

# Cálculo de Programas

2.º ano

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

2018/19 - Ficha nr.º 6

1. Considere o seguinte inventário de tipos indutivos de dados:

(a) Números naturais:

$$T = \mathbb{N}_0 \quad \begin{cases} F X = 1 + X \\ F f = id + f \end{cases} \quad \text{in} = [\underline{0}, \text{succ}]$$

Haskell: *Int* inclui  $\mathbb{N}_0$ .

(b) Listas de elementos em  $A$ :

$$T = A^* \quad \begin{cases} F X = 1 + A \times X \\ F f = id + id \times f \end{cases} \quad \text{in} = [\text{nil}, \text{cons}]$$

Haskell:  $[a]$

(c) Árvores com informação de tipo  $A$  nos nós:

$$T = \text{BTree } A \quad \begin{cases} F X = 1 + A \times X^2 \\ F f = id + id \times f^2 \end{cases} \quad \text{in} = [\underline{\text{Empty}}, \text{Node}]$$

Haskell: `data BTree a = Empty | Node (a, (BTree a, BTree a))`

(d) Árvores com informação de tipo  $A$  nas folhas:

$$T = \text{LTree } A \quad \begin{cases} F X = A + X^2 \\ F f = id + f^2 \end{cases} \quad \text{in} = [\text{Leaf}, \text{Fork}]$$

Haskell: `data LTree a = Leaf a | Fork (LTree a, LTree a)`

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

$$T = \text{FTree } B A \quad \begin{cases} F X = B + A \times X^2 \\ F f = id + id \times f^2 \end{cases} \quad \text{in} = [\text{Unit}, \text{Comp}]$$

Haskell: `data FTree b a = Unit b | Comp (a, (FTree b a, FTree b a))`

(f) Árvores de expressão:

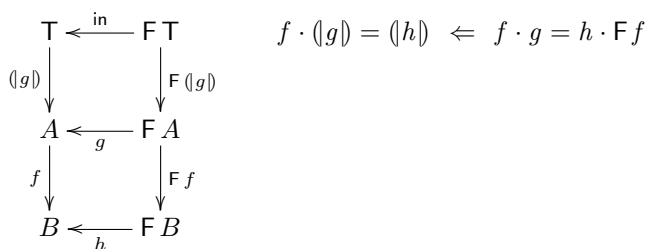
$$T = \text{Expr } V O \quad \begin{cases} F X = V + O \times X^* \\ F f = id + id \times \text{map } f \end{cases} \quad \text{in} = [\text{Var}, \text{Op}]$$

Haskell: `data Expr v o = Var v | Op (o, [Expr v o])`

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

- $\text{positivos} = \langle g \rangle$  — retira de uma lista de inteiros (1b) todos os números negativos
- $\text{concat} = \langle g \rangle$  — concatena uma lista de listas numa só lista (1b)
- $\text{zeros} = \langle g \rangle$  — substitui todas as folhas de uma árvore de tipo (1d) por zero.
- $\text{conta} = \langle g \rangle$  — conta o número de nós de uma árvore de tipo (1c).
- $\text{mirror} = \langle g \rangle$  — espelha uma árvore de tipo (1d), i.e., roda-a de 180°.
- $\text{converte} = \langle g \rangle$  — converte árvores de tipo (1e) em árvores de tipo (1c) eliminando os  $B$ s que estão na primeira.

- $vars = \langle g \rangle$  — lista as variáveis de uma árvore expressão de tipo  $(1f)$ .
2. Converta a função  $vars$  do exercício 1 numa função com variáveis em Haskell sem quaisquer combinadores *pointfree*.
  3. O diagrama que se segue representa a lei genérica para fusão de catamorfismos (identifique-a no formulário)



Apresente justificações para o cálculo que se segue dessa lei:

$$\begin{aligned}
 & f \cdot \langle g \rangle = \langle h \rangle \\
 \equiv & \quad \{ \dots \} \\
 & f \cdot \langle g \rangle \cdot \text{in} = h \cdot F(f \cdot \langle g \rangle) \\
 \equiv & \quad \{ \dots \} \\
 & f \cdot \langle g \rangle \cdot \text{in} = h \cdot Ff \cdot F \langle g \rangle \\
 \equiv & \quad \{ \dots \} \\
 & f \cdot g \cdot F \langle g \rangle = h \cdot Ff \cdot F \langle g \rangle \\
 \leftarrow & \quad \{ \dots \} \\
 & f \cdot g = h \cdot Ff \\
 & \square
 \end{aligned}$$

4. A função seguinte, em Haskell

$$\begin{aligned}
 \text{sumprod } a \ [] &= 0 \\
 \text{sumprod } a \ (h : t) &= a * h + \text{sumprod } a \ t
 \end{aligned}$$

é o catamorfismo de listas

$$\text{sumprod } a = \langle [\text{zero}, \text{add} \cdot ((a*) \times \text{id})] \rangle \tag{F1}$$

onde  $\text{zero} = 0$  e  $\text{add } (x, y) = x + y$ .

Mostre, como exemplo de aplicação da propriedade de **fusão-cata** para listas, que

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

onde  $\text{sum} = \langle [\text{zero}, \text{add}] \rangle$ . **NB:** não ignore propriedades elementares da aritmética que lhe possam ser úteis.