Cálculo de Programas

2.° ano da Licenciatura em Engenharia Informática da Universidade do Minho

2010/11 - 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. Resolva a equação $(\!|x|\!) = id$ em ordem a x e demonstre assim a lei de *relexão-cata*, válida para qualquer tipo de dados (naturais, listas, árvores, etc). Faça um diagrama que ilustre esta situação bem particular do cálculo de catamorfismos.

3. 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:

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

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

   $\mathsf{mirror}\ (\mathsf{Leaf}\ a) = \mathsf{Leaf}\ a$
   $\mathsf{mirror}\ (\mathsf{Fork}\ (x, y)) = \mathsf{Fork}\ (\mathsf{mirror}\ y, \mathsf{mirror}\ x)$

   que “espelha” árvores binárias desse tipo.

(a) Mostre que

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

onde

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

(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:

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

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

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

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

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