Métodos de Programação I
Exame – 2a. chamada

2 de Fevereiro 2005

Escreva o seu nome e número nas 3 folhas deste exame

1. Considere o seguinte raciocínio:

$$
\begin{array}{ll}
 & k = [f, g] \iff \left\{ \begin{array}{l} k \cdot i_1 = f \\ k \cdot i_2 = g \end{array} \right. \\
\equiv & \qquad \{ \ldots \text{(justifique)} \ldots \} \\
 & h \cdot [i, j] = [f, g] \iff \left\{ \begin{array}{l} (h \cdot [i, j]) \cdot i_1 = f \\ (h \cdot [i, j]) \cdot i_2 = g \end{array} \right. \\
\equiv & \qquad \{ \ldots \text{(justifique)} \ldots \} \\
 & h \cdot [i, j] = [f, g] \iff \left\{ \begin{array}{l} h \cdot ([i, j] \cdot i_1) = f \\ h \cdot ([i, j] \cdot i_2) = g \end{array} \right. \\
\equiv & \qquad \{ \ldots \text{(justifique)} \ldots \} \\
 & h \cdot [i, j] = [f, g] \iff \left\{ \begin{array}{l} h \cdot i = f \\ h \cdot j = g \end{array} \right. \\
\equiv & \qquad \{ \ldots \text{(justifique)} \ldots \} \\
 & h \cdot [i, j] = [h \cdot i, h \cdot j]
\end{array}
$$

   Diga qual a propriedade de que o raciocínio partiu e indique a lei utilizada em cada passo.

2. Relembre

$$assocr : (A \times B) \times C \to A \times (B \times C)$$

   e escreva a sua dual, isto é, a bijecção que estabelece o isomorfismo

$$A + (B + C) \quad \cong \quad (A + B) + C$$

   Codifique essa função em Haskell.

3. Seja $f = [\langle i_1, i_1 \rangle, i_2 \times i_2]$ e $g = \langle id + \pi_1, id + \pi_2 \rangle$.
   Mostre que as funções têm o mesmo tipo e prove que nenhuma das duas é um isomorfismo.

4. Uma árvore binária (etiquetada nos nós) pode ser vista como uma lista de pares (elemento, árvore), a que se chama a *espinha* da árvore.

   Por exemplo, a árvore

$$
\begin{aligned}
&(\mathit{Node}\ 1\ (\mathit{Node}\ 2\ (\mathit{Node}\ 4\ \mathit{Empty}\ \mathit{Empty})\ (\mathit{Node}\ 5\ \mathit{Empty}\ \mathit{Empty}))\\
&\qquad\qquad (\mathit{Node}\ 3\ (\mathit{Node}\ 6\ \mathit{Empty}\ \mathit{Empty})\ (\mathit{Node}\ 7\ \mathit{Empty}\ \mathit{Empty})))
\end{aligned}
$$

   tem a seguinte espinha à direita:

$$
\begin{aligned}
&[(1, (\mathit{Node}\ 2\ (\mathit{Node}\ 4\ \mathit{Empty}\ \mathit{Empty})\ (\mathit{Node}\ 5\ \mathit{Empty}\ \mathit{Empty}))),\\
&\ (3, (\mathit{Node}\ 6\ \mathit{Empty}\ \mathit{Empty})),\\
&\ (7, \mathit{Empty})]
\end{aligned}
$$

   Esta correspondência exprime o seguinte isomorfismo:

$$\mathit{spine} : \mathit{BTree}\ a \rightarrow [(a, \mathit{BTree}\ a)]$$

   (a) Escreva a função *inversa* de *spine* como catamorfismo de listas e desenhe o respectivo diagrama.

   (b) Escreva *spine* como anamorfismo de listas e desenhe o respectivo diagrama. Diga, justificando, se esta função pode ser escrita como catamorfismo de *BTree*.

5. Considere a seguinte definição recursiva *point-free*:

$$f = [\mathsf{succ} \cdot \underline{0}, \mathsf{plus} \cdot \langle f \cdot \mathsf{pred}, f \cdot \mathsf{pred} \cdot \mathsf{pred} \rangle] \cdot \mathsf{zeroOrOne}?$$

em que

$$\mathsf{zeroOrOne}\ x = (x == 0) \mid\mid (x == 1)$$
$$\mathsf{plus} = \mathsf{uncurry}\ (+)$$

Transforme a definição de f num hilomorfismo, justificando todos os passos. Desenhe o respectivo diagrama, identificando claramente o functor do tipo intermédio utilizado.

6. Considere a seguinte função

```
bif (0,x) = [x]
bif (1,x) = [x+1]
bif (n+2,x) = (bif (n+1,x+1)) ++ (bif (n,x+1))
```

(a) Escreva-a como um hilomorfismo.

(b) Diga qual a estrutura de dados intermédia envolvida no cálculo de `bif (4,0)`. Note a semelhança entre esta função e a do cálculo dos números de Fibonnacci. Defina uma função de cálculo do n-ésimo número de Fibonnacci usando esta função.