Métodos de Programação I

2.º Ano da LMCC (7003N5) + LESI (5303O7)
Ano Lectivo de 2001/2002

Exame (época especial) — 18 de Novembro 2002
17h00
Salas 2306

PROVA SEM CONSULTA (2 horas)

Questão 1 Derive a versão *pointwise* em Haskell da função f caracterizada pela seguinte equação

$$f \circ [\underline{0}, \text{succ}] = [\langle \underline{0}, \underline{1} \rangle, \langle \pi_2, \text{uncurry } (+) \rangle] \circ (id + f)$$

Defina f como um catamorfismo no tipo conveniente. Apresente os diagramas necessários.

Questão 2 Considere a seguinte função para calcular as posições de um elemento numa lista:

```
posicoes :: (Eq a) => a -> [a] -> [Int]
posicoes a l = [y | (x,y) <- zip l [1..], x == a]
```

1. Apresente uma versão recursiva desta função.
2. Reescreva a definição da função `posicoes x` definida na alínea anterior como um catamorfismo. Apresente os diagramas correspondentes.
3. Modifique o gene do catamorfismo da alínea anterior de forma a obter uma função que calcula quantas vezes um determinado elemento ocorre numa lista.

Questão 3 A função `cref :: (Eq a) => [a] -> [(a,Int)]` calcula, para cada elemento de uma lista, o número de vezes que ele ocorre nessa lista. Por exemplo, `cref [1,2,3,1,3,1] = [(1,3),(2,1),(3,2)]`.

1. Defina a função `cref?`como um anamorfismo.
2. Considere agora uma variante desta função que, em vez de calcular o número de ocorrências, calcula as posições onde os elementos ocorrem. Explique por palavras suas por que é que esta função não pode ser descrita por uma modificação do gene da função da alínea anterior.
3. Apresente uma possível definição da função referida na alínea anterior escrita como um hilomorfismo.

Questão 4 Considere a seguinte definição de funções de transformação de estado.

```
data ST estado valor = ST estado -> (estado,valor)
```

1. Complete a seguinte definição:

   ```
   instance Monad (ST estado) where
      return a = ST (\s -> ...)
      (ST g) >>= f = ST (\s -> ...)
   ```

2. Para esta instância de `Monad`, defina a multiplicação μ.
3. Defina `ST s` como uma instância da classe `Functor`.
