-NoValue-

int.1 λ-definível Arithmetical Functions

lam:int:rep:sec

How can the lambda calculus serve as a model of computation? At first, it is not even clear how to make sense of this statement. To talk about computability on the natural numbers, we need to find a suitable representation for such numbers. Here is one that works surprisingly well.

Definição int.1. For each natural number n, define the *Church numeral* $\overline{n}$ to be the lambda term $\lambda x.\, \lambda y.\, (x(x(x(\dots x(y))))),$ where there are n x's in all.

The terms $\overline{n}$ are "iterators": on input f, $\overline{n}$ returns the function mapping y to $f^n(y)$. Note that each numeral is normal. We can now say what it means for a lambda term to "compute" a function on the natural numbers.

Definição int.2. Let $f(x_0, \dots, x_{k-1})$ be an n-ary partial function from $\mathbb{N}$ to $\mathbb{N}$. We say a λ-term F *λ-definem* f iff for every sequence of natural numbers $n_0, \dots, n_{k-1}$,

$$F\, \overline{n_0}\, \overline{n_1} \dots \overline{n_{k-1}} \twoheadrightarrow \overline{f(n_0, n_1, \dots, n_{k-1})}$$

if $f(n_0, \dots, n_{k-1})$ is defined, and $F, \overline{n_0}\, \overline{n_1} \dots \overline{n_{k-1}}$ has no normal form otherwise.

lam:int:rep:thm:lambda-def

Teorema int.3. *A function f is a partial computable function if and only if it is λ-definido by a lambda term.*

This theorem is somewhat striking. As a model of computation, the lambda calculus is a rather simple calculus; the only operations are lambda abstraction and application! From these meager resources, however, it is possible to implement any computational procedure. explanation

Créditos das Imagens

Referências Bibliográficas