# Lambda Calculus

## Recursão

**Prof. Edson Alves**

Faculdade UnB Gama

# Sumário

# Recursão

- A recursão diz respeito a definição de uma função em termos de si mesma
- Ao contrário de outras notações, o cálculo $\lambda$ não permite esta definição diretamente, uma vez que os termos-$\lambda$ são anônimos
- Uma maneira de contornar isso é utilizar uma expressão-$\lambda$ que receba a si mesma como argumento
- Além disso, é preciso lidar com os dois aspectos fundamentais de uma função recursiva: o(s) caso(s) base(s) e a chamada recursiva

## Estrutura básica da recursão

$$\gamma(x) = \begin{cases} g(x), & \text{se } P(x), \\ h(x, \gamma), & \text{caso contrário} \end{cases}$$

- $P(x)$ é um predicado que retorna verdadeiro se $x$ é o valor que caracteriza um caso base
- Se $P(x)$ for verdadeiro, o valor de $\gamma$ em $x$ será dado pela função $g$
- Caso contrário, $\gamma(x)$ será dado por $h(x, \gamma)$, onde $h$ é uma função que depende de $x$ e de $\gamma$

## Representação da estrutura básica da recursão no cálculo-$\lambda$

$$\Gamma \equiv (\lambda\gamma x.(Px)(gx)(h))$$

- Observe que na definição da função recursiva $\Gamma$ é utilizado o termo-$\lambda$ $I_F$
- Se o predicado $(Px)$ retornar verdadeiro, o retorno será o primeiro parâmetro $(gx)$, que corresponde ao valor de $\Gamma$ para o caso base
- Se falso, será avaliada a função $h = h(x, \gamma)$
- Não há garantias, contudo, que $\Gamma \equiv \gamma$, pois no cálculo $\lambda$ os termos são anônimos
- É preciso, portanto, definir um termo que garanta esta equivalência

# Teorema do Ponto Fixo

## Teorema do Ponto Fixo

Para qualquer termo-$\lambda$ $G$ existe um termo $X$ tal que $GX \equiv X$.

## Demonstração

Seja $G$ um termo-$\lambda$ qualquer. Defina $W \equiv (\lambda x.G(xx))$ e $X = WW$.
Deste modo,

$$X \equiv WW \equiv (\lambda x.G(xx))W \equiv G(WW) \equiv GX$$

# Combinador $\mathbf{Y}$

## Proposição (Combinador $\mathbf{Y}$)

O **combinador** $\mathbf{Y}$

$$\mathbf{Y} \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$$

é um termo-$\lambda$ tal que, para qualquer termo $G$,

$$\mathbf{Y}G \equiv G(\mathbf{Y}G)$$

## Demonstração

Seja $G$ um termo-$\lambda$ qualquer. Daí

$$\begin{aligned}
\mathbf{Y}G &\equiv (\lambda f.(\lambda x.f(xx))(\lambda x.f(xx)))G \\
&\equiv (\lambda x.G(xx))(\lambda x.G(xx)) \equiv G((\lambda x.G(xx))(\lambda x.G(xx))) \\
&\equiv G(\lambda f.(\lambda x.f(xx))(\lambda x.f(xx)))G) \equiv G(\mathbf{Y}G)
\end{aligned}$$

# Observações sobre o combinador $\mathbf{Y}$

- Veja que, para qualquer termo-$\lambda$ $G$, $\mathbf{Y}G$ é um ponto fixo de $G$
- Esta propriedade é o que faltava para a definição completa da recursão, pois ao aplicar $(\mathbf{Y}G)$ ao parâmetro $x$ da recursão, o resultado é

$$(\mathbf{Y}G)x \equiv G(\mathbf{Y}G)x,$$

  ou seja, o termo $G$ é aplicado aos parâmetros $\mathbf{Y}G$ e $x$, o que permite invocar $G$ novamente quantas vezes forem necessárias
- Assim, para definir uma função recursiva $\mathbf{Y}\Gamma$ no cálculo-$\lambda$, basta determinar o predicado $P$ e as funções $g$ e $h$ que compõem a função $\Gamma$

# Exemplo de recursão: fatorial

$$!n = \begin{cases} 1, & \text{se } n = 0, \\ n \times !(n-1), & \text{caso contrário} \end{cases}$$

- A notação está "invertida" para ficar consistente com a notação prefixada do cálculo lambda
- Na notação de recursão do cálculo lambda, $P \equiv Z, g \equiv 1$ e $h \equiv \times x(f(Px))$, onde $\times ab$ é a multiplicação dos naturais $a$ e $b$ e $Pn$ é o antecessor do natural $n$.
- Deste modo, $! = \mathbf{Y}\Gamma$, onde

$$\Gamma \equiv \lambda fx.(Zx)1(\times x(f(Px)))$$

# Exemplo de aplicação do fatorial

$$
\begin{aligned}
!3 \equiv (\mathbf{Y}\Gamma)3 \equiv \Gamma(\mathbf{Y}\Gamma)3 & \\
&\equiv (\lambda fx.(Zx)1(\times x(f(Px))))(\mathbf{Y}\Gamma)3 \\
&\equiv (Z3)1(\times 3((\mathbf{Y}\Gamma)(P3))) \\
&\equiv F1(\times 3((\mathbf{Y}\Gamma)(P3))) \\
&\equiv \times 3((\mathbf{Y}\Gamma)2)) \equiv \times 3(\Gamma(\mathbf{Y}\Gamma)2)) \\
&\equiv \times 3((Z2)1(\times 2((\mathbf{Y}\Gamma)(P2)))) \\
&\equiv \times 3(\times 2((\mathbf{Y}\Gamma)1)) \equiv \times 3(\times 2(\Gamma(\mathbf{Y}\Gamma)1)) \\
&\equiv \times 3(\times 2((Z1)1(\times 1((\mathbf{Y}\Gamma)(P1))) \\
&\equiv \times 3(\times 2(\times 1((\mathbf{Y}\Gamma)0))) \\
&\equiv \times 3(\times 2(\times 1((Z0)1(\times 0((\mathbf{Y}\Gamma)(P0)))))) \\
&\equiv \times 3(\times 2(\times 1(1))) \equiv \times 3(\times 2(1)) \equiv \times 3(2) \equiv 6
\end{aligned}
$$

# Referências

1. **BARENDREGT**, Henk; **BARENDSEN**, 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.