# Cálculo de Programas

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

## 2018/19 - Ficha nr.° 6

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

   (a) Números naturais:

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

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

   (b) Listas de elementos em $A$:

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

   Haskell: $[\,a\,]$

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

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

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

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

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

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

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

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

   Haskell: $\mathbf{data}\ \mathsf{FTree}\ b\ a = Unit\ b \mid Comp\ (a, (\mathsf{FTree}\ b\ a, \mathsf{FTree}\ b\ a))$

   (f) Árvores de expressão:

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

   Haskell: $\mathbf{data}\ \mathsf{Expr}\ v\ o = Var\ v \mid Op\ (o, [\mathsf{Expr}\ v\ o])$

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

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

- $vars = \llparenthesis g \rrparenthesis$ — 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)

$$
\begin{tikzcd}
\mathsf{T} \arrow[d, "\llparenthesis g \rrparenthesis"'] & \mathsf{F}\,\mathsf{T} \arrow[l, "\mathsf{in}"'] \arrow[d, "\mathsf{F}\,\llparenthesis g \rrparenthesis"] \\
A \arrow[d, "f"'] & \mathsf{F}\,A \arrow[l, "g"] \arrow[d, "\mathsf{F}\,f"] \\
B & \mathsf{F}\,B \arrow[l, "h"]
\end{tikzcd}
\qquad
f \cdot \llparenthesis g \rrparenthesis = \llparenthesis h \rrparenthesis \;\Leftarrow\; f \cdot g = h \cdot \mathsf{F}\,f
$$

Apresente justificações para o cálculo que se segue dessa lei:

$$
\begin{array}{ll}
& f \cdot \llparenthesis g \rrparenthesis = \llparenthesis h \rrparenthesis \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& f \cdot \llparenthesis g \rrparenthesis \cdot \mathsf{in} = h \cdot \mathsf{F}\,(f \cdot \llparenthesis g \rrparenthesis) \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& f \cdot \llparenthesis g \rrparenthesis \cdot \mathsf{in} = h \cdot \mathsf{F}\,f \cdot \mathsf{F}\,\llparenthesis g \rrparenthesis \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& f \cdot g \cdot \mathsf{F}\,\llparenthesis g \rrparenthesis = h \cdot \mathsf{F}\,f \cdot \mathsf{F}\,\llparenthesis g \rrparenthesis \\
\Leftarrow & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& f \cdot g = h \cdot \mathsf{F}\,f \\
\Box &
\end{array}
$$

4. A função seguinte, em Haskell

$$
\begin{array}{l}
sumprod\ a\ [\,] = 0 \\
sumprod\ a\ (h : t) = a * h + sumprod\ a\ t
\end{array}
$$

é o catamorfismo de listas

$$sumprod\ a \quad = \quad \llparenthesis [\mathsf{zero}\ , \mathsf{add} \cdot ((a*) \times id)] \rrparenthesis \tag{F1}$$

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

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

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

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