

8ª Lista de Exercícios

Aluno(a): _____ Matrícula: _____

1. Considere o seguinte tipo de árvores binárias:

```
type Tree = Leaf | Node Tree Tree.
```

Defina `qtdFolhas :: Tree -> Int` que retorna o número de folhas em uma árvore.

2. Tal tipo de árvore é balanceada se o número de folhas na subárvores esquerda e direita de cada nó difere de no máximo um. Defina uma função `balanced :: Tree -> Bool` que decide se uma árvore é balanceada.

3. Na lógica proposicional temos fórmulas que são construídas a partir de símbolos atômicos como p , q e r e conectivos lógicos \wedge , \vee , \rightarrow e \neg . Um tipo para fórmulas da lógica proposicional pode ser definido da seguinte forma:

```
data Form = Atom Char | And Form Form | Or Form Form  
          | Imply Form Form | Not Form.
```

Por exemplo, a fórmula $(p \wedge q) \rightarrow \neg r$ pode ser representada como

```
((Atom 'p') 'And' (Atom 'q')) 'Imply' (Not (Atom 'r')).
```

Para atribuir um valor-verdade para uma fórmula precisamos saber o valor de cada atômica. Para isso, vamos definir o tipo `type Val = [(Char,Bool)]`. Defina uma função `val :: Val -> Form -> Bool` que retorna o valor-verdade de uma fórmula. Por exemplo:

```
> val [(('p',True),('q',False))] ((Atom 'p') 'And' (Not (Atom 'q')))  
True
```

4. Defina uma função `atomics :: Form -> [Char]` que retorna a lista de todas as atômicas de uma fórmula. Por exemplo:

```
> atomics (((Atom 'p') 'And' (Atom 'q')) 'Imply' (Not (Atom 'r')))  
['p','q','r']
```