

Teste de Programação Funcional

DCC/FCUP

3 de Junho de 2014

Nome: _____

Nº mecanográfico: _____

- Este teste contém 4 questões e 2 páginas.
- Responda às questões 1 e 2 no espaço marcado no enunciado.
- Pode usar funções auxiliares e/ou do prelúdio-padrão de Haskell

1. (30%)

- (a) Escreva uma definição da função `insert` que insere um valor numa lista cujos valores estão em ordem crescente; se o valor já ocorrer na lista, então deve ser repetido; a lista resultante deve manter a ordenação. Exemplos:

```
insert 3 [1,2,4] = [1,2,3,4]
insert 'd' "abcde" = "abcdde"
```

- (b) Indique o tipo mais geral da função `insert`.

2. (30%) Complete as seguintes definições apresentadas nas aulas de funções recursivas para calcular o *tamanho* (número de nós) e a *altura* de uma árvore binária.

```
data Arv a = Vazia | No a (Arv a) (Arv a)

tamanho :: Arv a -> Int
tamanho Vazia          = _____
tamanho (No x esq dir) = _____

altura :: Arv a -> Int
altura Vazia           = _____
altura (No x esq dir) = _____
```

Responda às questões seguintes numa folha de exame.

3. (20%) Usando indução sobre xs , mostre que

$$\text{length } (xs ++ ys) = \text{length } xs + \text{length } ys$$

para quaisquer listas finitas xs, ys .

Sugestão: comece por escrever as definições recursivas das funções `length` e `++`.

4. (20%) Considere a definição seguinte da função que lista por ordem os valores de uma árvore binária.

```
listar :: Arv a -> [a]
listar Vazia = []                                     (listar.1)
listar (No x esq dir) = listar esq ++ [x] ++ listar dir (listar.2)
```

Usando indução sobre árvores, mostre que

$$\text{length } (\text{listar } t) = \text{tamanho } t$$

para qualquer árvore t .

Use a definição recursiva da função `tamanho` da resposta ao exercício 2 (**copie-a para a sua folha de exame**). Pode ainda usar o resultado do exercício 3 como um lema auxiliar na sua prova.