

ldf.1 Introduction

lam:ldf:int:
sec

At first glance, the lambda calculus is just a very abstract calculus of expressions that represent functions and applications of them to others. Nothing in the syntax of the lambda calculus suggests that these are functions of particular kinds of objects, in particular, the syntax includes no mention of natural numbers. Its basic operations—application and lambda abstractions—are operations that apply to any function, not just functions on natural numbers.

Nevertheless, with some ingenuity, it is possible to define arithmetical functions, i.e., functions on the natural numbers, in the lambda calculus. To do this, we define, for each natural number $n \in \mathbb{N}$, a special λ -term \bar{n} , the *Church numeral* for n . (Church numerals are named for Alonzo Church.)

Definition ldf.1. If $n \in \mathbb{N}$, the corresponding *Church numeral* \bar{n} represents n :

$$\bar{n} \equiv \lambda f x. f^n(x)$$

Here, $f^n(x)$ stands for the result of applying f to x n times. For example, $\bar{0}$ is $\lambda f x. x$, and $\bar{3}$ is $\lambda f x. f(f(f x))$.

The Church numeral \bar{n} is encoded as a lambda term which represents a function accepting two arguments f and x , and returns $f^n(x)$. Church numerals are evidently in normal form.

A representation of natural numbers in the lambda calculus is only useful, of course, if we can compute with them. Computing with Church numerals in the lambda calculus means applying a λ -term F to such a Church numeral, and reducing the combined term $F \bar{n}$ to a normal form. If it always reduces to a normal form, and the normal form is always a Church numeral \bar{m} , we can think of the output of the computation as being the number m . We can then think of F as defining a function $f: \mathbb{N} \rightarrow \mathbb{N}$, namely the function such that $f(n) = m$ iff $F \bar{n} \rightarrow \bar{m}$. Because of the Church–Rosser property, normal forms are unique if they exist. So if $F \bar{n} \rightarrow \bar{m}$, there can be no other term in normal form, in particular no other Church numeral, that $F \bar{n}$ reduces to.

Conversely, given a function $f: \mathbb{N} \rightarrow \mathbb{N}$, we can ask if there is a term F that defines f in this way. In that case we say that F *λ -defines* f , and that f is *λ -definable*. We can generalize this to many-place and partial functions.

Definition ldf.2. Suppose $f: \mathbb{N}^k \rightarrow \mathbb{N}$. We say that a lambda term F *λ -defines* f if for all n_0, \dots, n_{k-1} ,

$$F \bar{n}_0 \bar{n}_1 \dots \bar{n}_{k-1} \rightarrow \overline{f(n_0, n_1, \dots, n_{k-1})}$$

if $f(n_0, \dots, n_{k-1})$ is defined, and $F \bar{n}_0 \bar{n}_1 \dots \bar{n}_{k-1}$ has no normal form otherwise.

A very simple example are the constant functions. The term $C_k \equiv \lambda x. \bar{k}$ *λ -defines* the function $c_k: \mathbb{N} \rightarrow \mathbb{N}$ such that $c_k(n) = k$. For $C_k \bar{n} \equiv (\lambda x. \bar{k}) \bar{n} \rightarrow \bar{k}$ for any n . The identity function is *λ -defined* by $\lambda x. x$. More complex functions are of course harder to define, and often require a lot of ingenuity. So it is

perhaps surprising that every computable function is λ -definable. The converse is also true: if a function is λ -definable, it is computable.

Photo Credits

Bibliography