Cálculo de Programas

Lic. C. Computação (2º ano)
Lic./Mest. em Engenharia Informática (3º ano)
UNIVERSIDADE DO MINHO

2021/22 - Ficha nr.° 10

1. Recorra às leis dos catamorfismos para demonstrar a propriedade natural

$$(\mathsf{LTree}\ f) \cdot \mathsf{mirror} \quad = \quad \mathsf{mirror} \cdot (\mathsf{LTree}\ f) \tag{F1}$$

onde mirror é o catamorfismo

$$\mathsf{mirror} :: \mathsf{LTree}\ a \rightarrow \mathsf{LTree}\ a$$
$$\mathsf{mirror} = (\!|\ \mathsf{in} \cdot (id + \mathsf{swap})\ |\!)$$

que "espelha"uma árvore e $\mathsf{LTree}\ f = (\!|\ \mathsf{in} \cdot (f + id)\ |\!)$ é o correspondente functor de tipo.

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:

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

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 o anamorfismo que calcula os sufixos de uma lista

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

é a função

$$\mathit{suffixes}\ [\,] = [\,]$$
$$\mathit{suffixes}\ (h : t) = (h : t) : \mathit{suffixes}\ t$$

5. 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.

6. O algorítmo "bubble-sort" é o ciclo-for

$$
\begin{array}{l}
bSort\ xs = \mathsf{for\ bubble}\ xs\ (\mathsf{length}\ xs)\ \mathbf{where} \\
\quad \mathsf{bubble}\ (x : y : xs) \\
\qquad \mid x > y = y : \mathsf{bubble}\ (x : xs) \\
\qquad \mid \mathsf{otherwise} = x : \mathsf{bubble}\ (y : xs) \\
\quad \mathsf{bubble}\ x = x
\end{array}
$$

cujo corpo de ciclo é um hilomorfismo $\mathsf{bubble} = [\![conquer, divide]\!]$. Identifique os genes $divide$ e $conquer$ desse hilomorfismo. **Sugestão**: siga a heurística que foi usada nas aulas teóricas para fazer o mesmo para a função de Fibonacci.[1]

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

$$\mathbf{while}\ p\ f\ g = \mathbf{tailr}\ ((g + f) \cdot (\neg \cdot p)?) \tag{F3}$$

recorrendo ao combinador de *"tail recursion"* $\mathbf{tailr}\ f = [\![\nabla, f]\!]$, 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 $h = [\![f, g]\!]$ é tal que $h = f \cdot \mathsf{F}\, h \cdot g$.

(b) Complete a demonstração da lei de fusão de **tailr**[2]

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

que se segue:

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

[1] Se não tiver vindo à aula teórica veja a mesma heurística a ser aplicada à função de Fibonacci no vídeo T9b, a começar em t=28:09.

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