Cálculo de Programas

2.º Ano de LCC+MiEI (Universidade do Minho)
Ano Lectivo de 2019/20

Exame de Recurso — 18 de Julho de 2020
10h00–13h00
Prova realizada on-line via BBC

- *Esta prova consta de **10** questões que valem, cada uma, 2.0 valores. O tempo médio estimado para resolução de cada questão é de 15 min.*
- *Ao submeterem o seu teste no BB os alunos estão a declarar implicitamente que **subscrevem** a seguinte declaração:*

  Tendo presente o Código de Conduta Ética da Universidade do Minho e o Regulamento Disciplinar dos Estudantes da Universidade do Minho (Despacho RT-80/2019) e tendo conhecimento das sanções aplicáveis a atos de infração disciplinar, declaro por minha honra que pautarei a minha conduta na resolução desta prova de avaliação pelos valores de ética e integridade académica vigentes na Universidade do Minho. Declaro que realizarei a prova autonomamente e recorrendo exclusivamente aos elementos de consulta autorizada. Confirmo ainda que não incorrerei em qualquer ato de desonestidade, nomeadamente, os que violam os princípios éticos inerentes a processos de avaliação, como a prática de plágio ou qualquer outra forma indevida de utilização de informações. Mais declaro que não me envolverei em situações de prestação de informação ou apoio indevidos no decurso das provas que venha a realizar.
- *Os alunos devem escrever o seu **número mecanográfico** nas folhas cujas imagens vão submeter electronicamente.*

PROVA COM CONSULTA (3h)

Questão 1 Considere a função $\alpha = \langle \mathsf{coswap} \cdot \pi_1, \mathsf{swap} \cdot \pi_2 \rangle$. Infira o tipo mais geral de α e deduza a respectiva propriedade grátis, que deverá verificar **analiticamente**.

Questão 2 Complete a demonstração da seguinte propriedade de uma guarda p

$$(p \cdot \pi_1)? = \mathsf{distl} \cdot (p? \times id) \tag{E1}$$

feita tal como se segue:

$$
\begin{array}{ll}
& \mathsf{distl} \cdot (p? \times id) = (p \cdot \pi_1)? \\
\equiv & \qquad \{ \boxed{1} \} \\
& p? \times id = \boxed{2} \cdot (p \cdot \pi_1)? \\
\equiv & \qquad \{ \boxed{3} \} \\
& (p \rightarrow i_1\ ,\ i_2) \times id = p \cdot \pi_1 \rightarrow i_1 \times id\ ,\ i_2 \times id \\
\equiv & \qquad \{ \boxed{4} \} \\
& \boxed{5} \\
\equiv & \qquad \{ \boxed{6} \} \\
& true \\
\square &
\end{array}
$$

Questão 3 Considere a função $\mathsf{mirror} = (\!|\, \mathsf{in} \cdot (id + \mathsf{swap}) \,|\!)$ que espelha uma árvore de tipo LTree e as duas funções seguintes:

- $lm = (\!|\, [id\, , \pi_1] \,|\!)$ — vai buscar a folha mais à esquerda (lm abrevia 'leftmost')
- $rm = (\!|\, [id\, , \pi_2] \,|\!)$ — vai buscar a folha mais à direita (rm abrevia 'rightmost')

Mostre, por fusão-cata (em LTree) que:

$$rm \cdot \mathsf{mirror} = lm \tag{E2}$$

Questão 4 Suponha que tem um catamorfismo $(\!|\, \alpha \,|\!)$ em que $\alpha : \mathsf{F}\ A \to A$ é um isomorfismo, isto é, $\alpha^\circ : A \to \mathsf{F}\ A$ é um isomorfismo também. Mostre, por fusão-cata, que

$$[\!(\alpha^\circ)\!] \cdot (\!|\, \alpha \,|\!) = id. \tag{E3}$$

se verifica.

Questão 5 Considere a seguinte série definida por recorrência da seguinte forma:

$$s_0 = 1$$
$$s_{n+1} = 2 * s_n + n + 2$$

Assim, a lista $[1, 4, 11, 26, 57, 120, 247, 502, \ldots]$ mostra os primeiros termos da série. Mostre, por aplicação da lei de recursividade mútua, que a seguinte função

$$s = \pi_1 \cdot \mathsf{for}\ loop\ \mathsf{init}\ \mathbf{where}$$
$$\quad loop\ (x, y) = (2 * x + y, y + 1)$$
$$\quad \mathsf{init} = (1, 2)$$

calcula o n-ésimo termo da série.

