Cálculo de Programas

Lic. C. Computação (2º ano)
Lic./Mest. em Engenharia Informática (3º ano)
UNIVERSIDADE DO MINHO

2021/22 - Ficha nr.° 9

1. O diagrama genérico de um catamorfismo de gene g sobre o *tipo paramétrico* $\mathsf{T}\ X \cong \mathsf{B}\ (X, \mathsf{T}\ X)$ cuja base é o bifunctor B, bem como a sua propriedade universal, são representados a seguir:

$$\begin{tikzcd}
\mathsf{T}\ X \arrow[d, "(\!|g|\!)"'] & \mathsf{B}\ (X, \mathsf{T}\ X) \arrow[l, "\mathsf{in}"'] \arrow[d, "\mathsf{B}\ (id, (\!|g|\!)) = \mathsf{F}\ (\!|g|\!)"] \\
B & \mathsf{B}\ (X, B) \arrow[l, "g"]
\end{tikzcd}
\qquad\qquad
k = (\!|g|\!) \;\equiv\; k \cdot \mathsf{in} = g \cdot \underbrace{\mathsf{B}\ (id, k)}_{\mathsf{F}\ k}$$

(Repare-se que se tem sempre $\mathsf{F}\ k = \mathsf{B}\ (id, k)$.) Partindo da definição *genérica* de map associado ao tipo T, $\mathsf{T}\ f = (\!|\mathsf{in} \cdot \mathsf{B}\ (f, id)|\!)$ dada no formulário, mostre que o map das sequências finitas (vulg. listas) é a função

$$\begin{array}{l}
f^*\ [\,] = [\,] \\
f^*\ (h : t) = f\ h : f^*\ t
\end{array}$$

2. 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.

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

$$\mathsf{concat} = (\!|[\mathsf{nil}\ , \mathsf{conc}]|\!) \tag{F3}$$

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

$$\mathsf{length} \cdot \mathsf{concat} = \mathsf{sum} \cdot \mathsf{map\ length} \tag{F4}$$

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

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

$\Leftarrow \quad \{ \; \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \; \}$

$\quad \mathsf{length} \cdot [\mathsf{nil}\,, \mathsf{conc}] = [\underline{0}\,, \mathsf{add} \cdot (\mathsf{length} \times id)] \cdot (id + id \times \mathsf{length})$

$\equiv \quad \{ \; \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \; \}$

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

$\equiv \quad \{ \; \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \; \}$

$\quad \mathsf{length} \cdot \mathsf{conc} = \mathsf{add} \cdot (\mathsf{length} \times \mathsf{length})$

$\equiv \quad \{ \; \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \; \}$

$\quad true$

$\Box$

4. Consultando as bibliotecas em Haskell disponíveis no material pedagógico, complete o seguinte quadro relativo aos tipos indutivos que aí se codificam:

T	**Descrição**	in	$\mathsf{B}\ (X, Y)$	$\mathsf{B}\ (f, g)$	$\mathsf{F}\ f$	$\mathsf{T}\ f$
A^*	Sequências finitas de A					
$\mathsf{BTree}\ A$	Árvores binárias de A					
$\mathsf{LTree}\ A$	Árvores com A nas folhas					
$\mathbb{N}_0$	Números naturais					

5. Considere a função $depth = (\![\,[\mathsf{one}\,, \mathsf{succ} \cdot umax]\,]\!)$ que calcula a profundidade de árvores do tipo

$$\mathsf{T}\ X = \mathsf{LTree}\ X \qquad \left\{ \begin{array}{l} \mathsf{B}\ (X, Y) = X + Y^2 \\ \mathsf{B}\ (f, g) = f + g^2 \end{array} \right. \qquad \mathsf{in} = [Leaf\,, Fork]$$

Haskell: **data** $\mathsf{LTree}\ a = Leaf\ a \mid Fork\ (\mathsf{LTree}\ a, \mathsf{LTree}\ a)$

onde $umax\ (a, b) = max\ a\ b$. Mostre, por absorção-cata, que a profundidade de uma árvore t não é alterada quando aplica uma função f a todas as suas folhas:

$$depth \cdot \mathsf{LTree}\ f = depth \tag{F5}$$

6. Um *anamorfismo* é um *"catamorfismo ao contrário"*, isto é, uma função $k : A \to \mathsf{T}$ tal que

$$k = \mathsf{in} \cdot \mathsf{F}\ k \cdot g \tag{F6}$$

escrevendo-se $k = [\!(g)\!]$. Mostre que o anamorfismo de listas

$$k = [\!((id + \langle f, id\rangle) \cdot \mathsf{out}_{\mathbb{N}_0})\!] \tag{F7}$$

descrito pelo diagrama

$$\begin{tikzcd}
\mathbb{N}_0{}^* & & 1 + \mathbb{N}_0 \times \mathbb{N}_0{}^* \arrow[ll, "\mathsf{in}"'] \\
\mathbb{N}_0 \arrow[u, "k"] \arrow[r, "\mathsf{out}_{\mathbb{N}_0}"'] & 1 + \mathbb{N}_0 \arrow[r, "id + \langle f, id\rangle"'] & 1 + \mathbb{N}_0 \times \mathbb{N}_0 \arrow[u, "id + id \times k"']
\end{tikzcd}$$

é a função

$$\begin{array}{l} k\ 0 = [\,] \\ k\ (n + 1) = (2\ n + 1) : k\ n \end{array}$$

para $f\ n = 2\ n + 1$. (Que faz esta função?)