Cálculo de Programas

2.° ano
Lic. Ciências da Computação e Mestrado Integrado em Engenharia Informática
UNIVERSIDADE DO MINHO

2016/17 - Ficha nr.° 11

1. Recorra à lei da absorção-cata, entre outras, para verificar as seguintes propriedades sobre listas

$$\mathsf{length} = \mathsf{sum} \cdot (\mathsf{map}\ \underline{1}) \tag{F1}$$
$$\mathsf{length} = \mathsf{length} \cdot (\mathsf{map}\ f) \tag{F2}$$

onde length, sum e map são catamorfismos de listas que conhece, e sabendo que o functor de base para listas é $\mathsf{B}\ (f, g) = id + f \times g$, de onde se deriva $\mathsf{F}\ f = \mathsf{B}\ (id, f) = id + id \times f$.

2. Na sequência da questão anterior, mostre que $\mathsf{map}\ f$ tem as propriedades de functor, isto é:

$$\mathsf{map}\ id = id \tag{F3}$$
$$(\mathsf{map}\ f) \cdot (\mathsf{map}\ g) = \mathsf{map}\ (f \cdot g) \tag{F4}$$

Sugestão: recorra à absorção-cata na prova da segunda propriedade.

3. A função concat, extraída do *Prelude* do Haskell, é o catamorfismo de listas

$$\mathsf{concat} = (\![\mathsf{nil}\ , \mathsf{conc}]\!) \tag{F5}$$

onde $\mathsf{conc}\ (x, y) = x +\!\!+ y$ e $\mathsf{nil}\ \_ = [\,]$. Apresente justificações para a prova da propriedade

$$\mathsf{length} \cdot \mathsf{concat} = \mathsf{sum} \cdot \mathsf{map\ length} \tag{F6}$$

que a seguir se apresenta, onde é de esperar que as leis de *fusão*-cata e *absorção*-cata desempenhem um papel importante:

$$
\begin{array}{cl}
& \mathsf{length} \cdot \mathsf{concat} = \mathsf{sum} \cdot \mathsf{map\ length} \\
\equiv & \qquad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& \mathsf{length} \cdot \mathsf{concat} = (\![\underline{0}\ , \mathsf{add}]\!) \cdot (\!\mathsf{in} \cdot \mathsf{B}\ (\mathsf{length}, id)\!) \\
\equiv & \qquad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& \mathsf{length} \cdot \mathsf{concat} = (\![\underline{0}\ , \mathsf{add}] \cdot (id + \mathsf{length} \times id)\!) \\
\equiv & \qquad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& \mathsf{length} \cdot (\![\mathsf{nil}\ , \mathsf{conc}]\!) = (\![\underline{0}\ , \mathsf{add} \cdot (\mathsf{length} \times id)]\!) \\
\Leftarrow & \qquad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& \mathsf{length} \cdot [\mathsf{nil}\ , \mathsf{conc}] = [\underline{0}\ , \mathsf{add} \cdot (\mathsf{length} \times id)] \cdot (id + id \times \mathsf{length})
\end{array}
$$

$$
\begin{array}{l}
\equiv \qquad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
\qquad \left\{ \begin{array}{l} \mathsf{length} \cdot \mathsf{nil} = \underline{0} \\ \mathsf{length} \cdot \mathsf{conc} = \mathsf{add} \cdot (\mathsf{length} \times id) \cdot (id \times \mathsf{length}) \end{array} \right. \\
\equiv \qquad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
\qquad \mathsf{length} \cdot \mathsf{conc} = \mathsf{add} \cdot (\mathsf{length} \times \mathsf{length}) \\
\equiv \qquad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
\qquad true \\
\Box
\end{array}
$$

4. Mostre que o anamorfismo $\mathsf{repeat} = [\!(i_2 \cdot \langle id, id \rangle)\!]$ é tal que a propriedade

$$\mathsf{map}\ f \cdot \mathsf{repeat} = \mathsf{repeat} \cdot f \tag{F7}$$

se verifica.

5. Recorra à lei de absorção-cata para demonstrar a propriedade

$$count \cdot (\mathsf{BTree}\ f) \quad = \quad count \tag{F8}$$

onde $\mathsf{BTree}\ A \xrightarrow{count} \mathbb{N}_0$ é o catamorfismo

$$count = (\!|[\mathsf{zero}\ , \mathsf{succ} \cdot \mathsf{add} \cdot \pi_2]|\!)$$

onde $\mathsf{zero}\ \_ = 0$, $\mathsf{succ}\ n = n + 1$ e $\mathsf{add}\ (a, b) = a + b$. **NB:** recorda-se que a base do tipo BTree é $\mathsf{B}\ (f, g) = id + f \times (g \times g)$.

6. Recorra às leis dos catamorfismos para demonstrar a propriedade natural

$$(\mathsf{T}\ f) \cdot \mathsf{mirror} \quad = \quad \mathsf{mirror} \cdot (\mathsf{T}\ f) \tag{F9}$$

onde mirror é o catamorfismo

$$
\begin{array}{l}
\mathsf{mirror} :: \mathsf{LTree}\ a \rightarrow \mathsf{LTree}\ a \\
\mathsf{mirror} = (\!|\ \mathbf{in} \cdot (id + \mathsf{swap})|\!)
\end{array}
$$

que “espelha”uma árvore e $\mathsf{T}\ f = (\!|\ \mathbf{in} \cdot (f + id)|\!)$ é o correspondente functor de tipo.