Cálculo de Programas

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

2011/12 - Ficha nr.° 8

1. Seja dada a função seguinte, em Haskell:

$$\mathsf{sumprod}\ a\ [\,] = 0$$
$$\mathsf{sumprod}\ a\ (h : t) = a * h + \mathsf{sumprod}\ a\ t$$

   (a) Mostre que

$$\mathsf{sumprod}\ a \quad = \quad (\![[\mathsf{zero}\ , \mathsf{add} \cdot ((a*) \times id)]]\!) \tag{1}$$

   onde $\mathsf{zero} = \underline{0}$, $\mathsf{add} = \widehat{(+)}$ e o catamorfismo é de listas, isto é, tem padrão de recursividade $\mathsf{F}\ f = id + id \times f$.

   (b) Mostre, como exemplo de aplicação da propriedade de fusão-cata para listas, que

$$\mathsf{sumprod}\ a \quad = \quad (a*) \cdot \mathsf{sum} \tag{2}$$

   onde $\mathsf{sum} = (\![[\mathsf{zero}\ , \mathsf{add}]]\!)$. **NB:** não ignore propriedades elementares da aritmética que lhe possam ser úteis.

2. A função

$$\mathsf{map}\ f\ [\,] = [\,]$$
$$\mathsf{map}\ f\ (h : t) = (f\ h) : \mathsf{map}\ f\ t$$

é o catamorfismo de listas $\mathsf{map}\ f = (\!|\ \mathbf{in} \cdot (id + f \times id) |\!)$, como facilmente apura. Mostre, usando as leis de reflexão e fusão-cata (entre outras), que as seguintes propriedades se verificam:

$$\mathsf{map}\ id \quad = \quad id \tag{3}$$
$$(\mathsf{map}\ f) \cdot (\mathsf{map}\ g) \quad = \quad \mathsf{map}\ (f \cdot g) \tag{4}$$

3. Mostre que a função $f = look\ k$ onde

$$look :: \mathsf{Eq}\ a \Rightarrow a \rightarrow [(a, b)] \rightarrow \mathsf{Maybe}\ b$$
$$look\ k\ [\,] = \mathsf{Nothing}$$
$$look\ k\ ((a, b) : r)$$
$$\quad |\ a \equiv k = \mathsf{Just}\ b$$
$$\quad |\ otherwise = look\ k\ r$$

é um catamorfismo de listas.

4. Considere o tipo das árvores binárias com informação nas folhas

$$\mathbf{data}\ \mathsf{LTree}\ a = \mathsf{Leaf}\ a \mid \mathsf{Fork}\ (\mathsf{LTree}\ a, \mathsf{LTree}\ a)$$

e a função

$$\begin{array}{l} \mathsf{mirror}\ (\mathsf{Leaf}\ a) = \mathsf{Leaf}\ a \\ \mathsf{mirror}\ (\mathsf{Fork}\ (x, y)) = \mathsf{Fork}\ (\mathsf{mirror}\ y, \mathsf{mirror}\ x) \end{array}$$

que "espelha" árvores binárias desse tipo.

(a) Mostre que

$$\mathsf{mirror} = (\!| \mathit{inLTree} \cdot (id + \mathsf{swap}) |\!) \tag{5}$$

onde

$$\mathit{inLTree} = [\mathsf{Leaf}\ , \mathsf{Fork}] \tag{6}$$

(b) Desenhe o digrama que representa o catamorfisno mirror.

(c) É fácil provar que mirror é um isomorfismo de árvores mostrando que a função é a sua própria inversa:

$$\mathsf{mirror} \cdot \mathsf{mirror} = id \tag{7}$$

Complete a seguinte demonstração desta propriedade:

$$\begin{array}{ll} & \mathsf{mirror} \cdot \mathsf{mirror} = id \\ \equiv & \quad \{ \ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \ \} \\ & \mathsf{mirror} \cdot (\!| \mathit{inLTree} \cdot (id + \mathsf{swap}) |\!) = (\!| \mathit{inLTree} |\!) \\ \Leftarrow & \quad \{ \ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \ \} \\ & \mathsf{mirror} \cdot \ldots = \mathit{inLTree} \cdot \ldots \\ \ldots & \quad \{ \ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \ \} \\ & (etc) \end{array}$$

5. A lei genérica de recursividade mútua generaliza a mais do que duas funções mutuamente recursivas, por exemplo a três:

$$\left\{ \begin{array}{l} f \cdot in = h \cdot \mathsf{F}\,\langle f, \langle g, j \rangle\rangle \\ g \cdot in = k \cdot \mathsf{F}\,\langle f, \langle g, j \rangle\rangle \\ j \cdot in = l \cdot \mathsf{F}\,\langle f, \langle g, j \rangle\rangle \end{array} \right. \equiv \ \langle f, \langle g, j \rangle\rangle = (\!| \langle h, \langle k, l \rangle\rangle |\!) \tag{8}$$

Justifique detalhadamente os passos do seguinte cálculo dessa versão da lei:

$$\begin{array}{ll} & \langle f, \langle g, j \rangle\rangle = (\!| \langle h, \langle k, l \rangle\rangle |\!) \\ \equiv & \quad \{ \ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \ \} \\ & \langle f, \langle g, j \rangle\rangle \cdot in = \langle h, \langle k, l \rangle\rangle \cdot \mathsf{F}\,\langle f, \langle g, j \rangle\rangle \\ \equiv & \quad \{ \ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \ \} \\ & \langle f \cdot in, \langle g, j \rangle \cdot in \rangle = \langle h \cdot \mathsf{F}\,\langle f, \langle g, j \rangle\rangle, \langle k, l \rangle \cdot \mathsf{F}\,\langle f, \langle g, j \rangle\rangle\rangle \\ \equiv & \quad \{ \ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \ \} \\ & \left\{ \begin{array}{l} f \cdot in = h \cdot \mathsf{F}\,\langle f, \langle g, j \rangle\rangle \\ \langle g, j \rangle \cdot in = \langle k, l \rangle \cdot \mathsf{F}\,\langle f, \langle g, j \rangle\rangle \end{array} \right. \\ \equiv & \quad \{ \ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \ \} \\ & \left\{ \begin{array}{l} f \cdot in = h \cdot \mathsf{F}\,\langle f, \langle g, j \rangle\rangle \\ g \cdot in = k \cdot \mathsf{F}\,\langle f, \langle g, j \rangle\rangle \\ j \cdot in = l \cdot \mathsf{F}\,\langle f, \langle g, j \rangle\rangle \end{array} \right. \end{array}$$