

Métodos de Programação I / Cálculo de Programas

2.º Ano de LESI (5303O7) / LCC (8504N1)
Ano Lectivo de 2006/07

Exame (época de recurso) — 18 de Setembro 2007
14h00
Salas 1201, 1206, 1212

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 Identifique qual das seguintes funções

$$[id, id] \quad (1)$$

$$[\langle \text{TRUE}, id \rangle, \langle \text{FALSE}, id \rangle] \quad (2)$$

$$[\langle \text{TRUE}, \text{FALSE} \rangle, id] \quad (3)$$

$$id + id \quad (4)$$

estabelece o isomorfismo

$$Bool \times A \cong A + A$$

da direita para a esquerda. Derive a versão *point-wise* dessa função.

Questão 2 Parta da propriedade universal do produto para provar que a igualdade

$$(g \cdot h) \times (i \cdot j) = (g \times i) \cdot (h \times j) \quad (5)$$

se verifica.

Questão 3 Considere o seguinte tipo de dados indutivo:

```
data BTree a = Empty | Node (a, (BTree a, BTree a))
```

Defina `inBTree`, `outBTree`, `cataBTree` e `anaBTree`. Acompanhe as suas definições com os diagramas respectivos.

GRUPO II

Questão 4

Considere a seguinte função definida sobre elementos do tipo BTree.

```
mirror Empty = Empty
mirror (Node (x, (l,r))) = Node (x, (mirror r, mirror l))
```

Demonstre que `mirror` pode ser definida como o seguinte catamorfismo de árvores:

```
mirror = cataLTree (inBTree . (id -|- id >< swap))
```

Questão 5 Defina a função `preorder :: BTree a -> [a]` como um catamorfismo sobre árvores binárias e como um anamorfismo sobre listas.

Questão 6 Use a lei de fusão-cata para provar a validade do seguinte facto, em Haskell:

$$x + \text{foldr } (+) \ y = \text{foldr } (+) \ (x+y)$$

Sugestão: Interprete `foldr` como um catamorfismo de listas, assumindo $f = (x+)$. Na prova pode usar a seguinte propriedade da soma (*uncurried*)

$$f \ . \ (+) = (+) \ . \ (\text{id} \ >< \ f)$$

que corresponde em notação *pointwise* a $x + (y + z) = y + (x + z)$.

GRUPO III

Questão 7 Considere, em Haskell, a função

```
monbf :: .....
monbf f x y = do { a <- x;
                  b <- y;
                  return(f a b)
                }
```

Complete as reticências e apresente (justificando) os resultados que um interpretador de Haskell deverá dar como resultado da avaliação das expressões seguintes:

```
monbf (*) [2,3] [4,5]
monbf (/) Nothing (Just 0)
monbf (:) [1,0] [[]]
monbf (/) [] [1,0]
```

Questão 8 Demonstre a propriedade da composição monádica

$$(f \cdot g) \bullet h = f \bullet (Tg \cdot h) \tag{6}$$
