

# anotações em aula

Questão 2 - indução finita

Base:  $n=1$   $z=1$   $n=2$   $Z = 1!+2! = 1+ 2 \times 1$

Hipótese  $\rightarrow z$  é ímpar (Verdade)

Tese

---

fatorial 0 = 1 // Base

fatorial n = fatorial(n-1)\*n // Indução

---

Tipos em Haskell:

- Tipos básicos:
  - Int - números inteiros
  - Integer - inteiros sem limite
  - Float/Double - números decimais
  - Char - caracteres
  - Bool - valores lógicos True/False
  - String - lista de caracteres

Operadores:

- Aritméticos: +, -, \*, /, div, mod
- Relacionais: ==, /=, <, >, <=, >=
- Lógicos: &&, ||, not

**Listas:**

- Tem número infixo de elementos, ser de um *único tipo*
- Definição: [1,2,3]

"X :: [a]" "y :: [[a]]" "[ ] :: [[a]]"

[1,2,3,4] :: [Int]

`[ ] :: [Int]`

`[[1,2],[4,7],[ ]] : [[Int]] → [ ] :: [[Int]]`

- Operadores:
  - `(:)` - cons, adiciona elemento
  - `(++)` - concatena listas
  - `(!!)` - acessa elemento por índice

Tuplas:

- Definição: `(1,"texto",True)`
- Tamanho fixo e tipos diferentes
- Funções: `fst` (primeiro), `snd`(ultimo) (para pares)

---

Uma lista pode ser concatenada

`[1,2,3] ++ [5,7,8] = [1,2,3,5,7,8]`

`3:[2,5] = [3,2,5]`

`'a':[3] = erro` pois a lista é de tipo `Int`

`[2,3,4]:[ ] = [[2,3,4]]` // lista de lista de inteiros

`2:[ ] = erro` por ser lista de lista de inteiros, o elemento tem que ser uma lista

`[2,3]:[ ] = [[2,3],[ ]]`

`[ ] ++ [2] = concatena duas listas`

---

*: Insere elemento do tipo a em [a]*

*++ concatena duas listas [a]*

`[3,9,4,2]` Insere na última posição o elemento 10

`[3,9,4,2] ++ [10]`

`[ ]:[ ] = [ ]` // considera que `[ ]` é elemento da lista `[ ]`

$[] ++ [] = []$

$[[3,4]]:[ ] = [[[3,4]]]$  // lista de lista de lista de inteiro

$[ ]:[[[3,4]]] = [[ ],[[3,4]]]$

$[[ ],[1,2,3],[5]] :: [Int]$

$[2]:[[ ],[1,2,3],[6]] = [[2],[ ],[1,2,3],[6]]$

$(2:[ ]):[[1,2,3],[5]] = [[2],[1,2,3],[5]]$  // os parenteses não fazem com que seja uma tupla, apenas estão isolando para ser executada primeiramente

---

— soma elementos de uma  $[Int]$  = Ela percorre a lista recursivamente, somando cada elemento até que a lista esteja vazia, retornando 0.

$f :: [Int] \rightarrow Int$

$f[ ] = 0$

$f(a:b) = a + f(b)$  ; a é primeiro elemento e b a calda da lista

---

Associativa à direita

Quebra por parênteses:

- Precedencia - entre os operadores
- Associatividade - para o mesmo operador

- Avaliação Preguiçosa

$a:[ ]:[ ] = a:[[ ]]$

$3:4:2:7:[ ] = [3,4,2,7]$

$2:4:7:[3]:[ ] = 2:4:7:[3]$  = continuação dá erro!

→ Associativa à direita

$(a,x) \rightarrow$  tupa (dupla) /\* **atenção!**

(a:x) → construtores de lista

---

maiorV :: MV → [(Dia,Venda)] → Int

maiorV mV [ ] = mV

maiorV mV (a:x)