

Cálculo de Programas

Licenciatura em Engenharia Informática

Ficha 1

1. Defina funções Haskell que testemunhem os seguintes isomorfismos:

- (a) $Maybe\ a \cong Either\ a\ ()$
- (b) $Either\ ()\ () \cong Bool$
- (c) $(a, Either\ b\ c) \cong Either\ (a, b)\ (a, c)$
- (d) $Bool \rightarrow a \cong (a, a)$

2. Defina as seguintes funções usando um *foldr*:

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

3. Defina a seguinte função que concatena duas listas usando um *foldr*:

$$\begin{aligned} cat &:: [a] \rightarrow [a] \rightarrow [a] \\ cat\ []\ l &= l \\ cat\ (h : t)\ l &= h : cat\ t\ l \end{aligned}$$

- (a) Comece por definir um *foldr* sobre o primeiro argumento usando o segundo argumento como uma constante.

$$cat\ m\ l = foldr\ (\dots)\ (\dots)\ m$$

- (b) Tente eliminar a referência explícita ao segundo parâmetro, definindo um *foldr* que devolve funções.

$$cat = foldr\ (\dots)\ (\dots)$$

4. Considere o seguinte tipo de dados para representar números naturais.

data *Nat* = *Zero* | *Succ Nat*

- (a) Defina uma função $soma :: Nat \rightarrow Nat \rightarrow Nat$ que soma dois números naturais.
- (b) Defina funções que testemunhem o isomorfismo $[()] \cong Nat$. Use um *foldr* para definir a função de tipo $[()] \rightarrow Nat$.

5. Defina as seguintes funções usando um *foldr*:

(a) Dividir uma lista usando um predicado.

$$\begin{aligned} \textit{partition} &:: (a \rightarrow \textit{Bool}) \rightarrow [a] \rightarrow ([a], [a]) \\ \textit{partition } p \textit{ xs} &= (\textit{filter } p \textit{ xs}, \textit{filter } (\neg \circ p) \textit{ xs}) \end{aligned}$$

(b) Eliminar os elementos repetidos de uma lista.

$$\textit{nub} :: \textit{Eq } a \Rightarrow [a] \rightarrow [a]$$

(c) Calcular os segmentos iniciais de uma lista.

$$\textit{inits} :: [a] \rightarrow [[a]]$$