Cálculo de Programas

2.º Ano de MiEI+LCC (Universidade do Minho)
Ano Lectivo de 2016/17

Exame de Recurso — 30 de Junho de 2017
16h00–18h00
Cantina de Gualtar

---

*Este teste consta de **8** questões que valem, cada uma, 2.5 valores. O tempo médio estimado para resolução de cada questão é de 12 min.*

PROVA SEM CONSULTA (1h30m)

Questão 1 A função g é tal que a propriedade

$$(id \times \pi_2) \cdot \langle id \times \pi_2, id \times \pi_1 \rangle \cdot g = id \tag{E1}$$

se verifica. Determine o tipo mais geral de g sem calcular a sua definição, e formule a respectiva propriedade grátis.

Questão 2 Seja dado um predicado p e uma função f tal que

$$p \cdot f = p \tag{E2}$$

se verifica. Mostre que

$$(p \rightarrow id, f) \cdot (p \rightarrow f, id) = f$$

se verifica sabendo-se que, entre outras leis que conhece, se tem:

$$p \rightarrow (p \rightarrow a, b), (p \rightarrow c, d) \quad = \quad p \rightarrow a, d \tag{E3}$$

$$p \rightarrow a, a \quad = \quad a \tag{E4}$$

Questão 3 Considere a função

$$sum_9 [\,] = 0$$
$$sum_9 (h : t) = \{h + sum_9 \, t\}_9$$

que soma todos os números de uma lista de naturais "tirando os noves", onde a operação "noves fora" é definida por

$$\{n\}_9 = n \text{ `}mod\text{` } 9$$

e obedece à propriedade:

$$\{a + \{b\}_9\}_9 = \{a + b\}_9 \tag{E5}$$

Encare sum_9 como um catamorfismo de listas e que mostre, recorrendo às leis dos catamorfismos, que se pode implementar essa mesma função fazendo apenas um cálculo de resto de divisão por 9,

$$sum_9 = \{\_\}_9 \cdot \mathsf{sum} \tag{E6}$$

onde $\mathsf{sum} = (\![[\mathsf{zero}, \mathsf{add}]]\!)$ e zero, add são funções que conhece.

Questão 4 O combinador

$$\begin{array}{l} \mathit{flip} :: (a \to b \to c) \to b \to a \to c \\ \mathit{flip}\ f\ x\ y = f\ y\ x \end{array}$$

troca a ordem dos argumentos de uma função. É fácil de ver que *flip* é um isomorfismo de exponenciais:

$$\begin{array}{ccccccc} (C^B)^A & \cong & C^{A\times B} & \cong & C^{B\times A} & \cong & (C^A)^B \\ f & \mapsto & \widehat{f} & \mapsto & \widehat{f}\cdot \mathsf{swap} & \mapsto & \overline{\widehat{f}\cdot \mathsf{swap}} = \mathit{flip}\ f \end{array}$$

Apresente justificações para os passos seguintes do cálculo desse isomorfismo a partir da sua definição ao ponto (*pointwise*):

$$\begin{array}{ll} & \mathit{flip}\ f\ x\ y = f\ y\ x \\ \equiv & \quad \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\ & (\mathsf{ap}\cdot(\mathit{flip}\ f \times id))\ (x,y) = (\mathsf{ap}\cdot(f\times id))\ (y,x) \\ \equiv & \quad \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\ & \mathsf{ap}\cdot(\mathit{flip}\ f\times id) = \mathsf{ap}\cdot(f\times id)\cdot\mathsf{swap} \\ \equiv & \quad \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\ & \mathit{flip}\ f = \overline{\mathsf{ap}\cdot(f\times id)\cdot\mathsf{swap}} \\ \equiv & \quad \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\ & \mathit{flip}\ f = \overline{\mathsf{ap}\cdot(\overline{\widehat{f}}\times id)\cdot\mathsf{swap}} \\ \equiv & \quad \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\ & \mathit{flip}\ f = \overline{\widehat{f}\cdot\mathsf{swap}} \end{array}$$

Questão 5 A seguinte função

$$\begin{array}{l} \mathit{odds}\ 0 = [\,] \\ \mathit{odds}\ (n+1) = (2\ n+1) : \mathit{odds}\ n \end{array}$$

lista os n-primeiros ímpares por ordem decrescente. Mostre, recorrendo à lei de recursividade múltipla, que odds é a função

$$\mathit{odds} = \pi_2 \cdot \mathbf{for}\ \mathit{body}\ (1,[\,])\ \mathbf{where}\ \mathit{body}\ (i,x) = (i+2, i:x)$$

Questão 6 Desenhe o diagrama do seguinte anamorfismo de listas

$$\begin{array}{l} f : \mathsf{BTree}\ A \to A^* \\ f = [\!(\alpha\cdot\mathsf{out}_{\mathsf{BTree}})\!] \end{array} \tag{E7}$$

onde $\alpha = id + id\times\pi_1$, e mostre que f se pode também escrever como um catamorfismo:

$$f = (\!|\mathsf{in}_{\mathsf{List}}\cdot\alpha|\!) \tag{E8}$$

Sugestão: a propriedade grátis de α pode ser-lhe útil.

