Cálculo de Programas

2.º Ano de LEI+MiEI (Universidade do Minho)
Ano Lectivo de 2023/24

2º Teste — 11 de Dezembro de 2023, 17h00–19h00
Salas 1.03 + 1.01 + 0.20 + 1.13 do Edifício 2.

PROVA PRESENCIAL INDIVIDUAL SEM CONSULTA (2h)

Importante — *Ler antes de iniciar a prova:*

- *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.*

Questão 1 Os tipos $\mathbb{B}$ e $1+1$ — onde $1 = \{()\}$ — são isomorfos, podendo a função out $: \mathbb{B} \to 1+1$ ser escrita em Haskell da seguinte maneira:

$$\begin{array}{l} \mathsf{out} : \mathbb{B} \to 1+1 \\ \mathsf{out}\ \mathrm{FALSE} = i_1\ () \\ \mathsf{out}\ \mathrm{TRUE} = i_2\ () \end{array}$$

Apresente uma definição, sem recorrer a variáveis, para a função in (inversa de out), derivada por cálculo analítico a partir da definição dada acima de out.

RESOLUÇÃO: Tem-se (justificar os passos que são dados):

$$\begin{array}{ll}
& \left\{\begin{array}{l} \mathsf{out}\ \mathrm{FALSE} = i_1\ () \\ \mathsf{out}\ \mathrm{TRUE} = i_2\ () \end{array}\right. \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& \left\{\begin{array}{l} \mathsf{out}\ (\underline{\mathrm{FALSE}}\ ()) = i_1\ () \\ \mathsf{out}\ (\underline{\mathrm{TRUE}}\ ()) = i_2\ () \end{array}\right. \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& \left\{\begin{array}{l} \mathsf{out} \cdot \underline{\mathrm{FALSE}} = i_1 \\ \mathsf{out} \cdot \underline{\mathrm{TRUE}} = i_2 \end{array}\right. \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& \mathsf{out} \cdot [\underline{\mathrm{FALSE}}, \underline{\mathrm{TRUE}}] = [i_1, i_2] \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& [\underline{\mathrm{FALSE}}, \underline{\mathrm{TRUE}}] = \mathsf{in}
\end{array}$$

$\square$

Questão 2 Considere a função

$$\alpha\ p = \mathsf{swap} \cdot (p \to \pi_1\ ,\ \pi_2)$$

Determine o tipo mais geral de $\alpha\ p$ e, a partir dele, a sua propriedade grátis.

RESOLUÇÃO: Tem-se que:

- Em $p \to f, g$ os tipos de f e de g são iguais.
- Partindo de $\pi_1 : A \times B \to A$ e $\pi_2 : C \times D \to D$ ter-se-á $A = B = C = D$ e portanto $p \to \pi_1\ ,\ \pi_2 : A \times A \to A$.
- Como $\mathsf{swap} : X \times Y \to Y \times X$, $A := X \times Y$, ficamos com $\alpha\ p : (X \times Y) \times (X \times Y) \to Y \times X$.

Logo a lei grátis é (fazer diagrama, etc):

$$(g \times f) \cdot (\alpha\ p) = (\alpha\ p) \cdot ((f \times g) \times (f \times g))$$

$\square$

Questão 3 Considere a estrutura de dados que se segue, estudada nas aulas:

Árvores com informação de tipo A nos nós :

$$\mathsf{T} = \mathsf{BTree}\ A \qquad \begin{cases} \mathsf{B}\ (X, Y) = 1 + X \times Y^2 \\ \mathsf{B}\ (g, f) = id + g \times f^2 \end{cases} \qquad \mathsf{in} = [\underline{Empty}\ , Node]$$

Haskell: **data** $\mathsf{BTree}\ a = Empty \mid Node\ (a, (\mathsf{BTree}\ a, \mathsf{BTree}\ a))$

Defina como um catamorfismo a função

$$f : \mathsf{BTree}\ A \to A^*$$
$$f = (\!|\, g \,|\!)$$

isto é, identifique g tal que $f\ t$ seja o caminho a percorrer na árvore t para atingir o seu nó terminal mais à direita.