Questão 6 Considere o isomorfismo

$$\alpha = [\langle \mathsf{nil}, id \rangle\, , (\mathsf{cons} \times id) \cdot \mathsf{assocl}] \tag{E4}$$

onde, como sabe,

$$\mathsf{assocl} = \langle id \times \pi_1, \pi_2 \cdot \pi_2 \rangle \tag{E5}$$

Vamos designar por $\langle\!| h |\!\rangle$ o hilomorfismo $[\![h, \alpha^\circ]\!]$:

$$\begin{tikzcd}
A^* \times B \arrow[r, "\alpha^\circ"] \arrow[d, "\langle\!| h |\!\rangle"'] & B + A \times (A^* \times B) \arrow[d, "id + id \times \langle\!| h |\!\rangle"] \\
C & B + A \times C \arrow[l, "h"]
\end{tikzcd}$$

É possível mostrar que, por α ser um isomorfismo, se tem a propriedade *universal*

$$k = \langle\!| h |\!\rangle \quad \Leftrightarrow \quad k \cdot \alpha = h \cdot \mathsf{F}\ k \tag{E6}$$

onde $\mathsf{F}\ f = id + id \times f$ é o functor das listas. Demonstre a partir de (E6) a propriedade

$$\pi_1 = \langle\!| \mathsf{in} |\!\rangle$$

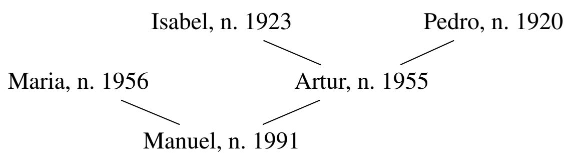
onde $\mathsf{in} = [\mathsf{nil}\, , \mathsf{cons}]$.

Questão 7 Considere o anamorfismo de números naturais

$$
\begin{array}{l}
f = [\![g]\!]\ \mathbf{where} \\
\quad g\ (x, y) = \mathbf{if}\ x > 0 \wedge y > 0\ \mathbf{then}\ i_2\ (x - 1, y - 1)\ \mathbf{else}\ i_1\ ()
\end{array}
$$

1. Faça um diagrama para f e deduza uma definição *pointwise* de f em que não ocorra nenhum combinador *pointfree* estudado nesta disciplina.
2. A função f implementa um algoritmo conhecido — consegue identificá-lo?

Questão 8 Uma árvore genealógica ascendente indica, para cada indivíduo, a árvore da sua mãe e a do seu pai, quando conhecidas. Por exemplo:



(E7)

Em Haskell podemos definir o tipo genérico para estas árvores que se segue:

$$\mathbf{data}\ GenT\ a = GenT\ \{ ind :: a, mother :: Maybe\ (GenT\ a), father :: Maybe\ (GenT\ a) \}$$

isto é:

$$GenT\ A \cong A \times (1 + GenT\ A)^2$$

1. Defina os combinadores ana-cata-hilo para este tipo de dados.
2. Seja dado, em Haskell, o tipo

   $$\mathbf{data}\ Ind\ \{ name :: String, birth :: String \}$$

   para descrever pessoas. Escreva como um anamorfismo a função

   $$names :: GenT\ Ind \to GenT\ String$$

   que elimina da árvore argumento toda a informação que não seja o nome de cada pessoa.

Questão 9 Recorde a seguinte questão do teste de 13 de Junho: dada uma função $g : X \to X$, construir um catamorfismo de números naturais $many\ g : \mathbb{N}_0 \to X \to X$ tal que, dado um número natural n e um valor x, $many\ g\ n\ x$ retorna a aplicação de g ao valor x, n vezes:

$$
\begin{array}{l}
many\ g\ 0\ x = x \\
many\ g\ (n + 1)\ x = g\ (many\ g\ n\ x)
\end{array}
$$

Desenhe em detalhe o diagrama do hilomorfismo

$$h\ g = \mathbf{tailr}\ (gen\ g)\ \mathbf{where}\ gen\ g = (\pi_2 + (id \times g)) \cdot \mathsf{distl} \cdot (\mathsf{out}_{\mathbb{N}_0} \times id)$$

que é tal que $many\ g = \overline{\mathbf{tailr}\ (g\ k)}$.

Questão 10 Relembrando o mónade de estado, complete a prova da seguinte iguadade de acções

$$modify\ f = \mathbf{do}\ \{ a \leftarrow get; put\ (f\ a) \} \tag{E8}$$

preenchendo as caixas numeradas:

$$
\begin{array}{cl}
& \mathbf{do}\ \{ a \leftarrow get; put\ (f\ a) \} \\
= & \qquad \{\ \boxed{1}\ \} \\
& get \ggg (put \cdot f) \\
= & \qquad \{\ \text{definição de } (\ggg)\ \} \\
& (\mu \cdot \boxed{2})\ get \\
= & \qquad \{\ \boxed{3}\ \} \\
& \mathsf{ap} \cdot (put \cdot f \times id) \cdot \boxed{4} \\
= & \qquad \{\ \boxed{5}\ \} \\
& \widehat{put} \cdot (f \times id) \cdot get \\
= & \qquad \{\ \text{definições de } put \text{ e } get \text{, e } \boxed{6}\ \} \\
& \boxed{7} \cdot \langle f, id \rangle \\
= & \qquad \{\ \boxed{8}\ \} \\
& \langle !, f \rangle \\
= & \qquad \{\ \text{definição de } modify\ \} \\
& modify\ f \\
\square &
\end{array}
$$