

Polimorfismo: Permite que uma função trabalhe com diferentes tipos de dados. Ex.: Cálculo λ mbda.

Sintaxe

1) Variáveis

2) Definição de funções

3) Aplicação de funções

* Sintaxe original:

$\lambda x. x$
função \swarrow \downarrow definição
variável

* Em Haskell:

λ por \backslash
 \cdot por \rightarrow

* Abstracção:

$\backslash x \rightarrow e$

* Aplicação:

$e_1(e_2(e_3))$

Acceptor Sintático

original

$((e_1 e_2) e_3) e_4$

$\backslash x \rightarrow (\backslash y \rightarrow (\backslash z \rightarrow e))$

$\backslash x \rightarrow \backslash y \rightarrow \backslash z \rightarrow e$

accept sintático

$e_1 e_2 e_3 e_4$

$\backslash x \rightarrow \backslash y \rightarrow \backslash z \rightarrow$

$\backslash x y z \rightarrow e$

Escopo

↳ Contexto em que as variáveis estão definidas

↳ é composto por abstrações (funções) e aplicações

↳ variável ligada: é a variável que está no corpo da função e também faz parte do parâmetro

↳ variável livre: fazem parte do corpo da função mas não entram como um parâmetro.

Substituições em Abstracção λ mbda

1) Abstracção λ mbda: função que recebe um argumento x

λx . expressão

2) Substituição

↳ Ao aplicar um valor a abstracção, substitui todas as ocorrências da variável por esse número

↳ A abstracção dessa parece após a substituição

Redução β

↳ É a aplicação de uma função a seus argumentos

Ex.: $(\lambda x. M) N$

↳ substituir x (em M) por N

↳ substitui as ocorrências da variável da abstracção pela expressão passada como argumento

Ordem de Avaliação

- ↳ O redex mais à esquerda mais externo é reduzido primeiro
 - ↳ expressão redutível

* Não entendi muito bem numerals de Church

Curificação

↳ É o processo de transformar uma função λ com mais argumentos em uma sequência de funções λ com um único argumento

↳ Ex.:

↳ $x = x + y$ (normal)

↳ $x = \lambda y \rightarrow x + y$ (curificada)

Divergência

↳ É quando a redução não atinge um resultado final

Cidadão de Primeira Classe

↳ Funções que são tratadas como valores normais, ou seja elas podem ser usadas da mesma forma que outros valores

Notação Lambda

↳ Ex. em Sintaxe básica: $\lambda x. x + 1$

↳ Ex. em Haskell: $\backslash x \rightarrow x + 1$

↳ Aplicação de argumentos: $(\backslash x \rightarrow x + 1) 5$