RESOLUÇÃO: $f = (\!|\, [\mathsf{nil}\ , g_2] \,|\!)$ onde $g_2 = \mathsf{cons} \cdot (id \times \pi_2)$, isto é, $g_2\ (a, (b, c)) = a : c$.

O que g_2 faz é, quanto já se tem em c o caminho a percorrer para atingir o nó mais à direita, acrescentar a (o nó onde se está de momento) a c. $\square$

Questão 4 A função nr (=*"no repeats"*) que se segue testa se uma lista tem elementos repetidos:

$$nr : A^* \to \mathbb{B}$$
$$nr = \pi_2 \cdot aux \ \textbf{where}$$
$$aux = (\!|\, [m\ , \langle n, h \rangle] \,|\!)$$
$$m\ \_ = ([\,], \text{TRUE})$$
$$n = \mathsf{cons} \cdot (id \times \pi_1)$$
$$h\ (a, (t, b)) = \neg\ (a \in t) \wedge b$$

Resolva em ordem a f e g a equação

$$\langle f, g \rangle = aux \tag{E1}$$

entregando essas funções definidas sem recurso a quaisquer combinadores pointfree estudados na disciplina. **Sugestão**: use a lei de recursividade mútua.

RESOLUÇÃO: Tem-se (justificar os passos que são dados):

$$\langle f, g\rangle = (\![\, [m, \langle n, h\rangle] \,]\!)$$

$$\{ \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \quad \}$$

$$\langle f, g\rangle = (\![\, [\langle \mathsf{nil}, \underline{\textsc{True}}\rangle, \langle n, h\rangle] \,]\!)$$

$$\{ \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \quad \}$$

$$\langle f, g\rangle = (\![\, \langle [\mathsf{nil}, n], [\underline{\textsc{True}}, h]\rangle \,]\!)$$

$$\{ \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \quad \}$$

$$\left\{ \begin{array}{l} f \cdot \mathsf{in} = [\mathsf{nil}, \mathsf{cons} \cdot (id \times \pi_2)] \cdot (id + id \times \langle f, g\rangle) \\ g \cdot \mathsf{in} = [\underline{\textsc{True}}, h] \cdot (id + id \times \langle f, g\rangle) \end{array} \right.$$

$$\{ \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \quad \}$$

$$\left\{ \begin{array}{l} f \cdot \mathsf{in} = \mathsf{in} \cdot (id + id \times f) \\ g \cdot \mathsf{in} = [\underline{\textsc{True}}, h \cdot \langle f, g\rangle] \end{array} \right.$$

$$\{ \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \quad \}$$

$$\left\{ \begin{array}{l} f = (\![\, \mathsf{in} \,]\!) \\ \left\{ \begin{array}{l} g \cdot \mathsf{nil} = \underline{\textsc{True}} \\ g \cdot \mathsf{cons} = h \cdot \langle f, g\rangle \end{array} \right. \end{array} \right.$$

$$\{ \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \quad \}$$

$$\left\{ \begin{array}{l} f = id \\ \left\{ \begin{array}{l} g\ [\,] = \textsc{True} \\ g\ (a : t) = h\ (f\ a, g\ t) \end{array} \right. \end{array} \right.$$

$$\{ \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \quad \}$$

$$\left\{ \begin{array}{l} f = id \\ \left\{ \begin{array}{l} g\ [\,] = \textsc{True} \\ g\ (a : t) = h\ (a, g\ t) \end{array} \right. \end{array} \right.$$

$$\{ \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \quad \}$$

$$\left\{ \begin{array}{l} f = id \\ \left\{ \begin{array}{l} g\ [\,] = \textsc{True} \\ g\ (a : t) = \neg\ (a \in t) \wedge g\ t \end{array} \right. \end{array} \right.$$

□

Questão 5 Nas aulas teórico-práticas demonstrou-se o seguinte resultado sobre a composição de catamorfismos:

$$(\![\, g \,]\!) \cdot (\![\, \mathsf{in} \cdot k \,]\!) = (\![\, g \cdot m \,]\!) \quad \Leftarrow \quad m \cdot \mathsf{F}\ f = \mathsf{F}\ f \cdot k \tag{E2}$$

