Cálculo de Programas

2.° ano das Licenciaturas em
Engenharia Informática (LEI) e Ciências da Computação (LCC)
da Universidade do Minho

2009/10 - Ficha nr.° 8

1. A lei universal

$$\begin{tikzcd} T \arrow[d, "(\!|g|\!)"'] & \mathsf{F}\,T \arrow[l, "\mathbf{in}"'] \arrow[d, "\mathsf{F}(\!|g|\!)"] \\ B & \mathsf{F}\,B \arrow[l, "g"] \end{tikzcd} \qquad k = (\!|g|\!) \ \equiv \ k \cdot \mathbf{in} = g \cdot \mathsf{F}\ k \tag{1}$$

dá-nos meios para raciocinarmos sobre uma função k sempre que esta se pode escrever como um catamorfismo de gene g sobre o tipo indutivo T.

Nesta disciplina vimos vários exemplos de T, por exemplo os números naturais ($\mathbb{N}_0$), as listas ($[A]$) e dois tipos de árvores binárias,

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

e

$$\mathbf{data}\ \mathsf{BTree}\ a = \mathsf{Empty} \mid \mathsf{Node}\ (a, (\mathsf{BTree}\ a, \mathsf{BTree}\ a))$$

A estes tipos podemos acrescentar outros como, por exemplo, o das listas não vazias

$$\mathbf{data}\ \mathsf{NEList}\ a = \mathsf{Sing}\ a \mid \mathsf{Add}\ (a, \mathsf{NEList}\ a)$$

e o das chamadas "rose trees":

$$\mathbf{data}\ \mathsf{Rose}\ a = \mathsf{Rose}\ a\ [\mathsf{Rose}\ a]$$

Preencha o quadro seguinte, em que a coluna da esquerda identifica funções sobre o tipo da coluna T, funções essas que conhece ou cujo significado facilmente identifica:

k	g	F X	F f	T	**in**	B
length		$1 + A \times X$		[A]	$[nil\ , \mathsf{cons}]$	$\mathbb{N}_0$
length			$id + id \times f$	NEList A		$\mathbb{N}_0$
$count$					$[\mathsf{Leaf}\ , \mathsf{Fork}]$	$\mathbb{N}_0$
$listify$	$[singl\ , +\!\!+]$			$LTree\ A$		$[A]$
$reverse$					$[nil\ , \mathsf{cons}]$	
sum		$1 + A \times X^2$				
sum					$[\mathsf{Sing}\ , \mathsf{Add}]$	
mirror	$\mathbf{in} \cdot (id + \mathsf{swap})$				$[\mathsf{Leaf}\ , \mathsf{Fork}]$	
filter p			$id + id \times f$	[A]		[A]
$gmax$	$[id\ , max]$	$A + A \times X$				A
$gmax$	$[id\ , max]$				$[\mathsf{Leaf}\ , \mathsf{Fork}]$	A

2. Considere 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" uma árvore binária de tipo LTree e que, como se viu na questão anterior, é o catamorfismo

$$\mathsf{mirror} = (\!|\mathit{inLTree} \cdot (id + \mathsf{swap})|\!) \tag{2}$$

onde

$$\mathit{inLTree} = [\mathsf{Leaf}\ , \mathsf{Fork}] \tag{3}$$

(a) Comece por desenhar o digrama que representa o ctamorfisno mirror.

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

$$\mathsf{mirror} \cdot \mathsf{mirror} = id \tag{4}$$

Complete a seguinte demonstração desta propriedade:

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

3. Considere o diagrama

$$\begin{tikzcd} {[\mathbb{N}_0]} \arrow[rr, bend left, "outList"] \arrow[rr, phantom, "\cong"] & & 1 + \mathbb{N}_0 \times [\mathbb{N}_0] \arrow[ll, bend left, "inList"] \\ \mathbb{N}_0 \arrow[u, "a"] \arrow[r, bend left, "outNat"] \arrow[r, phantom, "\cong"] & 1 + \mathbb{N}_0 \arrow[l, bend left, "inNat"] \arrow[r, "id + \langle x, id\rangle"'] & 1 + \mathbb{N}_0 \times \mathbb{N}_0 \arrow[u, "id + id \times a"'] \end{tikzcd}$$

que capta a seguinte propriedade da função a,

$$a \quad = \quad inList \cdot (id + id \times a) \cdot (id + \langle x, id\rangle) \cdot outNat$$

para $inNat = [\underline{0}\ , \mathsf{succ}]$ e $inList = [nil\ , \mathsf{cons}]$, onde $nil = \underline{[\,]}$ e $\mathsf{cons}\ (h, t) = h : t$.

Funções com esta estrutura dizem-se *anamorfismos* do seu tipo de saída (listas de naturais neste caso).

(a) Explique por que é que a propriedade dada se pode escrever, alternativamente, sob a forma

$$a \cdot inNat \quad = \quad inList \cdot (id + \langle x, a\rangle) \tag{5}$$

(b) Diga o que faz a função a para as situações seguintes: $x = \mathsf{succ}$ e $x = \mathsf{succ} \cdot (2*)$.