Cálculo de Programas

2.º Ano de LCC+LEI (Universidade do Minho)
Ano Lectivo de 2012/13

Exame de recurso — 08 de Julho de 2013
09h00
Salas CP2 201, 202, 203 e 204

PROVA SEM CONSULTA (2h30m)

Questão 1 Mostre que a expressão

$$\langle [f\ , h] \cdot (\pi_1 + \pi_1), [g\ , k] \cdot (\pi_2 + \pi_2) \rangle$$

simplifica em $[f \times g\ , h \times k]$.

Questão 2 Recordando da biblioteca `Cp.hs` o isomorfismo $\mathsf{undistl} = [i_1 \times id\ , i_2 \times id]$, use diagramas para:

- descrever o tipo de $\mathsf{undistl}$;
- inferir a propriedade *natural* (ie. "grátis'") da função distl que é inversa de $\mathsf{undistl}$. (**NB:** tem de formular essa propriedade mas não se pede para a provar analiticamente.)

Questão 3 Seja dado um predicado p e uma função f tal que:

$$p \cdot f = p \tag{1}$$

Mostre que a seguinte composição de condicionais de McCarthy envolvendo p e f

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

se pode reduzir a f sabendo-se que, entre outras leis que conhece, se tem:

$$p \rightarrow (p \rightarrow a, b), (p \rightarrow c, d) = p \rightarrow a, d \tag{2}$$

$$p \rightarrow a, a = a \tag{3}$$

Questão 4 Seja dada a função, em Haskell:

$$comp :: (a \rightarrow c, b \rightarrow a) \rightarrow b \rightarrow c$$
$$comp = \overline{ap \cdot (id \times ap) \cdot \mathsf{assocr}}$$

Apresente justificações para os passos do cálculo seguinte que mostra que $comp\ (f, g) = f \cdot g$:

$$comp = \overline{ap \cdot (id \times ap) \cdot \mathsf{assocr}}$$
$$\equiv \quad \{ \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \quad \}$$
$$ap \cdot (comp \times id) = ap \cdot (id \times ap) \cdot \mathsf{assocr}$$

$\equiv \quad \{ \ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \ \}$

$ap \cdot (comp \times id) = ap \cdot (\pi_1 \times ap) \cdot \langle \pi_1, \pi_2 \times id \rangle$

$\equiv \quad \{ \ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \ \}$

$(ap \cdot (comp \times id)) \ ((f, g), x) = (ap \cdot (\pi_1 \times ap) \cdot \langle \pi_1, \pi_2 \times id \rangle) \ ((f, g), x)$

$\equiv \quad \{ \ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \ \}$

$ap \ (comp \ (f, g), x) = (ap \cdot (\pi_1 \times ap)) \ ((f, g), (g, x))$

$\equiv \quad \{ \ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \ \}$

$comp \ (f, g) \ x = ap \ (f, g \ x)$

$\equiv \quad \{ \ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \ \}$

$comp \ (f, g) \ x = f \ (g \ x)$

$\equiv \quad \{ \ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \ \}$

$comp \ (f, g) = f \cdot g$

Questão 5 O seguinte par de funções mutuamente recursivas

$$\left\{ \begin{array}{l} impar \ 0 = False \\ impar \ (n+1) = par \ n \end{array} \right. \qquad \left\{ \begin{array}{l} par \ 0 = True \\ par \ (n+1) = impar \ n \end{array} \right.$$

que testam a paridade de um número, é equivalente ao seguinte par de equações

$$\begin{array}{l} impar \cdot \mathsf{in} = [\underline{False} \ , par] \\ par \cdot \mathsf{in} = [\underline{True} \ , impar] \end{array}$$

onde $\mathsf{in} = [\underline{0} \ , \mathsf{succ}]$ e $\mathsf{succ} \ n = n + 1$. Mostre, recorrendo às leis da recursividade múltipla e da troca, que par e $impar$ se podem combinar num único ciclo-for com duas variáveis,

$$\begin{array}{l} impar = \pi_1 \cdot imparpar \\ par = \pi_2 \cdot imparpar \\ imparpar = \mathsf{for} \ \mathsf{swap} \ (False, True) \end{array}$$

sabendo que catamorfismos de naturais são ciclos-for, isto é, $(\![[\underline{i} \ , b]]\!) = \mathsf{for} \ b \ i$.

