Cálculo de Programas

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

2016/17 - Ficha nr.° 12

1. Consultando as bibliotecas em Haskell disponíveis no material pedagógico, complete o seguinte quadro relativo aos tipos indutivos que aí se codificam:

T	**Descrição**	in	$\mathsf{B}\ (X, Y)$	$\mathsf{B}\ (f, g)$	$\mathsf{F}\ f$	$\mathsf{T}\ f$
$\mathbb{N}_0$	Números naturais					
A^*	Sequências finitas de A					
$\mathsf{BTree}\ A$	Árvores binárias de A					
$\mathsf{LTree}\ A$	Árvores com A nas folhas					

2. 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

$$\mu \cdot u = id = \mu \cdot \mathsf{T}\ u \tag{F1}$$
$$\mu \cdot \mu = \mu \cdot \mathsf{T}\ \mu \tag{F2}$$

com base nas quais se pode definir a *composição monádica* $f \bullet g = \mu \cdot \mathsf{T}\ f \cdot g$. Demonstre os factos seguintes:

$$\mu \quad = \quad id \bullet id \tag{F3}$$
$$f \bullet u = f \quad \wedge \quad f = u \bullet f \tag{F4}$$
$$(f \cdot g) \bullet h \quad = \quad f \bullet (\mathsf{T}\ g \cdot h) \tag{F5}$$
$$\mathsf{T}\ f \quad = \quad (u \cdot f) \bullet id \tag{F6}$$

3. O tipo $\mathsf{T}\ A = 1 + A$ foi o primeiro exemplo de mónade apresentado nesta disciplina, em que $\mu = [i_1\ , id]$ e $u = i_2$. Mostre que μ e u satisfazem as duas propriedades (F1) e (F2) que caracterizam um mónade, neste caso:

$$\mu \cdot u = id = \mu \cdot (id + u)$$
$$\mu \cdot \mu = \mu \cdot (id + \mu)$$

4. No mónade das listas tem-se $\mu = \mathsf{concat} = (\![\mathsf{nil}\ , \mathsf{conc}]\!)$ para $\mathsf{conc} = \widehat{(+\!\!+)}$. Logo a composição monádica $f \bullet g = \mu \cdot \mathsf{T}\ f \cdot g$ será, para o mónade das listas,

$$f \bullet g\ x = f'\ (g\ x)$$
$$\quad \mathbf{where}\ f' = \mathsf{concat} \cdot \mathsf{map}\ f$$

Com base na lei de absorção-cata, calcule a seguinte definição de f' com variáveis:

$$f'\ [\,] = [\,]$$
$$f'\ (h : t) = (f\ h) +\!\!+ f'\ t$$

5. Em Haskell, um mónade declara-se (ver a classe $Monad$) definindo a unidade u (que aí se designa por return) e uma operação $x \gg\!= f$, conhecida como aplicação monádica, ou "binding" de f a x, que é tal que

$$x \gg\!= f = (f \bullet id)\ x = (\mu \cdot \mathsf{T}\ f)\ x \qquad \text{(F7)}$$

Mostre que

$$x \gg\!= (f \bullet g) = (x \gg\!= g) \gg\!= f \qquad \text{(F8)}$$
$$\mu = (\gg\!= id) \qquad \text{(F9)}$$

6. O tipo

$$\mathbf{data}\ Err\ a = Err\ String \mid Ok\ a$$

que vamos querer usar para gerir a emissão de mensagens de erro em funções parciais, mostra-se facilmente ser um functor definindo

$$Err\ f = inE \cdot (id + f) \cdot outE \qquad \text{(F10)}$$

onde $inE = [Err\ , Ok]$ e

$$outE\ (Err\ s) = i_1\ s$$
$$outE\ (Ok\ a) = i_2\ a$$

(Verifique-o como trabalho de casa.) O tipo Err forma, ainda, um mónade desde que equipado com unidade $u = Ok$ e multiplicação

$$\mu :: Err\ (Err\ a) \rightarrow Err\ a$$
$$\mu\ (Err\ s) = Err\ s$$
$$\mu\ (Ok\ a) = a$$

Complete o cálculo que se segue mais abaixo da derivação do código acima a partir da sua definição *pointfree*

$$\begin{tikzcd} Err\ A & S + A \arrow[l, "inE"'] & S + (S + A) \arrow[l, "{[i_1\ , id]}"'] & Err\ (S + A) \arrow[l, "outE"'] & Err\ (Err\ A) \arrow[l, "(Err\ outE)"'] \arrow[llll, bend left, "\mu"] \end{tikzcd}$$

onde S abbrevia $String$:

$$\mu = inE \cdot [i_1\ , id] \cdot outE \cdot (Err\ outE)$$
$$= \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \}$$
$$\mu = inE \cdot [i_1\ , id] \cdot outE \cdot (inE \cdot (id + outE) \cdot outE)$$
$$= \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \}$$
$$\vdots$$
$$\equiv \quad \{ \ \ldots \text{alguns passos depois} \ldots \ \}$$
$$\vdots$$
$$= \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \}$$
$$\left\{ \begin{array}{l} \mu \cdot Err = Err \\ \mu \cdot Ok = id \end{array} \right.$$