Cálculo de Programas

2.º Ano de LCC+MiEI (Universidade do Minho)
Ano Lectivo de 2018/19

Teste — 30 de Maio de 2019
16h00–18h00
Cantina + Edif. 2 - 0.11/0.20/1.03/1.07

---

*Este teste consta de **8** questões que valem, cada uma, 2.5 valores. O tempo médio estimado para resolução de cada questão é de 15 min. Os alunos devem ler a prova antes de decidirem por que ordem responder às questões colocadas.*

PROVA SEM CONSULTA (2h)

Questão 1 Use a lei da troca para exprimir o isomorfismo $\mathsf{undistl} = [i_1 \times id, i_2 \times id]$ sob a forma alternativa seguinte: $\mathsf{undistl} = \langle \pi_1 + \pi_1, [\pi_2 , \pi_2] \rangle$.

RESOLUÇÃO: Tem-se:

$$
\begin{array}{cl}
 & \mathsf{undistl} = [i_1 \times id, i_2 \times id] \\
\equiv & \qquad \{ \ f \times g = \langle f \cdot \pi_1, g \cdot \pi_2 \rangle \text{ ; identidade } \} \\
 & \mathsf{undistl} = [\langle i_1 \cdot \pi_1, \pi_2 \rangle , \langle i_2 \cdot \pi_1, \pi_2 \rangle] \\
\equiv & \qquad \{ \text{ lei da troca } \} \\
 & \mathsf{undistl} = \langle [i_1 \cdot \pi_1 , i_2 \cdot \pi_1], [\pi_2 , \pi_2] \rangle \\
\equiv & \qquad \{ \ f + g = [i_1 \cdot f , i_2 \cdot g] \ \} \\
 & \mathsf{undistl} = \langle \pi_1 + \pi_1, [\pi_2 , \pi_2] \rangle
\end{array}
$$

$\square$

Questão 2 Considere a função

$$\alpha = (id + !) \cdot \mathsf{distl}. \tag{E1}$$

Qual é o seu tipo genérico — de $A \times B + 1$ para $(A + 1) \times B$ ou de $(A + 1) \times B$ para $A \times B + 1$? Justifique desenhando o diagrama respectivo. Seguidamente, derive (usando também um diagrama) a propriedade grátis de α.

RESOLUÇÃO: A função vai necessariamente para uma soma, cf. $id + !$. Logo estamos na situação do diagrama:

$$
\begin{tikzcd}
A \times B + 1 & A \times B + 1 \times B \arrow[l, "id+!"'] & (A + 1) \times B \arrow[l, "\mathsf{distl}"'] \arrow[ll, bend left, "\alpha"]
\end{tikzcd}
$$

A propriedade grátis

$$(f \times g + id) \cdot \alpha = \alpha \cdot ((f + id) \times g)$$

deduz-se directamente do tipo $\alpha : A \times B + 1 \leftarrow (A+1) \times B$, usando o diagrama

$$\begin{tikzcd}
A \times B + 1 \arrow[d, "f \times g + id"'] & (A+1) \times B \arrow[l, "\alpha"'] \arrow[d, "(f+id) \times g"] \\
A' \times B' + 1 & (A'+1) \times B' \arrow[l, "\alpha"]
\end{tikzcd}$$

tal como foi ensinado nas aulas. (NB: não se pede a prova analítica.)
□

Questão 3 Infira o tipo mais geral do parâmetro f da função condicional α que a seguir se define:

$$\alpha\ f = \pi_1 \cdot \pi_2 \rightarrow \mathsf{swap}\ ,\ f \tag{E2}$$

Justifique a sua resposta.