Use (E2) para provar a lei de absorção-cata:

$$(\![\, g \,]\!) \cdot \mathsf{T}\ h = (\![\, g \cdot \mathsf{B}\ (h, id) \,]\!)$$

NB: recordam-se as leis functoriais estendidas a bifunctores:

$$\mathsf{B}\ (id, id) = id \tag{E3}$$

$$\mathsf{B}\ (h \cdot f, k \cdot g) = \mathsf{B}\ (h, k) \cdot \mathsf{B}\ (f, g) \tag{E4}$$

RESOLUÇÃO: Tem-se (justificar os passos que são dados):

$$
\begin{array}{ll}
& (\!| g |\!) \cdot \mathsf{T}\ h = (\!| g \cdot \mathsf{B}\ (h, id) |\!) \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& (\!| g |\!) \cdot (\!| \mathsf{in} \cdot \mathsf{B}\ (h, id) |\!) = (\!| g \cdot \mathsf{B}\ (h, id) |\!) \\
\Leftarrow & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& \mathsf{B}\ (h, id) \cdot \mathsf{F}\ f = \mathsf{F}\ f \cdot \mathsf{B}\ (h, id) \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& \mathsf{B}\ (h, id) \cdot \mathsf{B}\ (id, f) = \mathsf{B}\ (id, f) \cdot \mathsf{B}\ (h, id) \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& \mathsf{B}\ (h, f) = \mathsf{B}\ (h, f) \\
\Box &
\end{array}
$$

$\Box$

Questão 6 Considere a função:

$$
\begin{array}{l}
\mathit{stake}\ \_\ [\,] = [\,] \\
\mathit{stake}\ 0\ \_ = [\,] \\
\mathit{stake}\ y\ (x : t) \\
\quad \mid y \equiv x = [x] \\
\quad \mid y < x = [y] \\
\quad \mid y > x = x : \mathit{stake}\ (y - x)\ t
\end{array}
$$

$\mathit{stake}\ x\ y$ dá o maior prefixo de y cuja soma não excede x, truncando se necessário o último elemento. Por exemplo, $\mathit{stake}\ 5\ [1, 2] = [1, 2]$ e $\mathit{stake}\ 5\ [-1, 2, 6] = [-1, 2, 4]$.

Defina divide tal que $\widehat{\mathit{stake}} = [\!(\mathit{divide})\!]$ seja um anamorfismo de listas. Apoie a sua resolução num diagrama.

RESOLUÇÃO: Seja $\mathit{untake} = \widehat{\mathit{stake}}$. Há a necessidade de fazer com que

$$
\begin{array}{l}
\mid y \equiv x = [x] \\
\mid y < x = [y]
\end{array}
$$

não sejam mais um caso de base mas sim um caso particular do recursivo, possível pois $[x] = x : [\,] = x : \mathit{stake}\ 0\ \_$, a mesma coisa para $[y]$. Assim:

$$
\begin{array}{l}
\mathit{ustake}\ (\_, [\,]) = [\,] \\
\mathit{ustake}\ (0, x) = [\,] \\
\mathit{ustake}\ (y, (x : t)) \\
\quad \mid y \equiv x = x : \mathit{ustake}\ (0, t) \\
\quad \mid y < x = y : \mathit{ustake}\ (0, t) \\
\quad \mid y > x = x : \mathit{ustake}\ ((y - x), t)
\end{array}
$$

Logo:

$$
\begin{array}{l}
\mathit{divide}\ (\_, [\,]) = i_1\ () \\
\mathit{divide}\ (0, x) = i_1\ ()
\end{array}
$$

$$
\begin{array}{l}
divide\ (y,(x:t)) \\
\quad \mid y \equiv x = i_2\ (x,(0,t)) \\
\quad \mid y < x = i_2\ (y,(0,t)) \\
\quad \mid y > x = i_2\ (x,((y-x),t))
\end{array}
$$