Questão 6 Seja $f : [A + B] \to [B]$ a função

$$\begin{array}{l} f \ [\,] = [\,] \\ f \ ((i_1 \ a) : t) = f \ t \\ f \ ((i_2 \ b) : t) = b : f \ t \end{array}$$

que selecciona todos os Bs que ocorrem numa lista de As ou Bs. Pretende-se mostrar que f é o catamorfismo

$$f = (\![[nil \ , [\pi_2 \ , cons] \cdot \mathsf{distl}]]\!)$$

onde distl é a inversa da função $\mathsf{undistl}$ que está codificada em Haskell no módulo `Cp.hs`. Faça-o segundo os passos seguintes:

- Comece por mostrar que a declaração $f = (\![[nil \ , [\pi_2 \ , cons] \cdot \mathsf{distl}]]\!)$ é equivalente ao par de equações

  $$\left\{ \begin{array}{l} f \cdot nil = nil \\ f \cdot cons = [\pi_2 \ , cons] \cdot ((id \times f) + (id \times f)) \cdot \mathsf{distl} \end{array} \right.$$

  e que a segunda equação desse par é equivalente a

  $$f \cdot cons \cdot \mathsf{undistl} = [f \cdot \pi_2 \ , cons \cdot (id \times f)].$$

- Termine apresentando justificações para os passos restantes do cálculo:

$$
\begin{array}{ll}
& \left\{\begin{array}{l} f \cdot nil = nil \\ f \cdot cons \cdot \mathsf{undistl} = [f \cdot \pi_2 \, , cons \cdot (id \times f)] \end{array}\right. \\
\equiv & \quad \{ \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \quad \} \\
& \left\{\begin{array}{l} f \cdot nil = nil \\ [f \cdot cons \cdot (i_1 \times id) \, , f \cdot cons \cdot (i_2 \times id)] = [f \cdot \pi_2 \, , cons \cdot (id \times f)] \end{array}\right. \\
\equiv & \quad \{ \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \quad \} \\
& \left\{\begin{array}{l} f \cdot nil = nil \\ f \cdot cons \cdot (i_1 \times id) = f \cdot \pi_2 \\ f \cdot cons \cdot (i_2 \times id) = cons \cdot (id \times f) \end{array}\right. \\
\equiv & \quad \{ \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \quad \} \\
& \left\{\begin{array}{l} f \; [\,] = [\,] \\ f \; ((i_1 \; a) : t) = f \; t \\ f \; ((i_2 \; b) : t) = b : f \; t \end{array}\right.
\end{array}
$$

Questão 7 A função que filtra uma lista por um predicado p

$$filter \; p = (\!|[nil \, , conc \cdot ((p \rightarrow singl, nil) \times id)]|\!) \tag{4}$$

é um catamorfismo de listas, onde $nil \; \_ = [\,]$, $singl \; a = [\, a \,]$ e $conc \; (x, y) = x +\!\!+ \; y$. Suponha agora que quer contar quantos elementos de uma lista satisfazem p:

$$count \; p = \mathsf{length} \cdot (filter \; p) \tag{5}$$

Usando leis dos catamorfismos (entre outras) mostre que $count \; p$ se obtém de $filter \; p$ substituindo nil por $zero$, $conc$ por add e $sngl$ por one,

$$count \; p = (\!|[zero \, , add \cdot ((p \rightarrow one, zero) \times id)]|\!).$$

em que $zero = \underline{0}$, $one = \underline{1}$ e $add \; (n, m) = n + m$. **NB:** assuma a propriedade $\mathsf{length} \cdot conc = add \cdot (\mathsf{length} \times \mathsf{length})$.

Questão 8 Suponha que sabe que a propriedade

$$g \cdot \mathsf{in} = ! + \langle cons, \pi_2 \rangle \tag{6}$$

é válida para o gene g do anamorfismo $suffixes = [\!(g)\!]$. Mostre, justificadamente, que $suffixes$ é a função que escreveria em Haskell desta forma:

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

Questão 9 Recorra às leis dos catamorfismos para demonstrar a propriedade natural

$$(\mathsf{LTree} \; f) \cdot \mathsf{mirror} \quad = \quad \mathsf{mirror} \cdot (\mathsf{LTree} \; f) \tag{7}$$

onde mirror é o catamorfismo

$$
\begin{array}{l}
\mathsf{mirror} :: \mathsf{LTree} \; a \rightarrow \mathsf{LTree} \; a \\
\mathsf{mirror} = (\!| \; \mathsf{in} \cdot (id + \mathsf{swap}) |\!)
\end{array}
$$

que "espelha" uma árvore e $\mathsf{LTree} \; f = (\!| \; \mathsf{in} \cdot (f + id) |\!)$ é o correspondente functor de tipo.

Questão 10 Demonstre ou refute a seguinte propriedade da composição monádica:

$$(u \cdot f) \bullet (u \cdot g) = u \cdot (f \cdot g)$$