

Cálculo de Programas

2.º ano das Licenciaturas em
Ciências da Computação e Engenharia Informática
UNIVERSIDADE DO MINHO

2013/14 - Ficha nr.º 4

1. Demonstre a segunda lei de fusão do condicional de McCarthy:

$$(p \rightarrow f, g) \cdot h = (p \cdot h) \rightarrow (f \cdot h), (g \cdot h) \quad (1)$$

2. Sabendo que as igualdades

$$p \rightarrow k, k = k \quad (2)$$

$$(p? + p?) \cdot p? = (i_1 + i_2) \cdot p? \quad (3)$$

se verificam, demonstre as seguintes propriedades do mesmo combinador:

$$\langle (p \rightarrow f, h), (p \rightarrow g, i) \rangle = p \rightarrow \langle f, g \rangle, \langle h, i \rangle \quad (4)$$

$$\langle f, (p \rightarrow g, h) \rangle = p \rightarrow \langle f, g \rangle, \langle f, h \rangle \quad (5)$$

$$p \rightarrow (p \rightarrow a, b), (p \rightarrow c, d) = p \rightarrow a, d \quad (6)$$

3. Considere a função $\text{undistl} = [i_1 \times id, i_2 \times id]$.

- (a) Identifique o isomorfismo que ela testemunha, desenhando o diagrama respectivo.
- (b) Recorra à lei da troca (ficha anterior) para re-escrever undistl sob a forma de um *split*.
- (c) Demonstre a seguinte propriedade: $\pi_1 \cdot \text{undistl} = \pi_1 + \pi_1$.
- (d) Derive a definição *pointwise* de undistl .

4. Considere a função $\text{iso} = \langle ! + !, [id, id] \rangle$.

- (a) Identifique o isomorfismo que ela testemunha, desenhando-a sob a forma de um diagrama de tipos.
- (b) Derive (por cálculo analítico) a propriedade (dita *grátis*) de iso ,

$$(id \times f) \cdot \text{iso} = \text{iso} \cdot (f + f)$$

desenhando previamente o correspondente diagrama.

- (c) Derive uma definição em Haskell *pointwise* de iso .

5. Considere a função $\text{coassocr} = [id + i_1, i_2 \cdot i_2]$ que é uma das testemunhas do isomorfismo $(A + B) + C \cong A + (B + C)$, da esquerda para a direita. Calcule a sua conversa coassocl a partir da equação

$$\text{coassocl} \cdot \text{coassocr} = id$$

Sugestão: Faça-o resolvendo em ordem a x, y e z a seguinte versão dessa equação:

$$\underbrace{[x, [y, z]]}_{\text{coassocl}} \cdot \underbrace{[id + i_1, i_2 \cdot i_2]}_{\text{coassocr}} = id$$

6. Demonstre a seguinte igualdade, em que participa um dos lados da função coassoc

$$(h + (g + f)) \cdot i_2 \cdot i_2 = i_2 \cdot i_2 \cdot f$$

e em que f , g e h se assumem devidamente tipadas.

7. Se tentar definir em Haskell o combinador

$$\text{comb } f \ g = \langle f, [g, f] \rangle$$

obterá a seguinte mensagem de erro:

```
comb.hs:3:29:
  Occurs check: cannot construct the infinite type: b = Either a b
    Expected type: b -> c
    Inferred type: Either a b -> b1
  In the second argument of `either`, namely `f`
  In the second argument of `split`, namely `(either g f)`
Failed, modules loaded: Cp.
```

Explique a mensagem de erro acima mostrando que é impossível construir um diagrama para o tipo de $\text{comb } f \ g$.

8. Considere a seguinte declaração de um tipo de *árvores binárias*, em Haskell:

```
data LTree a = Leaf a | Fork (LTree a, LTree a)
```

Indagando os tipos dos construtores `Leaf` e `Fork`, por exemplo no GHCi,

```
*LTree> :t Fork
Fork :: (LTree a, LTree a) -> LTree a
*LTree> :t Leaf
Leaf :: a -> LTree a
```

é fácil desenhar o diagrama que explica a construção da função

$$\text{inLTree} = [\text{Leaf}, \text{Fork}]$$

Desenhe-o e calcule a sua inversa

```
outLTree :: LTree a -> Either a (LTree a, LTree a)
outLTree (Leaf a) = i1 a
outLTree (Fork (x, y)) = i2 (x, y)
```

resolvendo a equação

$$\text{outLTree} \cdot \text{inLTree} = \text{id}$$

em ordem a outLTree .