

Construção de Software Primeira Avaliação

Rodrigo Bonifácio

11 de abril de 2015

Questão 1

Considere a definição

```
data Expr
  = Const Int
  | Plus Expr Expr
  | Mult Expr Expr
```

que é usada para representar expressões aritméticas básicas em Haskell. Implemente um módulo Haskell (**Expression.hs**) com a definição acima.

- (a) Escreva uma expressão (**exp1 :: Expr**) que representa o valor

(1 + 2)*(3 + 4)

como uma instância do *datatype* **Expr**. (**1pt**)

- (b) Escreva uma função **eval :: Expr -> Int** que computa o resultado da avaliação de uma expressão qualquer do tipo **Expr**. Para o exemplo do item (a), o resultado da avaliação seria 21. (**3 pts**)

Questão 2

Considerando a seguinte definição de uma árvore binária genérica, implemente um módulo Haskell (**Tree.hs**) contendo o *datatype* **Tree**.

```
data Tree a = Leaf
  | Node a (Tree a) (Tree a)
```

- (a) Implemente uma função que calcula o número de nós presentes em uma árvore, conforme a definição do nosso tipo **Tree**. (**3 pts**).
- (b) Implemente uma função de alta-ordem

```
mapT :: (a -> b) -> (Tree a) -> Tree b
mapT f tree = ⊥
```

que aplica uma função (**f :: a -> b**) aos elementos de uma árvore de "as" e retorna uma nova árvore de "bs". (**3 pts.**)