RESOLUÇÃO: O ponto de partida é a definição de condicional de McCarthy: $p \rightarrow f, g = [f\ , g] \cdot p?$. Temos assim que determinar X e Y em:

$$\begin{tikzcd}
 & X \arrow[d, "(\pi_1 \cdot \pi_2)?"'] & \\
X \arrow[r, "i_1"] \arrow[dr, "\mathsf{swap}"'] & X + X \arrow[d, dashed, "{[\mathsf{swap}, f]}"] & X \arrow[l, "i_2"'] \arrow[dl, "f"] \\
 & Y &
\end{tikzcd}$$

O predicado $\pi_1 \cdot \pi_2$ determina o tipo $X = A \times (\mathbb{B} \times C)$:

$$\begin{tikzcd}
 & A \times (\mathbb{B} \times C) \arrow[d, "(\pi_1 \cdot \pi_2)?"'] & \\
A \times (\mathbb{B} \times C) \arrow[r, "i_1"] \arrow[dr, "\mathsf{swap}"'] & A \times (\mathbb{B} \times C) + A \times (\mathbb{B} \times C) \arrow[d, dashed, "{[\mathsf{swap}, f]}"] & A \times (\mathbb{B} \times C) \arrow[l, "i_2"'] \arrow[dl, "f"] \\
 & Y &
\end{tikzcd}$$

Finalmente, swap determina $Y = (\mathbb{B} \times C) \times A$. Logo f terá tipo $(\mathbb{B} \times C) \times A \leftarrow A \times (\mathbb{B} \times C)$.
□

Questão 4 A função $\mathsf{length} = (\![\underline{0}, \mathsf{succ} \cdot \pi_2]\!)$ conta o número de elementos de uma lista. Se a lista tiver pelo menos um elemento a à cabeça, basta contar os elementos da cauda começando em 1 e vez de 0:

$$\mathsf{length} \cdot (a:) = (\![\underline{1}, \mathsf{succ} \cdot \pi_2]\!) \tag{E3}$$

Demonstre (E3) recorrendo à propriedade de fusão-cata, sabendo-se que

$$\mathsf{length} \cdot (a:) = \mathsf{succ} \cdot \mathsf{length} \tag{E4}$$

se verifica.

RESOLUÇÃO: Tem-se:

$$
\begin{array}{ll}
& \mathsf{length} \cdot (a{:}) = (\![[\underline{1}\,, \mathsf{succ} \cdot \pi_2]]\!) \\
\equiv & \quad \{ \text{ por (E4) e } \mathsf{length} = (\![[\underline{0}\,, \mathsf{succ} \cdot \pi_2]]\!) \ \} \\
& \mathsf{succ} \cdot (\![[\underline{0}\,, \mathsf{succ} \cdot \pi_2]]\!) = (\![[\underline{1}\,, \mathsf{succ} \cdot \pi_2]]\!) \\
\Leftarrow & \quad \{ \text{ por fusão-cata } \} \\
& \mathsf{succ} \cdot [\underline{0}\,, \mathsf{succ} \cdot \pi_2] = [\underline{1}\,, \mathsf{succ} \cdot \pi_2] \cdot (id + id \times \mathsf{succ}) \\
\equiv & \quad \{ \text{ por fusão-+ e absorção-+, seguida de natural-}\pi_2 \ \} \\
& [\mathsf{succ} \cdot \underline{0}\,, \mathsf{succ} \cdot \mathsf{succ} \cdot \pi_2] = [\underline{1}\,, \mathsf{succ} \cdot \mathsf{succ} \cdot \pi_2] \\
\equiv & \quad \{ \text{ por Eq-+ e igualdade entre iguais } \} \\
& \mathsf{succ} \cdot \underline{0} = \underline{1} \\
\equiv & \quad \{ \text{ por } f \cdot \underline{k} = \underline{f\ k} \text{ e } \mathsf{succ}\ 0 = 1 \ \} \\
& \mathit{true} \\
\Box &
\end{array}
$$

$\Box$

Questão 5 O número de movimentos que solucionam o "puzzle" das Torres de Hanoi, com n discos, é dado por

$$k\ n = 2^n - 1$$

Mostre (recorrendo à lei de recursividade mútua) que uma forma de calcular k é

$$
\begin{array}{l}
k = \pi_1 \cdot g \ \textbf{where} \\
\quad g = \mathsf{for}\ \mathit{loop}\ (0,1) \\
\quad \mathit{loop}\ (k,e) = (k + e, 2 * e)
\end{array}
$$

sabendo que k satisfaz as equações

$$
\begin{array}{l}
k\ 0 = 0 \\
k\ (n+1) = 2^n + k\ n
\end{array}
$$

(como facilmente se demonstra) e que $2^n = \mathsf{for}\ (2*)\ 1\ n$.

