

Introdução à Programação Funcional usando



Aula 5 – Listas.

Listas

Listas são coleções de objetos que podem ser acessados sequencialmente. Em Haskell listas são representadas por uma sequência de objetos de mesmo tipo, entre colchetes e separados por vírgulas. Por exemplo:

`[1, 1, 2, 3, 5, 8, 13] :: (Num a) => [a]`

`[True, False, False, True] :: [Bool]`

`['H', 'a', 's', 'k', 'e', 'l', 'l'] :: [Char] ou String`

Listas de caracteres são um caso especial e podem também ser representadas pela sequência de caracteres entre aspas. Portanto, em Haskell “ABC” e `['A', 'B', 'C']` representam listas de caracteres idênticas.

Listas

O tipo lista é definido indutivamente por meio de dois construtores de dados:

[] Constrói uma lista vazia.

$x : xs$ Constrói uma lista com cabeça x e cauda xs .

Por exemplo: $[1, 2, 3, 4] == 1: (2: (3: (4: [])))$

Construtores de Dados Lista

Os **construtores de dados** podem receber zero ou mais dados como parâmetros para construir um novo dado. Os construtores de dados podem ser usados para construir um novo dado ou para desmontar, no casamento de padrão, um dado passado como parâmetro para uma função:

$$[] :: [a]$$
$$(:) :: a \rightarrow [a] \rightarrow [a]$$

Construtores de Dados Lista

Os **construtores de dados** podem receber zero ou mais dados como parâmetros e constroem um novo dado. Os construtores podem ser usados para construir um novo dado ou para desmontar, no casamento de padrão, um dado passado como parâmetro para um função:

$$[] :: [a]$$
$$(:) :: a \rightarrow [a] \rightarrow [a]$$
$$\text{tamanho} :: [a] \rightarrow \text{Int}$$
$$\text{tamanho } [] = 0$$
$$\text{tamanho } (x:xs) = 1 + \text{tamanho } xs$$

Exemplos com Listas

`tamanho [] = 0`

`tamanho (x : xs) = 1 + tamanho xs`

`> tamanho [10, 20, 30, 40] -- 10:20:30:40:[]`

Exemplos com Listas

$\text{tamanho } [] = 0$

$\text{tamanho } (x : xs) = 1 + \text{tamanho } xs$

> tamanho [10, 20, 30, 40]

1 + tamanho [20, 30, 40]

Exemplos com Listas

$\text{tamanho } [] = 0$

$\text{tamanho } (x : xs) = 1 + \text{tamanho } xs$

> tamanho [10, 20, 30, 40]

1 + tamanho [20, 30, 40]

1 + 1 + tamanho [30, 40]

Exemplos com Listas

$\text{tamanho } [] = 0$

$\text{tamanho } (x : xs) = 1 + \text{tamanho } xs$

> tamanho [10, 20, 30, 40]

1 + tamanho [20, 30, 40]

1 + 1 + tamanho [30, 40]

1 + 1 + 1 + tamanho [40] -- 40:[]

Exemplos com Listas

`tamanho [] = 0`

`tamanho (x : xs) = 1 + tamanho xs`

`> tamanho [10, 20, 30, 40]`

`1 + tamanho [20, 30, 40]`

`1 + 1 + tamanho [30, 40]`

`1 + 1 + 1 + tamanho [40] -- 40:[]`

`1 + 1 + 1 + 1 + tamanho []`

Exemplos com Listas

$\text{tamanho } [] = 0$

$\text{tamanho } (x : xs) = 1 + \text{tamanho } xs$

$\text{length} :: [a] \rightarrow \text{Int}$

> tamanho [10, 20, 30, 40]

1 + tamanho [20, 30, 40]

1 + 1 + tamanho [30, 40]

1 + 1 + 1 + tamanho [40]

1 + 1 + 1 + 1 + tamanho []

1 + (1 + (1 + (1 + 0)))

Exemplos com Listas

somatorio :: (Num a) => [a] → a sum

somatorio [] = 0

somatorio (x:xs) = x + somatorio xs

> somatorio [10, 20, 30]

10 + somatorio [20, 30]

10 + 20 + somatorio [30]

10 + 20 + 30 + somatorio []

10 + (20 + (30 + 0))

Exemplos com Listas

$\text{nprimeiros} :: \text{Int} \rightarrow [a] \rightarrow [a]$ take

$\text{nprimeiros } 0 \text{ } _ = []$

$\text{nprimeiros } _ [] = []$

$\text{nprimeiros } n (x:xs) = x : \text{nprimeiros } (n - 1) xs$

> $\text{nprimeiros } 2 [10, 20, 30, 40]$

10 : $\text{nprimeiros } 1 [20, 30, 40]$

10 : 20 : $\text{nprimeiros } 0 [30, 40]$

10 : 20 : []