Cálculo de Programas

2.º Ano de LCC (8504N1)
Ano Lectivo de 2006/07

Exame (época de recurso) — 18 de Julho 2007
09h30
Salas 1103, 1104

NB: *Esta prova consta de* ***8*** *alíneas que valem, cada uma, 2.5 valores. Utilize folhas de resposta diferentes para cada grupo.*

PROVA SEM CONSULTA (2 horas)

GRUPO I

Questão 1 Qualquer programador de C sabe que, sempre que define uma `struct` de apontadores $(A+1) \times (B+1)$, consegue ter um tipo de dados que pode representar alternativamente o produto o $A \times B$ (`struct`) e o co-produto $A + B$ (`union`).

Apresente, em notação *pointfree*, as definições dos ismorfismos i_1 a i_5 em

$$
\begin{tikzcd}
(A+1)\times(B+1) & ((A+1)\times B)+((A+1)\times 1) \arrow[l, "i_1"'] & (A\times B+1\times B)+(A\times 1+1\times 1) \arrow[l, "i_2"'] \\
A\times B+((B+A)+1) \arrow[r, "i_5"'] & A\times B+(B+(A+1)) \arrow[r, "i_4"'] & (A\times B+B)+(A+1) \arrow[u, "i_3"']
\end{tikzcd}
$$

que testemunham esse facto.

Questão 2 Demonstre a seguinte propriedade do combinador condicional de McCarthy

$$(\neg \cdot p) \rightarrow f, g \quad = \quad p \rightarrow g, f \tag{1}$$

sabendo que é válida a propriedade

$$(\neg \cdot p)? \quad = \quad coswap \cdot (p?) \tag{2}$$

onde $coswap = [i_2\ , i_1]$.

GRUPO II

Questão 3 Considere a função, em Haskell

```
pow x 0 = 1
pow x (n+1) = x * pow x n
```

que calcula potências de expoente natural. Aplique-lhe a transformada *pointfree* e demonstre que, para todo o x, a função $(pow\ x)$ é o catamorfimo $(\!|[\underline{1}\ , (x*)]|\!)$.

Questão 4 Explorando a propriedade $x^{2n} = (x^n)^2 = (x^n)(x^n)$ é possível sofisticar o catamorfismo da Questão 3, transformando-o no hilomorfismo

$$pow\ x = [\![\ [\underline{1}\ , [(x*)\ , mul]], g\ ]\!] \qquad \text{— onde } mul(x,y) = x * y \tag{3}$$

com estrutura intermédia de tipo

```
data XT = Zero | One XT | Two (XT, XT)

inXT = either (const Zero) (either One Two)
```

Defina $outXT$ e $hyloXT$ e complete a definição do gene g do hilomorfismo proposto, que deverá testar se o exponente é zero, par ou ímpar e agir em conformidade. (Não se esqueça de desenhar o respectivo diagrama.)

Questão 5 Complete os passos da demonstração seguinte que prova a lei de absorção-cata para o tipo $LTree$:

$$\begin{array}{ll}
& (\!|\alpha|\!) \cdot (LTree\ f) = (\!|\alpha \cdot (f + id)|\!) \\
\equiv & \qquad \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\
& (\!|\alpha|\!) \cdot (\!|in \cdot (f + id)|\!) = (\!|\alpha \cdot (f + id)|\!) \\
\Leftarrow & \qquad \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\
& (\!|\alpha|\!) \cdot (in \cdot (f + id)) = \alpha \cdot (f + id) \cdot (id + (\!|\alpha|\!) \times (\!|\alpha|\!)) \\
\equiv & \qquad \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\
& \alpha \cdot (id + (\!|\alpha|\!) \times (\!|\alpha|\!)) \cdot (f + id) = \alpha \cdot (f + id) \cdot (id + (\!|\alpha|\!) \times (\!|\alpha|\!)) \\
\equiv & \qquad \{ \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ \} \\
& \alpha \cdot (f + (\!|\alpha|\!) \times (\!|\alpha|\!)) = \alpha \cdot (f + (\!|\alpha|\!) \times (\!|\alpha|\!)) \\
\equiv & \qquad \{ \ \text{trivial} \ \} \\
& \text{TRUE}
\end{array}$$

Questão 6 A função

```
rep :: BTree a -> [(Int, a)]
rep t = aux(t,1)

aux :: (BTree a, Int) -> [(Int, a)]
aux(Empty,i)          = []
aux(Node(a,(t,t)),i) = [(i, a)] ++ aux(t,2*i) ++ aux(t,2*i+1)
```

que representa árvores binárias sob a forma de *arrays* modelados por listas de pares *(posição,elemento)* foi assunto de um exercício do 2.º trabalho desta disciplina. Adapte rep à representação de árvores do tipo $LTree$ e desenhe-a sob a forma de um diagrama de hilmorfismo.

GRUPO III

Questão 7 Considere, em Haskell, a função

```
mmul x y = do { a <- x;
                b <- y;
                return(a*b)
              }
```

Apresente (justificando) os resultados que um interpretador de Haskell deverá dar como resultado da avaliação das expressões seguintes:

```
mmul [2,3][4,5]
mmul Nothing (Just 4)
mmul (Just 3) [7]
mmul [2,3][]
```

Questão 8 Complete a demonstração que se segue do facto

$$do \{ a \leftarrow x \, ; \, u(f\ a) \} \quad = \quad (\mathsf{T}\, f)\ x \tag{4}$$

(onde u é a função `return` do Haskell) válido para toda a mónada T:

$$
\begin{array}{cl}
 & do \{ a \leftarrow x \, ; \, u(f\ a) \} \\
= & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
 & x \gg\!\!= (\lambda a . u(f\ a)) \\
= & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
 & x \gg\!\!= (u \cdot f) \\
= & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
 & (\mu \cdot \mathsf{T}(u \cdot f))x \\
= & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
 & (\mu \cdot (\mathsf{T}\, u) \cdot (\mathsf{T}\, f))x \\
= & \quad \{ \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad \} \\
 & (\mathsf{T}\, f)x
\end{array}
$$