RESOLUÇÃO: Note-se que $k = \pi_1 \cdot g$ corresponde a $g = \langle k, h \rangle$, para um h a deduzir. Por outro lado, tirando as variáveis a loop, obtém-se $\mathit{loop} = \langle \mathsf{add}, (2*) \cdot \pi_2 \rangle$. Daí:

$$
\begin{array}{ll}
& \langle k, h \rangle = \mathsf{for}\ \mathit{loop}\ (0,1) \\
\equiv & \quad \{ \ \mathsf{for}\ b\ i = (\![[\underline{i}, b]]\!);\ \mathit{loop} = \langle \mathsf{add}, (2*) \cdot \pi_2 \rangle \ \} \\
& \langle k, h \rangle = (\![[\underline{(0,1)}\,, \langle \mathsf{add}, (2*) \cdot \pi_2 \rangle]]\!) \\
\equiv & \quad \{ \ \langle \underline{a}, \underline{b} \rangle = \underline{(a,b)} \text{ e lei da troca } \} \\
& \langle k, h \rangle = (\![\langle [\underline{0}, \mathsf{add}], [\underline{1}, (2*) \cdot \pi_2] \rangle]\!) \\
\equiv & \quad \{ \text{ recursividade mútua } \} \\
& \left\{ \begin{array}{l} k \cdot \mathsf{in} = [\underline{0}, \mathsf{add}] \cdot (id + \langle k, h \rangle) \\ h \cdot \mathsf{in} = [\underline{1}, (2*) \cdot \pi_2] \cdot (id + \langle k, h \rangle) \end{array} \right. \\
\equiv & \quad \{ \ \pi_2 \cdot \langle k, h \rangle = h \text{ e } \mathsf{for}\ b\ i = (\![[\underline{i}, b]]\!) \ \}
\end{array}
$$

$$\left\{\begin{array}{l} k \cdot \mathsf{in} = [\underline{0}, \mathsf{add}] \cdot (id + \langle k, h \rangle) \\ h = \mathsf{for}\ 2 *\ 1 \end{array}\right.$$

$\equiv$ { $\mathsf{in} = [\underline{0}, \mathsf{succ}]$; absorção-+; introdução de variáveis na 2ª igualdade }

$$\left\{\begin{array}{l} k \cdot [\underline{0}, \mathsf{succ}] = [\underline{0}, \mathsf{add} \cdot \langle k, h \rangle] \\ h\ n = \underbrace{\mathsf{for}\ 2 *\ 1\ n}_{2^n} \end{array}\right.$$

$\equiv$ { introdução de variáveis na 1ª igualdade }

$$\left\{\begin{array}{l} \left\{\begin{array}{l} k\ 0 = 0 \\ k\ (n+1) = h\ n + k\ n \end{array}\right. \\ h\ n = \underbrace{\mathsf{for}\ 2 *\ 1\ n}_{2^n} \end{array}\right.$$

$\equiv$ { pois $h\ n = 2^n$ }

$$\left\{\begin{array}{l} k\ 0 = 0 \\ k\ (n+1) = 2^n + k\ n \end{array}\right.$$

$\equiv$ { pela hipótese do problema }

$true$

□

□

Questão 6 A função $\mathsf{filter}\ p$, que seleciona de uma lista apenas os elementos que satisfazem p, pela ordem com que aí aparecem, pode ser definida como se segue,

$$\mathsf{filter}\ p = \mathsf{concat} \cdot \mathsf{map}\ (p \rightarrow singl, \mathsf{nil})$$

onde $\mathsf{concat} = (\![\mathsf{nil}, \mathsf{conc}]\!)$ e $singl\ a = [a]$.

Demonstre que $\mathsf{filter}\ p$ pode ser definida como um catamorfismo.

