Cálculo de Programas

3.º Ano de LEI+MiEI (Universidade do Minho)
Ano Lectivo de 2024/25

Exame de Recurso — 22 de Janeiro de 2025, 14h00–16h00
Salas E2-1.01 + E2-1.03

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 Deduza o tipo mais geral da função

$$\beta = (id \times \pi_2) \cdot \mathsf{assocr}$$

e infira a respectiva propriedade *grátis* (*natural*) através de um diagrama.

Questão 2 As projecções $A \xleftarrow{\pi_1} A \times B \xrightarrow{\pi_2} B$ são funções binárias e como tal podem ser "curried",

$$A^B \xleftarrow{\overline{\pi_1}} A \xrightarrow{\overline{\pi_2}} B^B \ .$$

Verifica-se que:

$$\overline{\pi_1} = \mathsf{const} \qquad \text{(E1)}$$
$$\overline{\pi_2} = \underline{id} \qquad \text{(E2)}$$

onde $\mathsf{const}\ a = \underline{a}$, a função constante que dá a como resultado. Complete as demonstrações respectivas que se seguem:

$$\begin{array}{cl} & \overline{\pi_1} = \mathsf{const} \\ \equiv & \qquad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\ & \mathsf{ap} \cdot (\mathsf{const} \times id) = \pi_1 \\ \equiv & \qquad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\ & (\mathsf{ap} \cdot (\mathsf{const} \times id))\ (a, b) = \pi_1\ (a, b) \\ \equiv & \qquad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\ & \vdots \end{array}$$

$$\begin{array}{cl} & \overline{\pi_2} = \underline{id} \\ \equiv & \qquad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\ & \vdots \end{array}$$

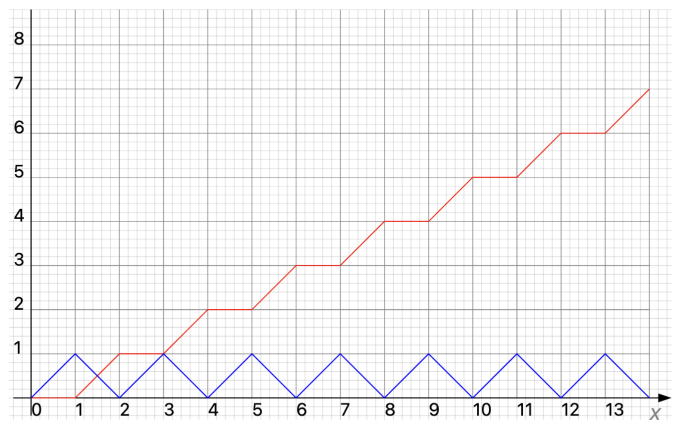
Questão 3 Mostre que $(c:)$ é o catamorfismo de listas

$$(\!|\, [\underline{[c]}, g_2] \,|\!) \ \mathbf{where}\ g_2\ (a, b : x) = b : a : x \tag{E3}$$

Acompanhe a sua resolução com um diagrama.

Questão 4 O gráfico representa as funções f e g que são soluções da equação:

$$\begin{array}{l} \langle f, g\rangle = \mathsf{for}\ loop\ (0,0)\ \mathbf{where} \\ \quad loop\ (a,b) = (a+b, h\ b) \\ \quad h\ b = b + \mathbf{if}\ even\ b\ \mathbf{then}\ 1\ \mathbf{else}\ (-1) \end{array}$$



Recorrendo à lei de recursividade mútua, mostre que f e g se poderiam ter definido *pointfree* como se segue,

$$\left\{ \begin{array}{l} f = [\underline{0}, \mathsf{add}] \cdot \mathsf{F}\ \langle f, g\rangle \cdot \mathsf{out}_{\mathbb{N}_0} \\ g = [\underline{0}, h] \cdot \mathsf{F}\ g \cdot \mathsf{out}_{\mathbb{N}_0} \end{array} \right.$$

assumindo:

$$\begin{array}{l} \mathsf{F}\ f = id + f \\ \mathsf{out}_{\mathbb{N}_0} = \mathsf{in}^{\circ} \\ \mathsf{in} = [\underline{0}, \mathsf{succ}] \end{array}$$

