



CÁLCULO - λ

Uma breve apresentação
sobre o Cálculo - λ Tipado

Por: Carolina Lima
Mateus Gontijo

SUMÁRIO

01

O QUE É

Breve definição

CARACTERÍSTICA

Aspecto relevante

02

03

O CÁLCULO – λ TIPADO

Conceito



01

O QUE É?

- Sistema formal que estuda funções recursivas computáveis referentes a teoria da computabilidade, e fenômenos relacionados.



02

CARACTERÍSTICA RELEVANTE

- Entidades que podem ser utilizadas como argumentos e retornadas como valores de outras funções

FUNÇÃO SUCCESSOR

value succ = fun (x: Int) $x+1$

- Deve ser aplicada apenas a expressões lambda que representam inteiros
- A notação infixa $x+1$ é uma abreviação para a notação funcional $+(x)(1)$
- Os símbolos 1 e $+$ devem ser vistos como abreviações para uma expressão pura de cálculo lambda para o número 1 e adição.

FUNÇÃO TWICE

```
value twice = fun(f: Int  $\rightarrow$  Int) fun (y: Int) f(f(y))
```

- Essa notação omite a especificação do tipo de resultado
- Podemos denotar o tipo do resultado com uma **return keyword**:

```
value succ = fun(x: Int) (returns Int) x + 1
```

- O tipo do resultado pode ser determinado a partir da forma da função body **x + 1**
- Devemos omitir as especificações do tipo de resultado por razões de brevidade

DECLARAÇÕES DE TIPO

- São introduzidas pelo tipo de palavra-chave
- Os nomes dos tipos começam com letras **maiúsculas**
- Os nomes de valor e função começam com letras **minúsculas**.

type IntPair = Int x Int

type IntFun = Int \rightarrow Int

DECLARAÇÕES DE TIPO

- O tipo de uma variável pode ser determinado a partir da forma do valor atribuído:

```
value intPair = (3,4)
```

- Indicando o tipo de uma variável pela notação `value var:T = value`

```
value intPair: IntPair = (3,4)
```

```
value succ: Int  $\rightarrow$  f Int = fun(x: Int) x + 1
```


VARIÁVEIS LOCAIS

- Podem ser declaradas pelo construct let-in

```
let a = 3 in a + 1
```

```
yields 4
```

- Introduz uma nova variável inicializada em um escopo local
- O valor do construct é o valor dessa expressão
- Se quisermos especificar tipos, podemos escrever:

```
let a : Int = 3 in a + 1
```

O LET-IN CONSTRUCT

- Pode ser definido em termos de expressões básicas de fun:

$\text{let } a:T = \text{MinN} \equiv (\text{fun}(a:T)N)(M)$

REFERÊNCIA

- Cardelli, Luca; Wegner, Peter. On understanding types, data abstraction, and polymorphism. Computing Surveys. Vol. 17. N° 4. December 1985. Pgs. 471-489.