

Aula Teórico-prática e Prática 11

Programação Funcional

LCC 1º ano (2009/2010)

Relembre a definição de árvores binárias:

```
data BTree a = Vazia | Nodo a (BTree a) (BTree a)
```

Nestas árvores a informação está nos nodos (as *extermidades* da árvore têm apenas uma marca – **Vazia**).

1. É também habitual definirem-se árvores em que a informação está apenas nas extremidades:

```
data LTree a = Tip a | Fork (LTree a) (LTree a)
```

Defina sobre este tipo as seguintes funções

- (a) `ltSum :: (Num a) => (LTree a) -> a` que soma as folhas de uma árvore.
- (b) `listaLT :: (LTree a) -> [a]` que lista as folhas de uma árvore (da esquerda para a direita)
- (c) `LTheight :: (LTree a) -> Int` que calcula a altura de uma árvore.

2. Estes dois conceitos podem ser agrupados num só, definindo o seguinte tipo:

```
data Tree a b = Leaf b | Node a (Tree a b) (Tree a b)
```

- (a) Defina a função `splitTree :: (Tree a b) -> (BTree a, LTree b)` que separa uma árvore com informação nos nodos e nas folhas em duas.
- (b) Defina ainda a função inversa `joinTrees :: (BTree a) -> (LTree b) -> Maybe Tree a b` que sempre que as árvores sejam *compatíveis* as junta numa só.

3. Nas árvores mencionadas acima, cada nodo tem exactamente dois *descendentes*. Daí dizermos que se tratam de **árvores regulares de grau 2**, ou de uma forma mais abreviada, árvores binárias. Este conceito pode ser generalizado para um qualquer outro grau, e teremos então árvores regulares de outro grau. Mas uma outra generalização consiste em não fixar o número de descendentes de cada nodo. Essas árvores dizem-se **irregulares**. Uma possível definição será:

```
data RTree a = R a [RTree a]
```

- (a) Defina uma função `rtHeight :: (RTree a) -> Int` que calcula a altura de uma árvore.
- (b) Defina uma função `rtSum :: (Num a) => (RTree a) -> a` que soma os elementos de uma árvore.
- (c) Defina uma função `bt2RT :: BTree a -> RTree a` que converte uma árvore binária.
- (d) Defina uma função `listaRT :: (RTree a) -> [a]` que lista os elementos de uma árvore.
- (e) Defina uma variação da função anterior que lista os elementos da árvore por níveis.