Homework 3 - due 1.02.2017

Note: please comment your code, writing in simple words which part of your function does what (and how), so I can see your understanding and not just the lines of code.

1. (3 points) Check whether the formula is closed. Complete the code (insert into FSynF):

```
isEmpty :: [Variable] -> Bool
isEmpty [] = True
isEmpty(x:xs) = False

sentence :: Formula Variable -> Bool
sentence f = isEmpty (listVariable f)

merge :: [a] -> [a] -> [a]
merge xs [] = xs
merge [] ys = ys
merge (x:xs) (y:ys) = x : y : merge xs ys

listVariable :: Formula Variable -> [Variable]
listVariable (Atom s []) = []
listVariable (Atom s xs) = xs
listVariable (Eq t1 t2) = t1 : t2 : []
listVariable (Neg form) = listVariable form
listVariable (Impl f1 f2) = merge (listVariable f1) (listVariable f2)
listVariable (Equi f1 f2) = merge (listVariable f1) (listVariable f2)
listVariable (Conj []) = []
listVariable (Disj []) = []
listVariable (Conj (f:fs))= listVariable f
```

2. Semantics for Sea Battle:
    1. (5 points) check that each ship occupies adjacent squares
    2. (3 points) Implement reaction “sunk”

3. (3 points) Reimplement the semantics of propositional formulas, using [String] type instead of [(String, Bool)] and indicating truth or falsity with presence/absence.