Teste de Programação Funcional

DCC/FCUP

22 de Abril de 2014

Versão A

Nome: _____

Nº mecanográfico: _____

- **Este teste contém 3 questões e 2 páginas.**
- **Responda à questão 1 no espaço marcado no enunciado.**
- **Pode usar funções auxiliares e/ou do prelúdio-padrão de Haskell**

1. (30%) Responda a cada uma das seguintes questões, indicando **apenas** o resultado de cada expressão.

(a) `[1,3]++([3]++[4,5])` = _____

(b) `head (reverse [5,6,7]))` = _____

(c) `tail (['a','b']:[]:['c']:[])` = _____

(d) `length [x | x<-[0..10], x*x<=9]` = _____

(e) `map (\x->3*x+1) [0,1,2]` = _____

(f) `drop 2 "ABCDEF"` = _____

(g) Indique um tipo admissível para `[(2,"abc"),(3,"def")]`.

_____

(h) Indique o tipo mais geral da função `(++)`.

_____

(i) Indique o tipo mais geral da função `take`.

_____

(j) Indique o tipo mais geral da função `filter`.

_____

Responda às questões 2 e 3 numa folha de exame.

2. (35%) Escreva uma definição recursiva da função

```
intersperse :: a -> [a] -> [a]
```

que intercala um elemento entre valores consecutivos numa lista; se a lista ter menos de dois valores deve ficar inalterada. Exemplos:

```
> intersperse 0 [1..4]
[1,0,2,0,3,0,4]
> intersperse ',' "abcd"
"a,b,c,d"
[]
> intersperse ',' "a"
"a"
```

3. (35%) Considere uma função `group :: String -> [String]` que parte uma lista de carateres em sub-sequências de carateres iguais de comprimento máximo e cuja concatenação dá a lista original. Exemplos:

```
> group "Abba"
["A", "bb", "a"]
> group "Mississippi"
["M", "i", "ss", "i", "ss", "i", "pp", "i"]
```

(a) Escreva uma definição recursiva da função `group`. Sugestão: pode utilizar as funções *takeWhile* e *dropWhile* do prelúdio-padrão.

(b) A função `group` pode ser aplicada a listas de outros valores. Qual será o tipo mais geral desta função?