Cálculo de Programas

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

2019/20 - Ficha nr.° 11

1. O facto de $\mathsf{length} : A^* \to \mathbb{N}_0$ poder ser definida tanto como *catamorfismo* de listas como *anamorfismo* de naturais (que foi assunto de uma questão de uma ficha anterior) pode generalizar-se da forma seguinte: sejam dados dois tipos indutivos

$$\begin{tikzcd} \mathsf{T}_1 \arrow[rr, bend left, "\mathsf{out}_1"] \arrow[rr, phantom, "\cong"] & & \mathsf{F}\ \mathsf{T}_1 \arrow[ll, bend left, "\mathsf{in}_1"] \end{tikzcd} \qquad \begin{tikzcd} \mathsf{T}_2 \arrow[rr, bend left, "\mathsf{out}_2"] \arrow[rr, phantom, "\cong"] & & \mathsf{G}\ \mathsf{T}_2 \arrow[ll, bend left, "\mathsf{in}_2"] \end{tikzcd}$$

e $\alpha : \mathsf{F}\ X \to \mathsf{G}\ X$, isto é, α satisfaz a propriedade *grátis*

$$\mathsf{G}\, f \cdot \alpha = \alpha \cdot \mathsf{F}\, f \qquad \text{(F1)}$$

Então $(\!|\, \mathsf{in}_2 \cdot \alpha \,|\!) = [\!(\alpha \cdot \mathsf{out}_1)\!]$, como se mostra a seguir (complete as justificações):

$$\begin{array}{ll} & k = (\!|\, \mathsf{in}_2 \cdot \alpha \,|\!) \\ \equiv & \quad \{ \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \quad \} \\ & k \cdot \mathsf{in}_1 = \mathsf{in}_2 \cdot \alpha \cdot \mathsf{F}\ k \\ \equiv & \quad \{ \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \quad \} \\ & \mathsf{out}_2 \cdot k = \mathsf{G}\ k \cdot \alpha \cdot \mathsf{out}_1 \\ \equiv & \quad \{ \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \quad \} \\ & k = [\!(\alpha \cdot \mathsf{out}_1)\!] \\ \Box & \end{array}$$

Identifique T_1, T_2 e α no caso de length.

2. 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[1], repeat satisfaz a propriedade:[2]

$$\mathsf{map}\ f \cdot \mathsf{repeat} = \mathsf{repeat} \cdot f \qquad \text{(F2)}$$

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

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

3. 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 $\mathbf{tailr}$[3]

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

   que se segue:

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

4. Um mónade é um functor T equipado com duas funções μ e u,

$$A \xrightarrow{\ u\ } \mathsf{T}\ A \xleftarrow{\ \mu\ } \mathsf{T}\ (\mathsf{T}\ A)$$

que satisfazem (para além das naturais, ie. "grátis") as propriedades $\mu \cdot u = id = \mu \cdot \mathsf{T}\ u$ e $\mu \cdot \mu = \mu \cdot \mathsf{T}\ \mu$ — identifique-as no formulário — com base nas quais se pode definir a *composição monádica*:

$$f \bullet g = \mu \cdot \mathsf{T}\,f \cdot g.$$

(Identifique-a também no formulário.) Demonstre os factos seguintes:

$$\mu \ = \ id \bullet id \tag{F4}$$

$$f \bullet u = f \ \wedge \ f = u \bullet f \tag{F5}$$

$$(f \cdot g) \bullet h \ = \ f \bullet (\mathsf{T}\ g \cdot h) \tag{F6}$$

$$\mathsf{T}\,f \ = \ (u \cdot f) \bullet id \tag{F7}$$

5. A função $discollect : (A \times B^*)^* \to (A \times B)^*$ que apareceu (sem ser definida) numa questão das primeiras fichas não é mais do que

$$discollect = lstr \bullet id \tag{F8}$$

— onde $lstr\ (a, x) = [\,(a, b) \mid b \leftarrow x\,]$ — no mónade das listas, $\mathsf{T}\ A = A^*$,

$$A \xrightarrow{\ singl\ } A^* \xleftarrow{\ \mathsf{concat}\ } (A^*)^*$$

onde $u = singl$ e $\mu = \mathsf{concat} = (\!|\, [\mathsf{nil}\ , \mathsf{conc}] \,|\!)$. Recordando a lei de absorção-cata (para listas), derive uma definição recursiva para $discollect$ que não use nenhum dos combinadores 'point-free' estudados nesta disciplina.

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