**Cálculo de Programas**

3.º Ano de LEI+MiEI (Universidade do Minho)
Ano Lectivo de 2021/22

Exame de Recurso — 7 de Fevereiro 2022
14h00–16h00 - Salas E3-0.06, E3-0.07 e E3-0.08

---

- *Esta prova consta de **8** questões que valem, cada uma, 2.5 valores. O tempo médio estimado para resolução de cada questão é de 15 min.*
- *Recomenda-se que os alunos leiam a prova antes de decidirem por que ordem querem responder às questões que são colocadas.*

PROVA PRESENCIAL INDIVIDUAL SEM CONSULTA (2h)

**Questão 1** Considere o diagrama

$$
\begin{tikzcd}
(A \times B) \times C \arrow[rr, bend left, "\mathsf{assocr}"] \arrow[rr, phantom, "\cong"] & & A \times (B \times C) \arrow[ll, bend left, "\mathsf{assocl}"]
\end{tikzcd}
$$

onde $\mathsf{assocl} = \langle id \times \pi_1, \pi_2 \cdot \pi_2 \rangle$. Apresente justificações para o cálculo que se segue em que se resolve em ordem a $\mathsf{assocr}$ a equação $\mathsf{assocl} \cdot \mathsf{assocr} = id$:

$$
\begin{array}{ll}
& \mathsf{assocl} \cdot \mathsf{assocr} = id \\
\equiv & \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& \left\{ \begin{array}{l} (id \times \pi_1) \cdot \mathsf{assocr} = \pi_1 \\ \pi_2 \cdot \pi_2 \cdot \mathsf{assocr} = \pi_2 \end{array} \right. \\
\equiv & \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& \left\{ \begin{array}{l} \left\{ \begin{array}{l} \pi_1 \cdot \mathsf{assocr} = \pi_1 \cdot \pi_1 \\ \pi_1 \cdot \pi_2 \cdot \mathsf{assocr} = \pi_2 \cdot \pi_1 \end{array} \right. \\ \pi_2 \cdot \pi_2 \cdot \mathsf{assocr} = \pi_2 \end{array} \right. \\
\equiv & \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& \left\{ \begin{array}{l} \pi_1 \cdot \mathsf{assocr} = \pi_1 \cdot \pi_1 \\ \left\{ \begin{array}{l} \pi_1 \cdot \pi_2 \cdot \mathsf{assocr} = \pi_2 \cdot \pi_1 \\ \pi_2 \cdot \pi_2 \cdot \mathsf{assocr} = \pi_2 \end{array} \right. \end{array} \right. \\
\equiv & \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& \left\{ \begin{array}{l} \pi_1 \cdot \mathsf{assocr} = \pi_1 \cdot \pi_1 \\ \pi_2 \cdot \mathsf{assocr} = \langle \pi_2 \cdot \pi_1, \pi_2 \rangle \end{array} \right. \\
\equiv & \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& \mathsf{assocr} = \langle \pi_1 \cdot \pi_1, \pi_2 \times id \rangle \\
\Box &
\end{array}
$$

---

**Questão 2** Recorde a definição de guarda de um condicional de McCarthy:

$$\begin{tikzcd} A \arrow[r, "\langle p, id \rangle"] \arrow[rr, bend right, "p?"'] & 2 \times A \arrow[r, "\alpha"] & A + A \end{tikzcd} \quad \text{(E1)}$$

Mostre que

$$\nabla \cdot \alpha = \pi_2 \quad \text{(E2)}$$

onde $\nabla = [id\,, id]$, sabendo que $\alpha^{\circ} = [\langle \underline{\mathrm{TRUE}}, id \rangle\,, \langle \underline{\mathrm{FALSE}}, id \rangle]$.

---

**Questão 3** Use as leis da exponenciação para demonstrar a propriedade:

$$\overline{f \cdot \mathsf{ap}} \cdot \overline{h \cdot \mathsf{ap}} = \overline{f \cdot h \cdot \mathsf{ap}}$$

---

**Questão 4** O facto de haver tantos números pares como ímpares permite-nos pensar noutra forma de construir e manipular números naturais, nomeadamente usando — em vez da habitual álgebra $\mathsf{in} = [\mathsf{zero}\,, \mathsf{succ}]$ — a alternativa $\mathsf{in}^{\bullet}$ que se segue

$$\begin{array}{l} \mathsf{in}^{\bullet} : \mathbb{N}_0 \leftarrow 1 + (\mathbb{N}_0 + \mathbb{N}_0) \\ \mathsf{in}^{\bullet} = [\mathsf{zero}, [\mathsf{par}\,, \mathsf{impar}]] \ \mathbf{where} \\ \quad \mathsf{zero}\ \_ = 0 \\ \quad \mathsf{impar}\ n = 2 * n + 1 \\ \quad \mathsf{par}\ n = 2 * n + 2 \end{array}$$

cujo functor é $\mathsf{F}\,f = id + (f + f)$. Deduza a inversa $\mathsf{out}^{\bullet}$ de $\mathsf{in}^{\bullet}$ a partir da equação

$$\mathsf{out}^{\bullet} \cdot \mathsf{in}^{\bullet} = id \quad \text{(E3)}$$

deixando o resultado em *Haskell executável*.

