Cálculo de Programas
Algebra of Programming

UNIVERSIDADE DO MINHO
Lic. em Engenharia Informática (3º ano)
Lic. Ciências da Computação (2º ano)

2022/23 - Ficha *(Exercise sheet)* nr. 11

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

   A monad is a functor T equipped with two functions μ and u

$$A \xrightarrow{u} \mathsf{T}\ A \xleftarrow{\mu} \mathsf{T}\ (\mathsf{T}\ A) \tag{F1}$$

   que satisfazem as propriedades

   satisfying

$$\mu \cdot u = id = \mu \cdot \mathsf{T}\ u \tag{F2}$$
$$\mu \cdot \mu = \mu \cdot \mathsf{T}\ \mu \tag{F3}$$

   para além das respectivas propriedades "grátis", onde $\mathsf{T}^2\ f$ abrevia $\mathsf{T}\ (\mathsf{T}\ f)$:

   in addition to their "free" properties, where $\mathsf{T}^2\ f$ abbreviates $\mathsf{T}\ (\mathsf{T}\ f)$:

$$\mathsf{T}\ f \cdot u = u \cdot f \tag{F4}$$
$$\mathsf{T}\ f \cdot \mu = \mu \cdot \mathsf{T}^2\ f \tag{F5}$$

   Partindo da definição

   Starting from the definition of monadic composition,

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

   de *composição monádica*, demonstre os factos seguintes:

   prove the following facts:

$$\mu = id \bullet id \tag{F6}$$
$$f \bullet u = f \quad \wedge \quad f = u \bullet f \tag{F7}$$
$$(f \cdot g) \bullet h = f \bullet (\mathsf{T}\ g \cdot h) \tag{F8}$$
$$\mathsf{T}\ f = (u \cdot f) \bullet id \tag{F9}$$

---

2. Recorde que o tipo

   Remember that the parametric type

$$\mathbf{data}\ \mathit{Maybe}\ a = \mathsf{Just}\ a \mid \mathsf{Nothing}$$

   forma um mónade cuja operação de multiplicação pode ser captada pelo diagrama seguinte, onde $\mathsf{in} = [\mathsf{Just}\ , \underline{\mathsf{Nothing}}]$:

   is a monad whose multiplication operation can be captured by the following diagram, where $\mathsf{in} = [\mathsf{Just}\ , \underline{\mathsf{Nothing}}]$:

$$
\begin{tikzcd}
Maybe\ (Maybe\ a) \arrow[d, "\mu"'] & (Maybe\ a) + 1 \arrow[l, "\mathsf{in}"'] \arrow[d, "id + !"] \\
Maybe\ a & (Maybe\ a) + 1 \arrow[l, "{[id\ ,\ \mathbf{in} \cdot i_2]}"']
\end{tikzcd}
\qquad \mu \cdot \mathsf{in} = [id\ , \mathsf{in} \cdot i_2] \cdot (id + !)
$$

Derive deste diagrama a definição *pointwise* dessa função:

Derive from this diagram the pointwise definition of μ:

$$
\begin{array}{l}
\mu\ (\mathsf{Just}\ a) = a \\
\mu\ \mathsf{Nothing} = \mathsf{Nothing}
\end{array}
$$

---

3. Considere a função

Consider

$$
\begin{array}{l}
discollect : (A \times B^*)^* \to (A \times B)^* \\
discollect = lstr \bullet id
\end{array}
$$

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

where $lstr\ \ (a, x)\ =\ [(a, b)\ \mid\ b\ \leftarrow\ x]$, in the list-monad, $\mathsf{T}\ A = A^*$:

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

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

Recalling $concat = (\![\mathsf{nil}\ , \mathsf{conc}]\!)$ and cata-absorption (for lists), derive a recursive definition for discollect that uses none of the 'point-free' combinators studied in this course.

---

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

In Haskell, monads are declared by instantiating, in the class Monad, the unit u (which there is called return*) and an operation $x \ggg\!= f$, known as* "binding" *of f to x, which is such that*

$$
x \ggg\!= f = (f \bullet id)\ x = (\mu \cdot \mathsf{T}\ f)\ x \tag{F10}
$$

Mostre que:

Prove:

$$
\mu = (\ggg\!= id) \tag{F11}
$$

$$
g \bullet f = (\ggg\!= g) \cdot f \tag{F12}
$$

$$
x \ggg\!= (f \bullet g) = (x \ggg\!= g) \ggg\!= f \tag{F13}
$$

---

5. Recordando o combinador $\mathsf{for}\ b\ i = (\![\underline{i}\ , b]\!)$, seja definido o ciclo

Remembering $\mathsf{for}\ b\ i = (\![\underline{i}\ , b]\!)$, define the for-loop

$$
k = (\![\underline{u\ i}\ , g \bullet id]\!) \tag{F14}
$$

onde $g : A \to \mathsf{T}\ A$ para um dado mónade T (F1). Faça um diagrama para k e mostre que k é a função

where $g : A \to \mathsf{T}\ A$ for a given monad T (F1). Draw a diagram for k and show that it is the function

$$k\ 0 = \mathsf{return}\ i$$
$$k\ (n+1) = \mathbf{do}\ \{x \leftarrow k\ n; g\ x\}$$

NB: para a unidade de um monad usam-se as notações return e u indistintamente.
Sugestão: recorra, para além das leis que conhece do cálculo de mónades, à definição:

***NB**: for the unit of a monad the notations* return *and u are used indistinctly.*
***Hint**: use, in addition to the monad laws you know, the definition:*

$$(f \bullet g)\ a \quad = \quad \mathbf{do}\ \{b \leftarrow g\ a; f\ b\} \tag{F15}$$

---

6. Suponha um tipo indutivo $\mathsf{T}\ X$ cuja base é o bifunctor

Let an inductive type $\mathsf{T}\ X$ be given whose base is the bifunctor

$$\mathsf{B}\ (X, Y) = X + \mathsf{G}\ Y$$
$$\mathsf{B}\ (f, g) = f + \mathsf{G}\ g$$

onde G é um outro qualquer functor. Mostre que $\mathsf{T}\ X$ é um mónade em que

where G is any other functor. Show that $\mathsf{T}\ X$ is a monad in which

$$\begin{cases} \mu = (\!|\, [id\ , \mathsf{in} \cdot i_2]\, |\!) \\ u = \mathsf{in} \cdot i_1 \end{cases} \tag{F16}$$

onde $\mathsf{in} : \mathsf{B}\ (X, \mathsf{T}\ X) \to \mathsf{T}\ X$.

where $\mathsf{in} : \mathsf{B}\ (X, \mathsf{T}\ X) \to \mathsf{T}\ X$.

---

7. (a) Alguns mónades conhecidos, por exemplo LTree, resultam de (F16). Identifique G em cada caso. (b) Para $\mathsf{G}\ Y = 1$ (i.é $\mathsf{G}\ f = id$) qual é o mónade que se obtém por (F16)? E no caso em que $\mathsf{G}\ Y = O \times Y^*$, onde o tipo O se considera fixo à partida?

(a) Some known monads, for example LTree*, result from (F16). Identify G in each case. (b) For $\mathsf{G}\ Y = 1$ (ie $\mathsf{G}\ f = id$) what is the monad obtained by (F16)? And in the case where $\mathsf{F}\ Y = O \times Y^*$, where the type O is considered fixed at the outset?*