

# Aula Prática 6

Paradigmas da Programação I / Programação Funcional

ESI/MCC 1º ano (2005/2006)

## Iteração Sobre Listas: a função de ordem superior `foldr`

O operador `foldr`, tal como `map` e `filter`, permite escrever de forma expedita, sem recursividade explícita, um grande conjunto de funções (incluindo as próprias funções `map` e `filter`).

Uma forma simples de visualizar o funcionamento desta função é a seguinte: os constructores `(:)` e `[]` são substituídos pelos dois parâmetros de `foldr`.

Por exemplo, recordando que `[1,2,3] == 1:(2:(3:[]))`, tem-se que

```
foldr (+) 0 [1,2,3] => 1+(2+(3+0))
```

```
foldr (*) 1 [1,2,3] => 1*(2*(3*1))
```

Isto permite definir:

```
somaLista l = foldr (+) 0 l
```

```
multLista l = foldr (*) 1 l
```

**Tarefa 1** Investigue cada uma das seguintes funções do Haskell:

1. `concat`
2. `and`
3. `or`

Escreva definições destas funções usando `foldr`.

**Tarefa 2** Considere a seguinte definição de uma função que separa uma lista em duas partes de comprimento idêntico:

```
separa [] = ([], [])
```

```
separa (h:t) = (h:r,l)
```

```
  where (l,r) = separa t
```

Escreva uma nova definição desta função recorrendo a `foldr`.

**Tarefa 3** Considere as duas seguintes funções:

```
merge :: (Ord a) => [a] -> [a] -> [a]
insert :: (Ord a) => a -> [a] -> [a]
```

A primeira efectua a  *fusão*  de duas listas ordenadas de forma crescente; a segunda insere um elemento numa lista ordenada de forma crescente:

```
merge [1,4] [2,3] => [1,2,3,4]
insert "bb" ["aa","cc"] => ["aa","bb","cc"]
```

Uma definição possível de `insert` é

```
insert x [] = [x]
insert x (h:t)
  | (x<=h) = x:h:t
  | otherwise = h:(insert x t)
```

1. Escreva a função `merge` utilizando `foldr` e `insert`.
2. Relembre o algoritmo de ordenação *insertion sort*, implementado em Haskell por uma função:

```
isort :: (Ord a) => [a] -> [a]
```

Reescreva esta função utilizando o operador `foldr`.

3. (\*\*) Será possível definir-se *insert* usando também *foldr*?

**Tarefa 4** Revisite a ficha da aula prática 4. Identifique as funções aí definidas que podem ser alternativamente escritas com `foldr` e escreva-as dessa forma.