Lambda Calculus

Aritmética

Prof. Edson Alves

Faculdade UnB Gama

2020

Sumário

Contexto

- É natural que uma linguagem de programação seja capaz de realizar operações aritméticas com números inteiros
- Contudo, conforme dito anteriormente, o cálculo λ contém apenas dois termos primitivos: o símbolo λ e o ponto final
- Assim, como no caso dos valores lógicos, é preciso representar os números naturais por meio de expressões-λ
- Como os inteiros são infinitos, é preciso definir uma forma de deduzir todos eles a partir de algum valor inicial

Zero

Definição de zero

O número natural **zero** pode ser representado pelo termo-λ

$$0 \equiv \lambda sz.z$$

Observação: veja que, de acordo com a definição, acima $0 \equiv F$, onde F é o valor lógico falso.

Sucessor

Sucessor

O termo-λ

$$S \equiv \lambda wyx.y(wyx)$$

é denominado função sucessor, ou simplesmente, sucessor.

Observação: a função sucessor permite a definição de todos os números naturais a partir do zero: $1 \equiv S0, 2 \equiv S1, \ldots$.

Definição de 1

$$
\begin{aligned}
1 &\equiv S0 \\
&\equiv (\lambda wyx.y(wyx))(\lambda sz.z) \\
&\equiv (\lambda w.(\lambda yx.y(wyx)))(\lambda sz.z) \\
&\equiv (\lambda yx.y(wyx))[w := (\lambda sz.z)] \\
&\equiv \lambda yx.y((\lambda sz.z)yx) \\
&\equiv \lambda yx.y(x) \\
&\equiv \lambda sz.s(z)
\end{aligned}
$$

Observação: no último passo foi aplicada uma conversão-α para renomear as variáveis y e x, de modo a manter as variáveis s e z nas definições dos números naturais

Definição de 2

$$
\begin{aligned}
2 &\equiv S1 \\
&\equiv (\lambda wyx.y(wyx))(\lambda sz.s(z)) \\
&\equiv (\lambda w.(\lambda yx.y(wyx)))(\lambda sz.s(z)) \\
&\equiv (\lambda yx.y(wyx))[w := (\lambda sz.s(z))] \\
&\equiv \lambda yx.y((\lambda sz.s(z))yx) \\
&\equiv \lambda yx.y(y(x)) \\
&\equiv \lambda sz.s(s(z))
\end{aligned}
$$

Observação: a definição dos naturais pode interpretada como composições de funções. Se s é uma função, 0 significa simplesmente retornar o argumento z; 1 significa aplicar a função uma vez $s(z)$; 2 significa aplicar a função duas vezes: $s(s(z)) = s^2(s)$, e assim por diante.

Teste Condicional

Função $\mathbf{Z}$

O termo-λ

$$Z \equiv \lambda x.xF\neg F$$

o qual chamaremos **função $\mathbf{Z}$**, retorna verdadeiro (T) quando aplicada em 0, e retorna falso (F) para qualquer outro número natural.

Observações sobre a função $\mathbf{Z}$

- Para entender o comportamento da função $\mathbf{Z}$, observe que

$$Oyx \equiv (\lambda sz.z)yx \equiv x,$$

isto é, quando aplicada ao termo xy, 0 ignora a "função" y e retorna o argumento x

- O termo-λ F, quando aplicado em qualquer termo lambda z, retorna a identidade $\mathbf{I}$, pois

$$Fz \equiv (\lambda xy.y)z \equiv (\lambda x.(\lambda y.y))z \equiv (\lambda y.y)[x := z] \equiv \lambda y.y \equiv \mathbf{I}$$

- O natural N aplica N vezes o termo y ao argumento x:

$$Nyx \equiv (\lambda sz.s(s(\ldots s(z)))yx \equiv y(y(\ldots y(x)))$$

Observações sobre a função Z

- Assim,

$$\begin{aligned} Z0 &\equiv (\lambda.xF\neg F)0 \\ &\equiv 0F\neg F \equiv (0F\neg)F \\ &\equiv \neg F \equiv T \end{aligned}$$

- Para um natural N qualquer,

$$\begin{aligned} ZN &\equiv (\lambda.xF\neg F)N \\ &\equiv NF\neg F \equiv (NF\neg)F \\ &\equiv \mathbf{I}F \equiv F, \end{aligned}$$

pois uma ou mais aplicações de F ao argumento $\neg$ resulta na identidade $\mathbf{I}$

Referências

1. **ROJAS**, Raúl. *A Tutorial Introduction to the Lambda Calculus*, FU Berlin, WS-97/98.
2. **BARENDREGT**, Henk; **BARENDSEN**, Erik. *Introduction to Lambda Calculus*, March 2000.
3. Wikipédia. Combinatory logic, acesso em 07/01/2020.
4. Wikipédia. Lambda calculus, acesso em 03/01/2020.