

# Lambda Calculus

Aritmética

**Prof. Edson Alves**

Faculdade UnB Gama

# Sumário

1. Números Naturais
2. Relações entre números naturais
3. Adição e Multiplicação

## Contexto

- ▶ É esperado que uma linguagem de programação seja capaz de realizar operações aritméticas com números naturais
- ▶ Contudo, conforme dito anteriormente, o cálculo  $\lambda$  contém apenas dois termos primitivos: o símbolo  $\lambda$  e o ponto final
- ▶ Assim, como no caso dos valores lógicos, é preciso representar os números naturais por meio de expressões- $\lambda$
- ▶ 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- $\lambda$

$$0 \equiv \lambda s z . z$$

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

# Sucessor

## Sucessor

O termo- $\lambda$

$$S \equiv \lambda w y x. y(w y x)$$

é denominado função sucessor, ou simplesmente, **sucessor**, de um número natural.

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

## Definição de 1

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

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

## Definição de 2

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

**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:  $s(s(z)) = s^2(z)$ , e assim por diante.

# Teste Condicional

## Função Z

O termo- $\lambda$

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

o qual chamaremos **função 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

$$0yx \equiv (\lambda sz.z)yx \equiv x,$$

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

- ▶ O termo- $\lambda$   $F$ , 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  $x$ :

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

## Observações sobre a função $Z$

- Assim,

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

- Para um natural  $N$  qualquer,

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

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

# Pares

## Pares

No cálculo  $\lambda$ , o **par**  $(a, b)$  pode ser representado pela expressão- $\lambda$

$$(a, b) \equiv \lambda z. zab$$

O **primeiro elemento** do par pode ser extraído a partir da aplicação desta expressão ao termo  $T$ :

$$(\lambda z. zab)T \equiv Tab \equiv a$$

O **segundo elemento** é extraído por meio da aplicação da expressão ao termo  $F$ :

$$(\lambda z. zab)F \equiv Fab \equiv b$$

## Par sucessor

### Proposição

O termo- $\lambda$

$$\Phi \equiv (\lambda pz.z(S(pT))(pT))$$

transforma o par  $(n, n - 1)$  no par  $(n + 1, n)$ .

### Demonstração

De fato, o termo  $(pT)$  extrai o primeiro elemento do par, e o termo  $S(pT)$  é o sucessor deste elemento. Assim,

$$\begin{aligned} \Phi((n, n - 1)) &\equiv (\lambda pz.z(S(pT))(pT))(\lambda z.z(n)(n - 1)) \\ &\equiv (\lambda z.z(S((\lambda z.z(n)(n - 1))T))((\lambda z.z(n)(n - 1))T)) \\ &\equiv (\lambda z.z(S(Tn(n - 1)))(Tn(n - 1))) \\ &\equiv (\lambda z.z(Sn)n) \equiv (\lambda z.z(n + 1)n) \equiv (n + 1, n) \end{aligned}$$

# Antecessor

## Proposição

A expressão- $\lambda$

$$P \equiv (\lambda n.n\Phi(\lambda z.z00)F)$$

computa o **antecessor** de qualquer natural  $N$  maior que zero, e  $P0 = 0$ .

## Demonstração

Temos que

$$P0 \equiv ((\lambda n.n\Phi(\lambda z.z00))F)0 \equiv 0\Phi(\lambda z.z00)F \equiv (\lambda z.z00)F \equiv 0$$

Seja  $N$  um natural maior do que zero. Daí

$$PN \equiv ((\lambda n.n\Phi(\lambda z.z00))F)N \equiv N\Phi(\lambda z.z00)F \equiv (\lambda z.zN(N-1))F \equiv N-1,$$

pois  $N\Phi(\lambda z.z00)$  corresponde a  $N$  aplicações do termo  $\Phi$  ao par  $(0, 0)$ .

## Exemplo de antecessor

$$\begin{aligned}P3 &\equiv ((\lambda n.n\Phi(\lambda z.z00))F)3 \\&\equiv 3\Phi(\lambda z.z00)F \\&\equiv (\Phi(\Phi(\Phi(\lambda z.z00))))F \\&\equiv (\Phi(\Phi(\lambda z.z10)))F \\&\equiv (\Phi(\lambda z.z21))F \\&\equiv (\lambda z.z32)F \\&\equiv 2\end{aligned}$$

# Desigualdade

## Relação maior ou igual que

Sejam  $x$  e  $y$  dois números naturais. O termo- $\lambda$

$$G \equiv (\lambda xy. Z(xPy))$$

retorna verdadeiro ( $T$ ) se  $x$  é **maior ou igual que**  $y$ , ou falso ( $F$ ), caso contrário.

**Observação:** o termo  $G$  pode ser interpretado da seguinte maneira: se o resultado de se aplicar  $x$  vezes o antecessor  $P$  no natural  $y$  é zero, então  $x \geq y$ . Este procedimento remete à subtração  $y - x$ , exceto pelo fato de que o resultado será zero caso  $x$  seja maior do que  $y$ , e não o número negativo correspondente nos inteiros.

# Igualdade

## Relação igual a

O termo- $\lambda$

$$E \equiv (\lambda xy. \wedge (Z(xPy))(Z(yPx)))$$

retorna verdadeiro ( $T$ ) se  $x$  e  $y$  são números naturais iguais, e falso ( $F$ ), caso contrário.

**Observação:**  $x = y$  se  $x \geq y$  e  $y \geq x$ . Esta propriedade pode ser observada na definição da expressão  $E$ , na qual aparecem o termo- $\lambda$  correspondente à operação lógica  $\wedge$  e termos oriundos da definição da desigualdade maior ou igual que ( $G$ ).



# Adição

## Adição de números naturais

Seja  $S$  a expressão- $\lambda$  que computa o sucessor de um número natural. O termo- $\lambda$

$$+ \equiv (\lambda xy.xSy)$$

corresponde à **adição** de números naturais.

**Observação:** de acordo com a definição acima, a adição  $(+)$  é uma operação pré-fixada.

## Exemplo de adição

$$\begin{aligned} +23 &\equiv (\lambda xy.xSy)23 \\ &\equiv 2S3 \\ &\equiv S(S(3)) \\ &\equiv S(4) \\ &\equiv 5 \end{aligned}$$

# Multiplicação

## Multiplicação de números naturais

A expressão- $\lambda$

$$\times \equiv (\lambda x y z. x(yz))$$

corresponde à **multiplicação** de números naturais.

### Observações:

- (a) Do mesmo modo que foi observado na adição, a multiplicação ( $\times$ ) é uma operação pré-fixada
- (b) A interpretação desta expressão é a seguinte:  $x$  marca o número de vezes que será aplicada a função  $(yz)$ , a qual aplica  $y$  vezes a função  $z$

## Exemplo de multiplicação

$$\begin{aligned}\times 23 &\equiv (\lambda x y s. x(y s)) 23 \\ &\equiv \lambda s. 2(3s) \\ &\equiv \lambda s. (\lambda y z. y(y(z)))(3s) \\ &\equiv \lambda s. (\lambda z. 3s(3sz)) \\ &\equiv \lambda s. (\lambda z. 3s(s(s(s(z))))) \\ &\equiv \lambda s. (\lambda z. s(s(s(s(s(s(z))))))) \\ &\equiv \lambda s z. s(s(s(s(s(s(z)))))) \\ &\equiv 6\end{aligned}$$

## Referências

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