# Cálculo de Programas

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

2018/19 - Ficha nr.° 10

1. Considere a função $depth = (\!|[\mathsf{one}\ , \mathsf{succ} \cdot umax]|\!)$ que calcula a profundidade de árvores de tipo LTree, onde $umax\ (a, b) = max\ a\ b$.

   Mostre, por absorção-cata, que a profundidade de uma árvore $t$ não é alterada quando aplica uma função $f$ a todas as suas folhas:

   $$depth \cdot \mathsf{LTree}\ f = depth \tag{F1}$$

2. O formulário desta disciplina apresenta duas definições alternativas para o functor $\mathsf{T}\ f$ de um tipo indutivo, uma como *catamorfismo* e outra como *anamorfismo*. Identifique-as e acrescente justificações à seguinte prova de que essas definições são equivalentes:

$$\begin{array}{ll}
& \mathsf{T}\ f = (\!|\mathsf{in} \cdot \mathsf{B}\ (f, id)|\!) \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& \mathsf{T}\ f \cdot \mathsf{in} = \mathsf{in} \cdot \mathsf{B}\ (f, id) \cdot \mathsf{F}\ (\mathsf{T}\ f) \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& \mathsf{T}\ f \cdot \mathsf{in} = \mathsf{in} \cdot \mathsf{B}\ (id, \mathsf{T}\ f) \cdot \mathsf{B}\ (f, id) \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& \mathsf{out} \cdot \mathsf{T}\ f = \mathsf{F}\ (\mathsf{T}\ f) \cdot \mathsf{B}\ (f, id) \cdot \mathsf{out} \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& \mathsf{T}\ f = [\!(\mathsf{B}\ (f, id) \cdot \mathsf{out})\!] \\
\square &
\end{array}$$

3. Mostre que o catamorfismo de listas $\mathsf{length} = (\!|[\mathsf{zero}\ , \mathsf{succ} \cdot \pi_2]|\!)$ é a mesma função que o *anamorfismo* de naturais $[\!((id + \pi_2) \cdot \mathsf{out}_{\mathsf{List}})\!]$.

4. Mostre que a função mirror da ficha nr.º 7 se pode definir como o anamorfismo

   $$\mathsf{mirror} = [\!((id + \mathsf{swap}) \cdot \mathsf{out})\!] \tag{F2}$$

   onde out é a conversa de in. Volte a demonstrar a propriedade $\mathsf{mirror} \cdot \mathsf{mirror} = id$, desta vez recorrendo à lei de fusão dos anamorfismos.

5. Mostre que o anamorfismo $\mathsf{repeat} = [\!(i_2 \cdot \langle id, id \rangle)\!]$ definido pelo diagrama

$$\begin{tikzcd}
A^{\infty} & & 1 + A \times A^{\infty} \arrow[ll, "\mathsf{in}"'] \\
A \arrow[u, "\mathsf{repeat}"] \arrow[r, "\langle id, id \rangle"'] & A \times A \arrow[r, "i_2"'] & 1 + A \times A \arrow[u, "id + id \times \mathsf{repeat}"']
\end{tikzcd}$$

é a função:

$$\mathsf{repeat}\ a = a : \mathsf{repeat}\ a$$

Recorrendo às leis dos anamorfismos mostre que, apesar de não terminar[1], repeat satisfaz a propriedade:

$$\mathsf{map}\ f \cdot \mathsf{repeat} = \mathsf{repeat} \cdot f \tag{F3}$$

("Verifique" este facto comparando, por exemplo, $(\mathsf{take}\ 10 \cdot \mathsf{map}\ \mathsf{succ} \cdot \mathsf{repeat})\ 1$ com $(\mathsf{take}\ 10 \cdot \mathsf{repeat} \cdot \mathsf{succ})\ 1$.)

6. Numa página de STACK OVERFLOW[2] alguém respondeu afirmativamente à pergunta

   *Pode fazer-se* unzip *num só passo?*

   com a versão

$$\begin{array}{l}
\mathsf{unzip}\ [\,] = ([\,], [\,]) \\
\mathsf{unzip}\ ((a, b) : xs) = (a : as, b : bs)\ \mathbf{where}\ (as, bs) = \mathsf{unzip}\ xs
\end{array}$$

   Ora o que essa página não faz é explicar como é que os dois passos de

$$\mathsf{unzip}\ xs = (\mathsf{map}\ \pi_1\ xs, \mathsf{map}\ \pi_2\ xs)$$

   se fundem num só. Complete a derivação que se resume a seguir dessa evidência:

$$\begin{array}{cl}
& \mathsf{unzip}\ xs = (\mathsf{map}\ \pi_1\ xs, \mathsf{map}\ \pi_2\ xs) \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& \mathsf{unzip} = \langle \mathsf{map}\ \pi_1, \mathsf{map}\ \pi_2 \rangle \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& \mathsf{unzip} = \langle (\!|\mathsf{in} \cdot \mathsf{B}\ (\pi_1, id)|\!), (\!|\mathsf{in} \cdot \mathsf{B}\ (\pi_2, id)|\!) \rangle \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& \mathsf{unzip} = (\!|\langle \mathsf{in} \cdot \mathsf{B}\ (\pi_1, id) \cdot \mathsf{F}\ \pi_1, \mathsf{in} \cdot \mathsf{B}\ (\pi_2, id) \cdot \mathsf{F}\ \pi_2 \rangle|\!) \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& \mathsf{unzip} = (\!|\langle \mathsf{in} \cdot \mathsf{B}\ (\pi_1, \pi_1), \mathsf{in} \cdot \mathsf{B}\ (\pi_2, \pi_2) \rangle|\!) \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& \mathsf{unzip} \cdot \mathsf{in} = \langle [\mathsf{nil}\ , \mathsf{cons} \cdot (\pi_1 \times \pi_1)], [\mathsf{nil}\ , \mathsf{cons} \cdot (\pi_2 \times \pi_2)] \rangle \cdot (id + id \times \mathsf{unzip}) \\
\equiv & \quad \{ \text{ ... alguns passos mais ... } \} \\
& \ldots. \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& \left\{ \begin{array}{l}
\mathsf{unzip}\ [\,] = ([\,], [\,]) \\
\mathsf{unzip}\ ((a, b) : xs) = ((a : as), (b : bs))\ \mathbf{where}\ (as, bs) = \mathsf{unzip}\ xs
\end{array} \right.
\end{array}$$

[1]Por isso usamos, no diagrama, $A^{\infty}$ em vez de $A^*$, para incluir também as listas infinitas.

[2]Cf. https://stackoverflow.com/questions/18287848/unzip-in-one-pass.