

Cálculo de Programas

2.º ano das Licenciaturas em
Engenharia Informática e Ciências da Computação da
Universidade do Minho

2009/10 - Ficha nr.º 1

1. Atente na definição seguinte de um dos combinadores emblemáticos da linguagem Haskell, que já conhece de Programação Funcional:

$$\begin{aligned} \text{foldr} &:: (a \rightarrow b \rightarrow b) \rightarrow b \rightarrow [a] \rightarrow b \\ \text{foldr } g \ z \ [] &= z \\ \text{foldr } g \ z \ (x : xs) &= x \ 'g'\ (\text{foldr } g \ z \ xs) \end{aligned}$$

O que faz a função $f = \text{foldr } (:) \ []$? Tem duas formas de responder a esta pergunta: uma é inventar casos de teste e tentar generalizá-los; a outra, mais sistemática, é a seguinte: comece por copiar a definição dada para a sua folha 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.

2. Defina as seguintes funções usando foldr :

$$\begin{aligned} \text{(a) } \text{length} &:: [a] \rightarrow \text{Int} \\ \text{length } [] &= 0 \\ \text{length } (h : t) &= 1 + \text{length } t \\ \\ \text{(b) } \text{unzip} &:: [(a, b)] \rightarrow ([a], [b]) \\ \text{unzip } [] &= ([], []) \\ \text{unzip } ((x, y) : t) &= \mathbf{let} \ (xs, ys) = \text{unzip } t \\ &\quad \mathbf{in} \ (x : xs, y : ys) \end{aligned}$$

3. Considere as funções

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

e

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

- (a) Diga por palavras suas o que faz a função f .
(b) Escreva-a sob a forma de um foldr .
(c) Escreva concat sem recorrer ao combinador foldr (Sugestão: faça como na questão 1).

4. Considere a função m seguinte:

$$\begin{aligned}m &:: (a \rightarrow b) \rightarrow [a] \rightarrow [b] \\m f [] &= [] \\m f (h : t) &= (f h) : m f t\end{aligned}$$

- (a) Reescreva-a usando o combinador *foldr*.
- (b) Reescreva-a sem usar o combinador *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 *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.

5. Partindo da definição

$$\begin{aligned}\text{for } b \text{ } i \text{ } 0 &= i \\ \text{for } b \text{ } i \text{ } (n + 1) &= b (\text{for } b \text{ } i \text{ } n)\end{aligned}$$

inspecione o significado das expressões $\text{for } (a+) \text{ } 0$ e $\text{for } (a*) \text{ } 1$, Sugestão: para o primeiro caso, defina $f \text{ } a = \text{for } (a+) \text{ } 0$ e derive as cláusulas

$$\begin{aligned}f \text{ } a \text{ } 0 &= 0 \\ f \text{ } a \text{ } (n + 1) &= a + (f \text{ } a \text{ } n)\end{aligned}$$

substituindo b por $(a+)$ e i por 0 na definição dada, e depois $\text{for } (a+) \text{ } 0$ por $f \text{ } a$. Repita o exercício para o caso $\text{for } (a*) \text{ } 1$.