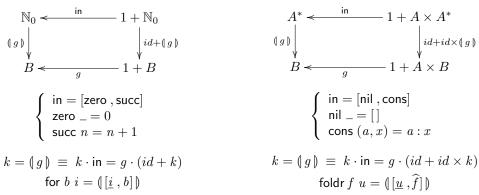
Cálculo de Programas

2.° ano

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

2020/21 - Ficha nr.º 6

1. Os diagramas seguintes representam as **propriedades universais** que definem o combinador **catamorfismo** para dois tipos de dados — números naturais \mathbb{N}_0 à esquerda e listas finitas A^* à direita:



onde \hat{f} abrevia uncurry f.

(a) Tendo em conta o diagrama da esquerda, codifique, em Haskell

$$(\!(g\,)\!) = g \cdot (id + (\!(g\,)\!)) \cdot \mathsf{out}$$
 e
$$\mathsf{for} \ b \ i = (\!([i\,,b])\!)$$

em que out foi calculada numa ficha anterior. De seguida, codifique

$$f = \pi_2 \cdot aux \text{ where } aux = \text{for } \langle \text{succ} \cdot \pi_1, \text{mul} \rangle \ (1, 1)$$

e inspeccione o comportamento de f. Que função é essa?

- (b) Identifique como catamorfismos de listas as funções seguintes, indicando o gene g para cada caso:¹
 - i. k é a função que multiplica todos os elementos de uma lista
 - ii. k = reverse
 - iii. k = concat
 - iv. k é a função map f, para um dado $f:A\to B$
 - v. k é a função que calcula o máximo de uma lista de números naturais (\mathbb{N}_0^*) .
 - vi. k = filter p onde

filter
$$p[] = []$$

filter $p(h:t) = x +$ filter $p(t) = x +$ filte

¹Apoie a sua resolução com diagramas.

- 2. Considere o seguinte inventário de quatro tipos de árvores:
 - (a) Árvores com informação de tipo A nos nós:

$$\mathsf{T} = \mathsf{BTree}\ A \qquad \left\{ \begin{array}{l} \mathsf{F}\ X = 1 + A \times X^2 \\ \mathsf{F}\ f = id + id \times f^2 \end{array} \right. \quad \mathsf{in} = [\underline{Empty}\ , Node]$$
 Haskell: $\mathbf{data}\ \mathsf{BTree}\ a = Empty\ |\ Node\ (a, (\mathsf{BTree}\ a, \mathsf{BTree}\ a))$

(b) Árvores com informação de tipo A nas folhas:

$$\mathsf{T} = \mathsf{LTree}\ A \qquad \begin{cases} \mathsf{F}\ X = A + X^2 \\ \mathsf{F}\ f = id + f^2 \end{cases} \qquad \mathsf{in} = [\mathit{Leaf}\ , \mathit{Fork}]$$
 Haskell: $\mathbf{data}\ \mathsf{LTree}\ a = \mathit{Leaf}\ a \mid \mathit{Fork}\ (\mathsf{LTree}\ a, \mathsf{LTree}\ a)$

(c) Árvores com informação nos nós e nas folhas:

T = FTree
$$B$$
 A
$$\begin{cases} \mathsf{F}\ X = B + A \times X^2 \\ \mathsf{F}\ f = id + id \times f^2 \end{cases} \quad \mathsf{in} = [\mathit{Unit}\ , \mathit{Comp}]$$
 Haskell: $\mathbf{data}\ \mathsf{FTree}\ b\ a = \mathit{Unit}\ b \mid \mathit{Comp}\ (a, (\mathsf{FTree}\ b\ a, \mathsf{FTree}\ b\ a))$

(d) Árvores de expressão:

T =
$$Expr \ V \ O$$

$$\begin{cases} F \ X = V + O \times X^* \\ F \ f = id + id \times \mathsf{map} \ f \end{cases}$$
 in = $[Var \ , Op]$ Haskell: $\mathbf{data} \ Expr \ v \ o = Var \ v \ | \ Op \ (o, [Expr \ v \ o])$

Defina o gene q para cada um dos catamorfismos seguintes desenhando, para cada caso, o diagrama correspondente:

- zeros = (g) substitui todas as folhas de uma árvore de tipo (2b) por zero.
- conta = (g) conta o número de nós de uma árvore de tipo (2a).
- mirror = (g) espelha uma árvore de tipo (2b), i.e., roda-a de 180°.
- converte = (g) converte árvores de tipo (2c) em árvores de tipo (2a) eliminando os Bs que estão na primeira.
- vars = (g) lista as variáveis de uma árvore expressão de tipo (2d).
- 3. Implemente mirror = (g) em Haskell definindo previamente outLTree e o combinador cataLTree (catamorfismo de LTrees).
- 4. Converta a função vars do exercício 2 numa função com variáveis em Haskell sem quaisquer combinadores pointfree.
- 5. A função seguinte, em Haskell

$$sumprod\ a\ [\]=0$$

 $sumprod\ a\ (h:t)=a*h+sumprod\ a\ t$

é o catamorfismo de listas

$$sumprod \ a = ([zero, add \cdot ((a*) \times id)])$$
 (F1)

onde zero = 0 e add (x, y) = x + y. Mostre, como exemplo de aplicação da propriedade de fusão-cata para listas, que

$$sumprod \ a = (a*) \cdot sum \tag{F2}$$

onde sum = ([zero , add]). NB: não ignore propriedades elementares da aritmética que lhe possam ser úteis.