Cálculo de Programas

Licenciatura em Engenharia Informática

Ficha 2

1. Usando os combinadores que aprendeu até momento defina versões *point-free* das seguintes funções:

   (a) $\mathsf{unzip} :: [(a, b)] \to ([a], [b])$

   (b) $\mathsf{fact} :: \mathsf{Int} \to \mathsf{Int}$

   (c) $\mathsf{tails} :: [a] \to [[a]]$ (segmentos finais de uma lista: $\mathsf{tails}\ [1..3] = [[1,2,3],[2,3],[3],[]]$).

   (d) $\mathsf{plus} :: (\mathsf{Int}, \mathsf{Int}) \to \mathsf{Int}$ (só para inteiros positivos).

   (e) $\mathsf{replicate} :: (\mathsf{Int}, a) \to [a]$ ($\mathsf{replicate}\ (n, x)$ cria n cópias de x).

   Para além das funções standard do *Prelude* do Haskell, poderá usar as seguintes:

   $$\begin{array}{l}\mathsf{downto} :: \mathsf{Int} \to [\mathsf{Int}]\\ \mathsf{downto}\ n = [n, n-1 \ldots 1]\\ \mathsf{cat} :: ([a], [a]) \to [a]\\ \mathsf{cat} = \mathsf{uncurry}\ (+\!\!+)\\ \mathsf{inits} :: [a] \to [[a]]\\ \mathsf{inits}\ [\,] \quad\ = [[\,]]\\ \mathsf{inits}\ (h : t) = [[\,]] : \mathsf{map}\ (h:)\ (\mathsf{inits}\ t)\end{array}$$

2. Tal como o produto é associativo, também a nível dos tipos é válido o isomorfismo $a \times (b \times c) \cong (a \times b) \times c$.

   (a) Defina versões *point-free* das funções $\mathsf{assocl} :: a \times (b \times c) \to (a \times b) \times c$ e $\mathsf{assocr} :: (a \times b) \times c \to a \times (b \times c)$. Desenhe os respectivos diagramas.

   (b) Demonstre que $\mathsf{assocr} \circ \mathsf{assocl} = \mathsf{id}$.

3. Demonstre as seguintes propriedades sobre produtos e desenhe os respectivos diagramas:

   (a) $\mathsf{id} \times \mathsf{id} = \mathsf{id}$

   (b) $(f \times g) \circ (h \vartriangle j) = f \circ h \vartriangle g \circ j$

   (c) $(f \times g) \circ (h \times j) = f \circ h \times g \circ j$

   (d) $\mathsf{swap} \circ (f \times g) = (g \times f) \circ \mathsf{swap}$

   (e) $\mathsf{assocr} \circ ((f \times g) \times h) = (f \times (g \times h)) \circ \mathsf{assocr}$