Questão 7 Considere o algoritmo de *insertion sort* tal como vem dado na biblioteca List.hs:

$$
\begin{aligned}
& iSort = (\![\mathsf{nil}\,,\mathsf{insert}]\!) \ \mathbf{where} \\
& \qquad \mathsf{insert}\ (x, [\,]) = [x] \\
& \qquad \mathsf{insert}\ (x, a : l) \\
& \qquad\quad \mid x < a = [x, a] +\!\!+\ l \\
& \qquad\quad \mid \mathsf{otherwise} = a : (\mathsf{insert}\ (x, l))
\end{aligned}
$$

A função auxiliar $\mathsf{insert} : A \times A^* \to A^*$ pode se construída como um hilomorfismo $\mathsf{insert} = [\![g, h]\!]$ onde

$$
\begin{aligned}
& h\ (x, [\,]) = i_1\ [x] \\
& h\ (x, a : l) \\
& \quad \mid x < a = i_1\ ([x, a] +\!\!+\ l) \\
& \quad \mid \mathsf{otherwise} = i_2\ (a, (x, l))
\end{aligned}
$$

Identifique o gene g e complete as reticências do seguinte diagrama desse hilomorfismo, evidenciando o respectivo functor de base:

$$
\begin{tikzcd}
A \times A^* \arrow[r, "h"] \arrow[d, "insert"'] & \ldots \arrow[d, "id+id\times\mathsf{insert}"] \\
A^* & \ldots \arrow[l, "g"]
\end{tikzcd}
$$

Justifique informalmente a sua resposta.

Questão 8 Recorde a função

$$
\begin{aligned}
& discollect : (A \times B^*)^* \to (A \times B)^* \\
& discollect\ [\,] = [\,] \\
& discollect\ ((a, x) : y) = [(a, b) \mid b \leftarrow x] +\!\!+\ discollect\ y)
\end{aligned}
$$

que foi assunto em fichas das aulas práticas desta disciplina. Sabendo que as listas formam um mónade, onde

$$\mu = \mathsf{concat} = (\![\mathsf{nil}\,, \mathsf{conc}]\!) \tag{E9}$$

e

$$\mathsf{conc}\ (x, y) = x +\!\!+\ y \tag{E10}$$

e recordando a lei de absorção-cata (para listas), mostre que a definição acima pode ser calculada a partir de

$$discollect = lstr \bullet id \tag{E11}$$

onde $lstr\ (a, x) = [(a, b) \mid b \leftarrow x]$.

ANEXO — Catálogo de tipos de dados estudados na disciplina.

1. Números naturais:

$$\mathsf{T} = \mathbb{N}_0 \qquad \left\{ \begin{array}{l} \mathsf{F}\ X = 1 + X \\ \mathsf{F}\ f = id + f \end{array} \right. \qquad \mathsf{in} = [\underline{0}, \mathsf{succ}] \tag{E12}$$

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

2. Listas de elementos em A:

$$\mathsf{T} = A^* \qquad \left\{ \begin{array}{l} \mathsf{F}\ X = 1 + A \times X \\ \mathsf{F}\ f = id + id \times f \end{array} \right. \qquad \mathsf{in} = [\mathsf{nil}, \mathsf{cons}] \tag{E13}$$

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

3. Árvores com informação de tipo A nos nós:

$$\mathsf{T} = \mathsf{BTree}\ A \qquad \left\{ \begin{array}{l} \mathsf{F}\ X = 1 + A \times X^2 \\ \mathsf{F}\ f = id + id \times f^2 \end{array} \right. \qquad \mathsf{in} = [\underline{Empty}, Node] \tag{E14}$$

   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:

$$\mathsf{T} = \mathsf{LTree}\ A \qquad \left\{ \begin{array}{l} \mathsf{F}\ X = A + X^2 \\ \mathsf{F}\ f = id + f^2 \end{array} \right. \qquad \mathsf{in} = [Leaf, Fork] \tag{E15}$$

   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:

$$\mathsf{T} = \mathsf{FTree}\ B\ A \qquad \left\{ \begin{array}{l} \mathsf{F}\ X = B + A \times X^2 \\ \mathsf{F}\ f = id + id \times f^2 \end{array} \right. \qquad \mathsf{in} = [Unit, Comp] \tag{E16}$$

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