RESOLUÇÃO: Estamos com listas, cujo functor de tipo é $\mathsf{T}\ f = \mathsf{map}\ f$ para o functor de base $\mathsf{B}\ (f, g) = id + f \times g$. Por outro lado, $\mathsf{concat} = (\![\mathsf{nil}, \mathsf{conc}]\!)$. Logo:

$$\mathsf{filter}\ p = \mathsf{concat} \cdot \mathsf{map}\ (p \rightarrow singl, \mathsf{nil})$$

$\equiv$ { catamorfismo concat; $\mathsf{T}\ f = \mathsf{map}\ f$ }

$$\mathsf{filter}\ p = (\![\mathsf{nil}, \mathsf{conc}]\!) \cdot \mathsf{T}\ (p \rightarrow singl, \mathsf{nil})$$

$\equiv$ { absorção-cata, para $\mathsf{B}\ (f, g) = id + f \times g$ }

$$\mathsf{filter}\ p = (\![\mathsf{nil}, \mathsf{conc}] \cdot (id + (p \rightarrow singl, \mathsf{nil}) \times id)]\!)$$

$\equiv$ { simplificação (absorção-+) }

$$\mathsf{filter}\ p = (\![\mathsf{nil}, \mathsf{conc} \cdot ((p \rightarrow singl, \mathsf{nil}) \times id)]\!)$$

□

Em suma, $\mathsf{filter}\ p$ é um catamorfismo de listas.

□

Questão 7 Considere a seguinte versão *tail-recursive* da função factorial:

$$
\begin{array}{l}
\mathsf{fact}\ n = aux\ (n, 1) \\
\quad \mathbf{where}\ aux\ (0, a) = a \\
\qquad aux\ (n+1, a) = aux\ (n, a * (n+1))
\end{array}
$$

Mostre que a função auxiliar aux é um hilomorfismo apresentando definições para as funções α, β, γ e $conquer$ do diagrama seguinte, onde A é um tipo numérico qualquer:

$$
\begin{tikzcd}
\mathbb{N}_0 \times A \arrow[r, "\alpha"] \arrow[rrr, bend left, "divide"] \arrow[d, "aux"'] & (1+\mathbb{N}_0)\times A \arrow[r, "\beta"] & 1\times A + \mathbb{N}_0 \times A \arrow[r, "\gamma"] & A + \mathbb{N}_0 \times A \arrow[d, "id+aux"] \\
A & & & A + A \arrow[lll, bend left, "conquer"]
\end{tikzcd}
$$

RESOLUÇÃO: Uma forma sintética de apresentar a resolução deste exercício é exprimi-lo na própria linguagem Haskell, após passagem pelo $lhs2tex$:

$$
\begin{array}{l}
aux = conquer \cdot (\mathsf{F}\ aux) \cdot divide \\
\quad \mathbf{where} \\
\qquad \mathsf{F}\ f = \mathsf{B}\ id\ f \\
\qquad \mathsf{B}\ x\ y = x + y \\
\qquad divide = \gamma \cdot \beta \cdot \alpha \\
\qquad conquer = [id, id] \\
\qquad \alpha = \mathsf{out}_{\mathbb{N}_0} \times id \\
\qquad \beta = \mathsf{distl} \\
\qquad \gamma = \pi_2 + \theta \\
\qquad \theta\ (n, a) = (n, a * (n+1))
\end{array}
$$

□

Questão 8 Na questão 8 da ficha 7 das aulas teorico-práticas desta disciplina abordou-se o functor

$$
\begin{array}{l}
\mathsf{T}\ X = X \times X \\
\mathsf{T}\ f = f \times f
\end{array}
$$

Este functor é um mónade que nos permite trabalhar com pares encarados como vectores (x, y) a duas dimensões. Por exemplo, neste mónade a expressão

$$\mathbf{do}\ \{x \leftarrow (2,3); y \leftarrow (4,5); \mathsf{return}\ (x+y)\}$$

dá $(6,8)$ como resultado — a soma dos vectores $(2,3)$ e $(4,5)$. Definindo

$$\mu = \pi_1 \times \pi_2 \tag{E5}$$

$$u = \langle id, id \rangle \tag{E6}$$

para este functor T, demonstre que μ e u satisfazem as propriedades (59) e (60) do formulário, essenciais à evidência de que

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

é, de facto, um mónade.

