

# Fundamentos de Cálculo Lambda

Pedro Pinto

Instituto Superior Técnico  
Universidade Técnica de Lisboa

Lisboa, 2 de Novembro de 2006

# Índice

Introdução

O Sistema

Aplicação e Abstracção

Axiomática

Teorema do Ponto Fixo

Definibilidade Lambda

Redução

Aplicações

# Aplicação

FA

F é o algoritmo aplicado ao input A.

# Abstracção

Se  $M \equiv M[x]$  é uma expressão que contém  $x$ , então  $\lambda x.M[x]$  exprime a função  $x \mapsto M[x]$ .

# Conversão $\beta$

O principal axioma do cálculo  $\lambda$  é:

$$(\lambda x.M)N = M[x := N], \forall M, N \in \Lambda$$

# Axiomas Lógicos e Regras

Igualdade:

$$M = N;$$

$$M = N \Rightarrow N = M;$$

$$M = N, N = L \Rightarrow M = L;$$

Compatibilidade:

$$M = M' \Rightarrow MZ = M'Z$$

$$M = M' \Rightarrow ZM = ZM'$$

$$M = M' \Rightarrow \lambda x.M = \lambda x.M'$$

# Axiomática

Se  $M = N$  é demonstrável, designa-se como  $\lambda \vdash M = N$  ou abreviadamente  $M = N$ .

Exemplo,  $(\lambda y. yy)x = xx$ , então

$\lambda \vdash \lambda x. x((\lambda y. yy)x)x = \lambda x. x(x x)x$

# Teorema do Ponto Fixo

## Theorem

1.  $\forall F \exists X FX = X$

# Teorema do Ponto Fixo

## Theorem

1.  $\forall F \exists X FX = X$
2. *Existe um determinador de pontos fixos*  
 $Y \equiv \lambda f. (\lambda x. f(x x)) (\lambda x. f(x x))$   
*tal que*  
 $\exists F F(YF) = YF$

# Demonstração do Teorema do Ponto Fixo

Proof.

1. Seja  $W \equiv \lambda x.F(x x)$  e  $X \equiv WW$ . Então  
 $X \equiv WW \equiv (\lambda x.F(x x))W = F(W W) \equiv FX$ .



# Demonstração do Teorema do Ponto Fixo

## Proof.

1. Seja  $W \equiv \lambda x.F(x x)$  e  $X \equiv WW$ . Então  
 $X \equiv WW \equiv (\lambda x.F(x x))W = F(W W) \equiv FX$ .
2. Por (1), temos  
 $YF = (\lambda x.F(x x))(\lambda x.F(x x)) \equiv X$



# Representação de Booleanos

Combinadores correspondentes a valores lógicos:

- ▶  $True \equiv \lambda xy.x$
- ▶  $False \equiv \lambda xy.y$

## Representação de Expressões Condicionais

Combinadores correspondentes à construção de expressões condicionais por:  $(If\_Then\_Else) \equiv \lambda bxy.bxy$ .

## Representação de Pares Ordenados e Projecções

Combinadores correspondentes à construção de pares ordenados por:  $[\_, \_] \equiv \lambda xyz.zxy$ ,  $(\_)_0 \equiv \lambda x.xTrue$  e  $(\_)_1 \equiv \lambda x.xFalse$ .

Assim temos as seguintes abreviaturas:

$$[M, N] \equiv [\_, \_]MN$$

$$(M)_0 \equiv (\_)_0M$$

$$(M)_1 \equiv (\_)_1M$$

# Teorema de Church-Rosser

## Theorem

*Se  $M \rightarrow_{\beta} N_1$ ,  $M \rightarrow_{\beta} N_2$ , então existe um  $N_3$  tal que  $N_1 \rightarrow_{\beta} N_3$  e  $N_2 \rightarrow_{\beta} N_3$ .*

## Consequências do Teorema de Church-Rosser

1. O cálculo  $\lambda$  é consistente, i.e.  $\lambda \not\vdash true = false$ .  
Senão  $true =_{\beta} false$ , o que é impossível, visto que  $true$  e  $false$  são  $\beta$ -nf's distintos.

## Consequências do Teorema de Church-Rosser

1. O cálculo  $\lambda$  é consistente, i.e.  $\lambda \not\vdash true = false$ .  
Senão  $true =_{\beta} false$ , o que é impossível, visto que  $true$  e  $false$  são  $\beta$ -nf's distintos.
2.  $\Omega \equiv (\lambda x.xx)(\lambda x.xx)$  não tem  $\beta$ -nf.

## Consequências do Teorema de Church-Rosser

1. O cálculo  $\lambda$  é consistente, i.e.  $\lambda \not\vdash true = false$ .  
Senão  $true =_{\beta} false$ , o que é impossível, visto que  $true$  e  $false$  são  $\beta$ -nf's distintos.
2.  $\Omega \equiv (\lambda x.xx)(\lambda x.xx)$  não tem  $\beta$ -nf.
3. Para determinar a  $\beta$ -nf de um termo, as varias subexpressões podem ser reduzidas por ordens diferentes, se for encontrada uma  $\beta$ -nf, ela é única.

# Teorema da Normalização

## Theorem

*Se  $M$  tem uma forma normal, a redução em  $M$  do redex cuja abstração surja mais à esquerda, corresponde a essa forma normal.*

# Aplicações

# Sumário

- ▶ O Cálculo- $\lambda$  **originou linguagens funcionais.**
- ▶ Vantagens de linguagens baseadas no Cálculo- $\lambda$ 
  - ▶ Uma base matemática elegante.
  - ▶ Definição formal independente da arquitectura onde é implementada.
  - ▶ Correspondência directa com objectos matemáticos.
  - ▶ Programas modulares.