

# Cálculo de Programas

2.º ano

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

2018/19 - Ficha nr.º 10

1. Considere a função  $depth = \llbracket [one, succ \cdot umax] \rrbracket$  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:

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

2. O formulário desta disciplina apresenta duas definições alternativas para o functor  $T\ f$  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:

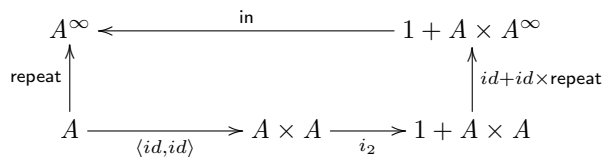
$$\begin{aligned} T\ f &= \llbracket in \cdot B\ (f, id) \rrbracket \\ \equiv & \{ \dots \} \\ T\ f \cdot in &= in \cdot B\ (f, id) \cdot F\ (T\ f) \\ \equiv & \{ \dots \} \\ T\ f \cdot in &= in \cdot B\ (id, T\ f) \cdot B\ (f, id) \\ \equiv & \{ \dots \} \\ out \cdot T\ f &= F\ (T\ f) \cdot B\ (f, id) \cdot out \\ \equiv & \{ \dots \} \\ T\ f &= \llbracket B\ (f, id) \cdot out \rrbracket \\ \square \end{aligned}$$

3. Mostre que o catamorfismo de listas  $length = \llbracket [zero, succ \cdot \pi_2] \rrbracket$  é a mesma função que o *anamorfismo* de naturais  $\llbracket (id + \pi_2) \cdot out_{List} \rrbracket$ .
4. Mostre que a função *mirror* da ficha nr.º 7 se pode definir como o anamorfismo

$$mirror = \llbracket (id + swap) \cdot out \rrbracket \tag{F2}$$

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

5. Mostre que o anamorfismo  $\text{repeat} = \llbracket i_2 \cdot \langle id, id \rangle \rrbracket$  definido pelo diagrama



é a função:

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

Recorrendo às leis dos anamorfismos mostre que, apesar de não terminar<sup>1</sup>,  $\text{repeat}$  satisfaz a propriedade:

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

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

6. Numa página de STACK OVERFLOW<sup>2</sup> alguém respondeu afirmativamente à pergunta

*Pode fazer-se unzip num só passo?*

com a versão

$$\begin{aligned}
 \text{unzip } [] &= ([], []) \\
 \text{unzip } ((a, b) : xs) &= (a : as, b : bs) \textbf{ where } (as, bs) = \text{unzip } xs
 \end{aligned}$$

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

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

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

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

<sup>1</sup>Por isso usamos, no diagrama,  $A^\infty$  em vez de  $A^*$ , para incluir também as listas infinitas.

<sup>2</sup>Cf. <https://stackoverflow.com/questions/18287848/unzip-in-one-pass>.