RESOLUÇÃO: Cálculo de $\mu \cdot \mathsf{T}\ \mu = \mu \cdot \mu$:

$$
\begin{array}{ll}
& \mu \cdot \mathsf{T}\ \mu \\
= & \quad \{\ \text{(E5) duas vezes ; } \mathsf{T}\ f = f \times f\ \} \\
& (\pi_1 \times \pi_2) \cdot ((\pi_1 \times \pi_2) \times (\pi_1 \times \pi_2)) \\
= & \quad \{\ \text{functor-}\times\text{ ; natural-}\pi_1\text{ ; natural-}\pi_2\ \} \\
& (\pi_1 \cdot \pi_1) \times (\pi_2 \cdot \pi_2) \\
= & \quad \{\ \text{functor-}\times\text{ ; (E5)}\ \} \\
& \mu \cdot \mu
\end{array}
$$

$\square$

Cálculo de $\mu \cdot u = id = \mu \cdot \mathsf{T}\ u$:

$$
\begin{array}{ll}
& \mu \cdot u = id = \mu \cdot \mathsf{T}\ u \\
\equiv & \quad \{\ \text{(E5) ; (E6)}\ \} \\
& (\pi_1 \times \pi_2) \cdot \langle id, id \rangle = id = (\pi_1 \times \pi_2) \cdot \mathsf{T}\ u \\
\equiv & \quad \{\ \text{absorção-}\times\text{ ; } \mathsf{T}\ f = f \times f\text{; functor-}\times\ \} \\
& \langle \pi_1, \pi_2 \rangle = id = (\pi_1 \cdot u) \times (\pi_2 \cdot u) \\
\equiv & \quad \{\ \text{(E6) ; cancelamento-}\times\ \} \\
& \langle \pi_1, \pi_2 \rangle = id = id \times id \\
\equiv & \quad \{\ \text{reflexão-}\times\text{ e functor-}\times\text{-id}\ \} \\
& true
\end{array}
$$

$\square$

$\square$

ANEXO — Catálogo de alguns tipos de dados estudados na disciplina.

1. Números naturais:

$$\mathsf{T} = \mathbb{N}_0 \qquad \left\{ \begin{array}{l} \mathsf{F}\ X = 1 + X \\ \mathsf{F}\ f = id + f \end{array} \right. \qquad \mathsf{in} = [\underline{0}, \mathsf{succ}] \tag{E7}$$

   Haskell: Int inclui $\mathbb{N}_0$.

2. Listas de elementos em A:

$$\mathsf{T} = A^* \qquad \left\{ \begin{array}{l} \mathsf{F}\ X = 1 + A \times X \\ \mathsf{F}\ f = id + id \times f \end{array} \right. \qquad \mathsf{in} = [\mathsf{nil}, \mathsf{cons}] \tag{E8}$$

   Haskell: $[\,a\,]$.

3. Árvores com informação de tipo A nos nós:

$$\mathsf{T} = \mathsf{BTree}\ A \qquad \left\{ \begin{array}{l} \mathsf{F}\ X = 1 + A \times X^2 \\ \mathsf{F}\ f = id + id \times f^2 \end{array} \right. \qquad \mathsf{in} = [\underline{Empty}, Node] \tag{E9}$$

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

4. Árvores com informação de tipo A nas folhas:

$$\mathsf{T} = \mathsf{LTree}\ A \qquad \left\{ \begin{array}{l} \mathsf{F}\ X = A + X^2 \\ \mathsf{F}\ f = id + f^2 \end{array} \right. \qquad \mathsf{in} = [Leaf, Fork] \tag{E10}$$

   Haskell: $\mathbf{data}\ \mathsf{LTree}\ a = Leaf\ a \mid Fork\ (\mathsf{LTree}\ a, \mathsf{LTree}\ a)$.

5. Árvores com informação nos nós e nas folhas:

$$\mathsf{T} = \mathsf{FTree}\ B\ A \qquad \left\{ \begin{array}{l} \mathsf{F}\ X = B + A \times X^2 \\ \mathsf{F}\ f = id + id \times f^2 \end{array} \right. \qquad \mathsf{in} = [Unit, Comp] \tag{E11}$$

   Haskell: $\mathbf{data}\ \mathsf{FTree}\ b\ a = Unit\ b \mid Comp\ (a, (\mathsf{FTree}\ b\ a, \mathsf{FTree}\ b\ a))$.