

# Cálculo de Programas

2.º ano das Licenciaturas em Engenharia Informática e Ciências da Computação  
UNIVERSIDADE DO MINHO

2011/12 - Ficha nr.º 1

1. Codifique em Haskell as funções  $\text{length} :: [a] \rightarrow \text{Int}$  e  $\text{reverse} :: [a] \rightarrow [a]$  que conhece da disciplina de Programação Funcional (PF) e que, respectivamente, calculam o comprimento da lista de entrada e a invertem.

2. Recorde o tipo que se usa em Haskell para representar valores opcionais:

```
data Maybe a = Nothing | Just a
```

Defina a função  $\text{catMaybes} :: [\text{Maybe } a] \rightarrow [a]$  que extrai o conteúdo útil da lista de entrada.

3. Considere a definição da função  $\text{split} :: a \rightarrow [a] \rightarrow ([a], [a])$  que é usada pelo algoritmo *quicksort* para partir uma lista em duas:

```
split a l = ([x | x ← l, x ≤ a], [x | x ← l, x > a])
```

Apresente uma definição alternativa (recursiva) de  $\text{split}$  que não use listas por compreensão e que percorra a lista uma única vez.

4. Apresente definições em Haskell das seguintes funções que estudou em PF:

```
uncurry :: (a → b → c) → (a, b) → c (que emparelha os argumentos de uma função)
```

```
curry :: ((a, b) → c) → a → b → c (que faz o efeito inverso da anterior)
```

```
flip :: (a → b → c) → b → a → c (que troca a ordem dos argumentos de uma função)
```

5. Atente na definição seguinte de um dos combinadores emblemáticos da linguagem Haskell, que já conhece de PF:

```
foldr :: (a → b → b) → b → [a] → b
```

```
foldr g z [] = z
```

```
foldr g z (x : xs) = x 'g' (foldr g z xs)
```

(a) Defina  $\text{length} :: [a] \rightarrow \text{Int}$  usando  $\text{foldr}$ .

(b) O que faz a função  $f = \text{foldr } (:) []$ ? **Sugestão:** comece por copiar a definição dada e faça literalmente as substituições  $g := (:)$  e  $z := []$ . De seguida substitua  $\text{foldr } (:) []$  por  $f$ . Obtém assim uma definição explícita de  $f$ , sem recorrer ao combinador dado, que é mais fácil de inspeccionar.

6. Partindo da definição

```
for b i 0 = i
```

```
for b i (n + 1) = b (for b i n)
```

inspecione o significado das expressões  $\text{for } (a+) 0$  e  $\text{for } (a*) 1$ , **Sugestão:** proceda pelo método de substituição da alínea 5b.

7. Considere as funções

$$f :: [\text{Int}] \rightarrow [\text{Int}]$$
$$f\ s = [a + 1 \mid a \leftarrow s, a > 0]$$

e

$$\text{concat} :: [[a]] \rightarrow [a]$$
$$\text{concat} = \text{foldr } (+) []$$

- (a) Diga por palavras suas o que faz a função  $f$ .
- (b) Escreva-a sob a forma de um  $\text{foldr}$ .
- (c) Escreva  $\text{concat}$  sem recorrer ao combinador  $\text{foldr}$  (Sugestão: faça como na questão 5).

8. Considere a função  $m$  seguinte:

$$m :: (a \rightarrow b) \rightarrow [a] \rightarrow [b]$$
$$m\ f\ [] = []$$
$$m\ f\ (h : t) = (f\ h) : m\ f\ t$$

- (a) Reescreva-a usando o combinador  $\text{foldr}$ .
- (b) Reescreva-a sem usar o combinador  $\text{foldr}$ .
- (c) Qual o tipo da expressão  $m\ (\lambda x \rightarrow [x])$ ? E o que faz essa expressão?
- (d) Abreviando a função  $\lambda x \rightarrow [x]$  pela designação  $\text{singl}$ , averigue qual o resultado das expressões

`let s = m singl "Calculo de Programas" in concat s`

e

`concat (singl "Calculo de Programas")`

correndo-as mentalmente. Tente generalizar o que apurou nesse exercício mental.