□

Questão 7 O desenho que se segue descreve a estratégia de programação conhecida por *divide & conquer*:

$$
\begin{tikzcd}[column sep=huge]
\boxed{\text{Problema}} \arrow[r, "divide\ \text{(análise)}"] \arrow[d, "solve\ \text{(problema)}"'] & \boxed{\text{Sub-problemas}} \arrow[d, "solve\ \text{(sub-problemas)}"] \\
\boxed{\text{Solução}} & \boxed{\text{Sub-soluções}} \arrow[l, "conquer\ \text{(síntese)}"]
\end{tikzcd}
$$

No Cálculo de Programas, esta estratégia é captada pelo conceito de *hilomorfismo*, definido como a composição

$$solve = (\!|\, conquer \,|\!) \cdot [\!(\, divide \,)\!] \tag{E5}$$

que se pode demonstrar ser tal que:

$$solve = conquer \cdot (\mathsf{F}\, solve) \cdot divide \qquad \begin{tikzcd} A \arrow[r, "divide"] \arrow[d, "solve"'] & \mathsf{F} A \arrow[d, "\mathsf{F} solve"] \\ B & \mathsf{F} B \arrow[l, "conquer"] \end{tikzcd} \tag{E6}$$

Complete o raciocínio que se segue em que se converte (E5) em (E6):

$$
\begin{array}{cl}
 & solve = (\!|\, conquer \,|\!) \cdot [\!(\, divide \,)\!] \\
\equiv & \{\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ \} \\
 & solve = conquer \cdot \mathsf{F}\, (\!|\, conquer \,|\!) \cdot \mathsf{out} \cdot [\!(\, divide \,)\!] \\
\equiv & \{\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ \} \\
 & \vdots \\
\equiv & \{\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ \} \\
 & solve = conquer \cdot \mathsf{F}\, solve \cdot divide
\end{array}
$$

RESOLUÇÃO: Tem-se (justificar os passos que são dados):

$$
\begin{array}{cl}
 & solve = (\!|\, conquer \,|\!) \cdot [\!(\, divide \,)\!] \\
\equiv & \{\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ \} \\
 & solve = conquer \cdot \mathsf{F}\ (\!|\, conquer \,|\!) \cdot \mathsf{out} \cdot [\!(\, divide \,)\!] \\
\equiv & \{\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ \} \\
 & solve = conquer \cdot \mathsf{F}\ (\!|\, conquer \,|\!) \cdot \mathsf{F}\ [\!(\, divide \,)\!] \cdot divide \\
\equiv & \{\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ \} \\
 & solve = conquer \cdot \mathsf{F}\ ((\!|\, conquer \,|\!) \cdot [\!(\, divide \,)\!]) \cdot divide \\
\equiv & \{\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ \} \\
 & solve = conquer \cdot \mathsf{F}\, solve \cdot divide
\end{array}
$$

□

Questão 8 Em qualquer monad T faz sentido definir a operação que emparelha cada elemento b de um seu habitante t com um determinado valor a:

$$str\ a\ t \quad = \quad do\ \{\ b\ \leftarrow t\ ;\ return(a,b)\ \} \tag{E7}$$

Mostre que (E7) e a definição *pointfree* (E8) que se segue coincidem:

$$str\ a \quad = \quad \mathsf{T}\ \langle \underline{a}, id \rangle \tag{E8}$$

Sugestão: recorde, das aulas práticas, o facto $\mathsf{T}\ f = (u \cdot f) \bullet id$.

RESOLUÇÃO: Tem-se (justificar os passos que são dados):

$$
\begin{array}{ll}
& str\ a = \mathsf{T}\ \langle \underline{a}, id \rangle \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& str\ a = (u \cdot \langle \underline{a}, id \rangle) \bullet id \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& str\ a\ t = ((u \cdot \langle \underline{a}, id \rangle) \bullet id)\ t \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& str\ a\ t = \mathbf{do}\ \{ b \leftarrow id\ t; (\mathsf{return} \cdot \langle \underline{a}, id \rangle)\ b \} \\
\equiv & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
& str\ a\ t = \mathbf{do}\ \{ b \leftarrow t; \mathsf{return}\ (a, b) \}
\end{array}
$$

$\square$