

Métodos de Programação I

Exame – 1a. chamada

18 de Janeiro 2005

Escreva o seu nome e número nas 3 folhas deste exame

1. Considere as seguintes funções

$$f = \langle [\pi_1, \pi_1], \pi_2 + \pi_2 \rangle$$
$$g = [id \times i_1, id \times i_2]$$

Determine o seu tipo e mostre que são de facto a mesma função (que implementa um conhecido isomorfismo).

2. Apresente, justificando todos os passos, uma prova equacional do facto

$$(id \times \pi_2) \cdot assocr = \pi_1 \times id$$

em que *assocr* é o bem conhecido isomorfismo de associatividade do produto:

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

3. Derive, justificando todos os passos, a versão *pointwise* em Haskell da função *f* caracterizada pela seguinte equação

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

Qual o tipo da função?

4. Para descrever documentos HTML pode-se utilizar o seguinte tipo:

```
data DocHtml = S String | T String DocHtml | L [DocHtml]
```

Por exemplo, o texto HTML

```
<html>
<head>
<title>
Hello
</title>
</head>
<body>
Hello Word
</body>
</html>
```

será representado pelo termo

```
hello = T "html" (L [T "head" (L [T "title" (S "Hello") ]),
                    T "body" (L [S "Hello Word"])] )
```

- (a) Diga qual o functor de base deste tipo, e desenhe o diagrama dos catamorfismos sobre DocHtml.
- (b) Defina, como um catamorfismo sobre DocHtml, uma função

```
showHtml :: DocHtml -> String
```

que produz uma *string* a partir de uma expressão do tipo DocHtml (por exemplo, a partir de `hello` poder-se-ia obter um texto parecido com o inicialmente apresentado).

5. A função que calcula as sublistas de uma lista pode ser definida por:

```
pSet [] = [[]]
pSet (h:t) = (map (h:) (pSet t)) ++ pSet t
```

- (a) Prove que esta definição é equivalente à seguinte, em que F é o functor de base do tipo `Lista` e `append = uncurry(++)`.

$$(\llbracket [] \rrbracket, \text{append} \cdot \langle \text{ap} \cdot ((\text{map} \cdot (:)) \times \text{id}), \pi_2 \rangle \rrbracket)_F$$

- (b) Uma definição alternativa desta função é como um hilomorfismo de árvores binárias cujo catamorfismo tem o gene

```
c = either (const [[]]) (\x a b -> (map (x:) a) ++ b)
```

Defina o anamorfismo que, composto com este catamorfismo é (intensionalmente) equivalente à função `pSet` acima. Desenhe os respectivos diagramas.

6. Escreva uma definição como hilomorfismo da seguinte versão da função factorial, que utiliza um acumulador. Desenhe os respectivos diagramas.

```
fact 0 y = y
fact n y = fact (n-1) (n*y)
```

Sugestão: Comece por reescrever a função, explicitando a função devolvida como resultado.