Questão 5 Considere definida a função

$$\begin{array}{l} \mathsf{filter}\ p\ [\,] = [\,] \\ \mathsf{filter}\ p\ (h : t) = (\mathbf{if}\ p\ h\ \mathbf{then}\ [h]\ \mathbf{else}\ [\,]) +\!\!+\ \mathsf{filter}\ t \end{array}$$

Mostra-se facilmente que a definição *pointfree* seguinte é equivalente à definição *pointwise* dada acima,

$$\mathsf{filter}\ p = \mathsf{concat} \cdot \mathsf{map}\ (p \rightarrow \mathsf{singl}, \mathsf{nil}) \tag{E4}$$

onde concat é a função (standard em Haskell) que concatena listas de listas, $\mathsf{singl}\ a = [a]$ e $\mathsf{nil}\ \_ = [\,]$. Assuma que alguém já demonstrou as seguintes propriedades de filter:

$$\mathsf{filter}\ p \cdot \mathsf{singl} = p \rightarrow \mathsf{singl}, \mathsf{nil} \tag{E5}$$

$$\mathsf{filter}\ p \cdot \mathsf{concat} = \mathsf{concat} \cdot \mathsf{map}\ (\mathsf{filter}\ p) \tag{E6}$$

Complete a demonstração iniciada abaixo de que a filtragem de listas é uma operação idempotente, isto é, filtrar duas vezes seguidas é a mesma coisa que filtrar uma vez só:

$$\begin{array}{cl} & \mathsf{filter}\ p \cdot \mathsf{filter}\ p = \mathsf{filter}\ p \\ \equiv & \quad \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\ & \mathsf{filter}\ p \cdot \mathsf{concat} \cdot \mathsf{map}\ (p \rightarrow \mathsf{singl}, \mathsf{nil}) = \mathsf{filter}\ p \\ \equiv & \quad \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\ & \vdots \end{array}$$

Questão 6 Mostre que o anamorfismo

$$suffixes = [\![g]\!] \ \mathbf{where}\ g = (id + \langle \mathsf{cons}, \pi_2 \rangle) \cdot \mathsf{out}$$

que calcula os sufixos de uma lista é a função:

$$\begin{cases} suffixes\ [\,] = [\,] \\ suffixes\ (h : t) = (h : t) : suffixes\ t \end{cases}$$

Questão 7 A função

$$\begin{array}{l} hilo\ [\,] = 0 \\ hilo\ x = \mathbf{if}\ m > i\ \mathbf{then}\ i + 1\ \mathbf{else}\ i\ \mathbf{where} \\ \quad m = minimum\ x \\ \quad i = hilo\ (delete\ m\ x) \end{array}$$

pode ser vista como um hilomorfismo de listas

$$\begin{tikzcd}
\mathbb{N}_0{}^* \arrow[r, bend left, "g"] \arrow[d, "{[\![g]\!]}"] \arrow[dd, bend right=50, "hilo"'] & \mathsf{F}\,\mathbb{N}_0{}^* \arrow[d, "{\mathsf{F}\,[\![g]\!]}"'] \arrow[dd, bend left=50, "\mathsf{F}\,hilo"] \\
\mathbb{N}_0{}^* \arrow[r, bend left, "\mathsf{out}=\mathsf{in}^\circ"] \arrow[r, phantom, "\cong"] \arrow[d, "(\!| f |\!)"] & \mathsf{F}\,\mathbb{N}_0{}^* \arrow[l, bend left, "\mathsf{in}"] \arrow[d, "\mathsf{F}\,(\!| f |\!)"'] \\
\mathbb{N}_0 & \mathsf{F}\,\mathbb{N}_0 \arrow[l, bend left, "f"]
\end{tikzcd}$$

onde:

$$f = [\underline{0}, h]\ \mathbf{where}\ h\ (m, i) = \mathbf{if}\ m > i\ \mathbf{then}\ i + 1\ \mathbf{else}\ i. \tag{E7}$$

- Infira o outro gene g do hilomorfismo.
- Descreva por palavras suas o que faz o anamorfismo $[\![g]\!]$.

Questão 8 Prove a seguinte generalização monádica da lei de fusão-$\times$:

$$\langle f \bullet g, h \bullet g \rangle = \langle f \bullet id, h \bullet id \rangle \cdot g$$
