Aula Teórico-prática e Prática 10

Programação Funcional

LCC 1º ano (2009/2010)

Relembre da aula anterior os tipos.

```
data ExpInt = Const Int
            | Simetrico ExpInt
            | Mais ExpInt ExpInt
            | Menos ExpInt ExpInt
            | Mult ExpInt ExpInt

type ExpN = [Parcela]
type Parcela = [Int]
```

juntamente com as funções

```
calcula :: ExpInt -> Int
expString :: ExpInt -> String
posfix:: ExpInt -> String
calcN :: ExpN -> Int
normaliza :: ExpInt -> ExpN
```

Uma possível generalização será considerar expressões cujas constantes são de um qualquer tipo numérico (i.e., da classe `Num`). A definição desses tipos será agora

```
data Exp a = Const a
           | Simetrico (Exp a)
           | Mais (Exp a) (Exp a)
           | Menos (Exp a) (Exp a)
           | Mult (Exp a) (Exp a)

data ExpN a = ExpN [Parcela a]
type Parcela a = [a]
```

1. Adapte as definições que apresentou das funções referidas a estes novos tipos.
2. Complete as seguintes definições de instâncias:
   (a) `instance (Eq a) => Eq (Exp a) where ...`
   (b) `instance (Show a) => Show (Exp a) where ...`
3. A definição da classe `Num` é:

   ```
   class (Eq a, Show a) => Num a where
     (+), (*), (-) :: a -> a -> a
     negate, abs, signum :: a -> a
     fromInteger :: Integer -> a
   ```

   Complete as seguintes definições:

   (a) `instance (Num a) => Num (Exp a) where ...`
   (b) `instance (Num a) => Num (ExpN a) where ...`