---

**Questão 5** Recorde o tipo

$$\mathbf{data}\ \mathsf{FTree}\ a\ c = Unit\ c \mid Comp\ a\ (\mathsf{FTree}\ a\ c, \mathsf{FTree}\ a\ c)$$

$$inFTree = [Unit\,, \widehat{Comp}]$$

$$\mathsf{F}\,f = id + id \times f^2$$

e dois catamorfismos seus:

- $cf = (\!|\, [\underline{1}\,, \mathsf{add} \cdot \pi_2] \,|\!)$ — conta as folhas de uma FTree
- $cn = (\!|\, [\underline{0}, \mathsf{succ} \cdot \mathsf{add} \cdot \pi_2] \,|\!)$ — conta os nós de uma *Ftree*

Pretendendo fazer as duas contagens numa travessia só, alguém aplicou a lei de *banana-split*, tendo obtido a seguinte função:

$$\begin{array}{l} cfn\ (Unit\ c) = (1, 0) \\ cfn\ (Comp\ a\ (l, r)) = (cl + cr, al + ar + 1)\ \mathbf{where} \\ \quad (cl, al) = cfn\ l \\ \quad (cr, ar) = cfn\ r \end{array}$$

Mostre que, de facto,

$$cfn = \langle cf\,, cn \rangle \quad \text{(E4)}$$

por aplicação da referida lei.

**Questão 6** A função $depth = (\!|\, [\underline{1}\,, \mathsf{succ} \cdot \widehat{max}] \,|\!)$ calcula a profundidade de árvores de tipo LTree, cujo functor de base é $\mathsf{B}\ (X, Y) = X + Y^2$.

Mostre que a profundidade de uma árvore não é alterada quando aplica uma função $f$ a todas as suas folhas; isto é, use as leis dos catamorfismos para provar a propriedade:

$$depth \cdot \mathsf{LTree}\ f = depth \tag{E5}$$

---

**Questão 7** Recorde uma função bem conhecida do Prelude do Haskell:

$$\mathsf{zip}\ [\,]\ \_ = [\,]$$
$$\mathsf{zip}\ \_\ [\,] = [\,]$$
$$\mathsf{zip}\ (a : x)\ (b : y) = (a, b) : \mathsf{zip}\ x\ y$$

Mostre que $\widehat{\mathsf{zip}} = [\!(h)\!]$ identificando o gene $h$ do anamorfismo no diagrama seguinte,

$$\begin{tikzcd} A^* \times B^* \arrow[r, "h"] \arrow[d, "\widehat{\mathsf{zip}}"'] & \cdots \arrow[d, "id+id\times\widehat{\mathsf{zip}}"] \\ (A \times B)^* & \cdots \arrow[l, "\mathsf{in}"] \end{tikzcd}$$

e preenchendo as reticências. Justifique a sua resposta.

---

**Questão 8** Pretende-se mostrar que o tipo

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

forma um mónade

$$X \xrightarrow{\mathsf{Just}} Maybe\ X \xleftarrow{\mu} Maybe\ (Maybe\ X)$$

cuja operação de multiplicação é descrita pelo diagrama seguinte:

$$\begin{tikzcd} Maybe\ (Maybe\ a) \arrow[d, "\mu"'] & (Maybe\ a) + 1 \arrow[l, "\mathsf{in}"'] \arrow[d, "id+!"] \\ Maybe\ a & (Maybe\ a) + 1 \arrow[l, "{[id\,,\,\mathbf{in}\cdot i_2]}"'] \end{tikzcd} \qquad \mu \cdot \mathsf{in} = [id\,, \mathsf{in} \cdot i_2] \cdot (id + !) \tag{E6}$$

onde $\mathsf{in} = [\mathsf{Just}\,, \underline{\mathsf{Nothing}}]$. Use (E6) para demonstrar a propriedade $\mu \cdot \mathsf{Just} = id$. (NB: não assuma que $Maybe$ é um monade – a prova pedida faz parte do exercício de mostrar que o é.)

---