Cálculo de Programas

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

Exame da época especial — 8 de Setembro de 2020
14h00–16h00
Anfiteatro ED2-B2

- *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 colocadas.*

PROVA PRESENCIAL SEM CONSULTA (2h)

Questão 1 Considere a função $\delta = [singl \cdot i_1 , \mathsf{map}\ i_2]$ para $singl\ x = [x]$. Infira o tipo mais geral de δ e deduza a respectiva propriedade grátis, que deverá verificar analiticamente. Tenha em consideração a propriedade natural de $singl$.

Questão 2 Demonstre a lei do condicional

$$p \to (q \to c\ ,\ d)\ ,\ c \quad = \quad (p \Rightarrow q) \to c\ ,\ d$$

sabendo que

$$(p \Rightarrow q)? \quad = \quad p \to q?\ ,\ i_1 \tag{E1}$$

é uma propriedade da implicação de predicados.

Questão 3 Mostre que $(\!| g |\!) \cdot (\!| \mathsf{in} \cdot k |\!) = (\!| g \cdot k |\!)$ desde que $k \cdot \mathsf{F}\ f = \mathsf{F}\ f \cdot k$ se verifique.

Questão 4 Considere a seguinte série definida por recorrência da seguinte forma:

$$s_0 = 3$$
$$s_{n+1} = n + (s_n * 3)$$

Assim, a lista $[9, 28, 86, 261, 787, 2366, \ldots]$ mostra os primeiros termos da série. Mostre, por aplicação da lei de recursividade mútua, que a seguinte função

$$\begin{array}{l} s = \pi_2 \cdot \mathsf{for}\ loop\ \mathsf{init}\ \mathbf{where} \\ \quad loop\ (x, y) = (x + 1, x + y * 3) \\ \quad \mathsf{init} = (0, 3) \end{array}$$

calcula o n-ésimo termo da série.

Questão 5 O tipo das listas simétricas que a seguir se define em Haskell,

$$\mathbf{data}\ SSeq\ a = Nil \mid One\ a \mid More\ a\ (SSeq\ a)\ a$$

tem por objectivo aceder, com a mesma complexidade, ao primeiro e ao último elemento de uma lista não vazia. Tomando

$$\mathsf{B}\ (X, Y) = 1 + (X + X \times (Y \times X)) \tag{E2}$$

como bifunctor de base defina in e out para $SSeq$ bem como a respectiva trilogia "ana-cata-hilo". Com base nisso, identifique o gene do catamorfismo $(\!|\ g\ |\!) : SSeq\ A \rightarrow A^*$ que converte listas $SSeq$ em listas "normais" em Haskell.

Questão 6 Demonstre a igualdade $\overline{f \cdot (g \times h)} = \overline{\mathsf{ap} \cdot (id \times h)} \cdot \overline{f} \cdot g$ com base nas leis dos produtos e da exponenciação.

Questão 7 Dada a função seguinte,

$$\begin{array}{l} pad\ a\ 0\ \_ = [\,] \\ pad\ a\ n\ [\,] = \mathsf{replicate}\ n\ a \\ pad\ a\ n\ (h : t) = h : pad\ a\ (n-1)\ t \end{array}$$

avalie mentalmente as expressões $pad\ 0\ 3\ [1,2]$ e $pad\ 1\ 3\ [\,]$ para perceber o que a função faz. De seguida, identifique as funções α, β, γ e δ na seguinte formulação de $pad\ a$ como um hilomorfismo:

$$\begin{tikzcd} \mathbb{N}_0 \times A^* \arrow[r, "\beta"] \arrow[d, "pad\ a"'] & \mathbb{N}_0 \times (1 + A \times A^*) \arrow[r, "\alpha"] & \mathbb{N}_0 + (1 + A \times (\mathbb{N}_0 \times A^*)) \arrow[d, "id + (id + id \times pad\ a)"] \\ A^* & & \mathbb{N}_0 + (1 + A \times A^*) \arrow[ll, "{[\gamma, \delta]}"] \end{tikzcd}$$

Questão 8 A última construção monádica que foi estudada nesta disciplina foi o chamado *mónade livre* induzido por um qualquer functor F,

$$X \xrightarrow{\mathsf{in} \cdot i_1} \mathsf{T_F}\ X \xleftarrow{(\!|\ [id, \mathsf{in} \cdot i_2]\ |\!)} \mathsf{T_F}^2\ X \tag{E3}$$

onde $\mathsf{T_F}\ X$ tem a base $\mathsf{B}\ (X, Y) = X + \mathsf{F}\ Y$, cf:

$$\begin{tikzcd} \mathsf{T_F}\ X \arrow[r, bend left, "\mathsf{out} = \mathsf{in}^\circ"] \arrow[r, phantom, "\cong"] & X + \mathsf{F}\ (\mathsf{T_F}\ X) \arrow[l, bend left, "\mathsf{in}"] \end{tikzcd}$$

$$u = \mathsf{in} \cdot i_1$$
$$\mu = (\!|\ [id, \mathsf{in} \cdot i_2]\ |\!)$$

Conidere o seguinte caso particular:

$$\mathsf{F}\ X = 1$$
$$\mathsf{T_F}\ X = Maybe\ X$$
$$\mathsf{in} = [\mathsf{Just}, \underline{\mathsf{Nothing}}]$$

Usando a propriedade universal-cata, calcule:

$$\mu\ (\mathsf{Just}\ x) = x$$
$$\mu\ \mathsf{Nothing} = \mathsf{Nothing}$$
