies f to its argument m times. Consequently, $\overline{n}(\overline{m} f)x$ 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 $\overline{nm}$. □

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

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

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

$$\mathrm{Mult}' \equiv \lambda ab.\, a(\mathrm{Add}\, a)\overline{0}.$$

Explain why this works.

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

$$\mathrm{Exp} \equiv \lambda be.\, eb.$$

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:

$$\mathrm{Exp}' \equiv \lambda be.\, e(\mathrm{Mult}\, b)\overline{1}.$$

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

Photo Credits

Bibliography