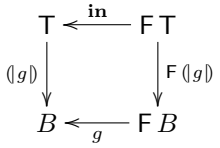
Cálculo de Programas

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

2010/11 - Ficha nr.° 9

1. Recorde o diagrama genérico de um catamorfsimo de gene g sobre o tipo T e a sua propriedade universal:



$$k = (\!|g|\!) \;\equiv\; k \cdot \mathbf{in} = g \cdot \mathsf{F}\ k$$

Nesta disciplina vimos vários exemplos de T, por exemplo os números naturais $\mathbb{N}_0$, 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	$\mathsf{F}\ \mathsf{X}$	$\mathsf{F}\ f$	T	$\mathbf{in}$	B
length		$1 + A \times X$		[A]	$[nil\ ,\mathsf{cons}]$	$\mathbb{N}_0$
length			$id + id \times f$	NEList A		$\mathbb{N}_0$
count					[Leaf , Fork]	$\mathbb{N}$
listify	$[singl\ ,\widehat{(+\!+)}]$			LTree A		$[A]$
reverse					$[nil\ ,\mathsf{cons}]$	
sum		$1 + A \times X^2$				
sum					[Sing , Add]	
mirror	$\mathbf{in} \cdot (id + \mathsf{swap})$				[Leaf , Fork]	
mirror					[Empty , Node]	
filter p			$id + id \times f$	[A]		[A]
gmax	$[id\ , max]$	$A + A \times X$				A
gmax	$[id\ , max]$				[Leaf , Fork]	A

2. A função que calcula a média dos elementos que se guardam numa árvore de tipo LTree pode escrever-se sob a forma

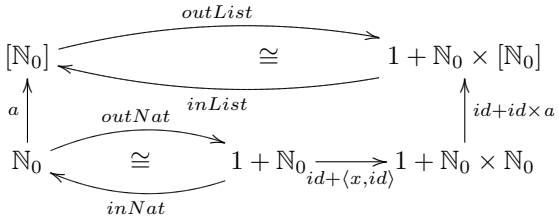
$$avg\ t = \frac{\mathsf{sum}\ t}{\mathsf{count}\ t}$$

em que sum e count são catamorfismos já identificdos. Essa função, que necessita de duas visitas à árvore t para fazer esse cálculo, pode ser convertida numa que só faz uma tal visita,

$$avg\ t = s\ /\ c\ \mathbf{where}\ (s, c) = (\!|g|\!)\ t$$

recorrendo à lei de *banana-split*. Calcule o gene g e converta $(\!|g|\!)$ para Haskell com variáveis.

3. Considere o diagrama



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 $\mathsf{cons}\ (h, t) = h : t$.

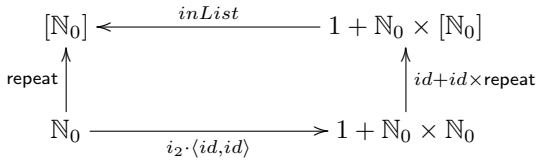
Funções com esta estrutura dizem-se *anamorfismos* do seu tipo de saída (listas de naturais neste caso).

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

$$a \cdot inNat \quad = \quad inList \cdot (id + \langle x, a\rangle) \tag{1}$$

(b) Diga o que faz a função a para $x = \mathsf{succ}$.

4. Investigue o comportamento do anamorfismo repeat cujo diagrama é:



Em particular, derive a definição *pointwise* de repeat.

5. Seja dada a função

$$odd = (id + \langle impar, id\rangle) \cdot outNat$$
$$\mathbf{where}\ impar\ n = 2 * n + 1$$

Faça o diagrama do anamorfismo de listas $odds = [\![odd]\!]$ e derive a correspondente versão em Haskell com váriáveis.