Cálculo de Programas

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

2020/21 - Ficha nr.° 10

1. 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:

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

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

3. Mostre que o anamorfismo que calcula os sufixos de uma lista

$$suffixes = [\!(g)\!]\ \textbf{where}\ g = (id + \langle \mathsf{cons}, \pi_2 \rangle) \cdot \mathsf{out}$$

é a função

$$
\begin{array}{l}
suffixes\ [\,] = [\,] \\
suffixes\ (h : t) = (h : t) : suffixes\ t
\end{array}
$$

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

$$\mathsf{mirror} = [\!((id + \mathsf{swap}) \cdot \mathsf{out})\!] \tag{F1}$$

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. Nas aulas teóricas viu-se que, sempre que um ciclo-*while* termina, ele pode ser definido por

$$\mathbf{while}\ p\ f\ g = \mathbf{tailr}\ ((g + f) \cdot (\neg \cdot p)?) \tag{F2}$$

recorrendo ao combinador de *"tail recursion"* $\mathbf{tailr}\ f = \llbracket \nabla, f \rrbracket$, que é um hilomorfismo de base $\mathsf{B}\ (X, Y) = X + Y$, para $\nabla = [id\ , id]$.

   (a) Derive a definição *pointwise* de $\mathbf{while}\ p\ f\ g$, sabendo que qualquer $h = \llbracket f, g \rrbracket$ é tal que $h = f \cdot \mathsf{F}\ h \cdot g$.

   (b) Complete a demonstração da lei de fusão de **tailr**[1]

   $$(\mathbf{tailr}\ g) \cdot f = \mathbf{tailr}\ h \ \Leftarrow\ (id + f) \cdot h = g \cdot f$$

   que se segue:

   $$\begin{array}{ll}
   & (\mathbf{tailr}\ g) \cdot f = \mathbf{tailr}\ h \\
   \equiv & \quad \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ \} \\
   & \llparenthesis\, \nabla \,\rrparenthesis \cdot [\!(g)\!] \cdot f = \llparenthesis\, \nabla \,\rrparenthesis \cdot [\!(h)\!] \\
   \Leftarrow & \quad \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ \} \\
   & [\!(g)\!] \cdot f = [\!(h)\!] \\
   \Leftarrow & \quad \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ \} \\
   & g \cdot f = (id + f) \cdot h \\
   \Box &
   \end{array}$$

6. O algoritmo de "quick-sort" foi definido nas aulas teóricas como o hilomorfismo $qSort = \llparenthesis\, inord \,\rrparenthesis \cdot [\!(qsep)\!]$ sobre árvores BTree, cujo catamorfismo recorre à função:

   $$inord = [\mathsf{nil}\ , f]\ \mathbf{where}\ f\ (x, (l, r)) = l +\!\!+ [x] +\!\!+ r$$

   Para um certo isomorfismo α, $h = \llparenthesis\, inord \cdot \alpha \,\rrparenthesis \cdot [\!(qsep)\!]$ ordenará a lista de entrada por ordem inversa. Identifique α, justificando informalmente.

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

   $$\begin{tikzcd}
   A^\infty & A \times A^\infty \arrow[l, "\mathsf{cons}"'] \\
   A \arrow[u, "\mathsf{repeat}"] \arrow[r, "\langle id, id \rangle"'] & A \times A \arrow[u, "id \times \mathsf{repeat}"']
   \end{tikzcd}$$

   é a função:

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

   De seguida, recorrendo às leis dos anamorfismos mostre que, apesar de não terminar[2], repeat satisfaz a propriedade:[3]

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

[1] **NB**: Assume-se que $(\mathbf{tailr}\ g) \cdot f$ termina.

[2] Por isso usamos, no diagrama, A^∞ em vez de A^*, para incluir também as listas infinitas.

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