Esercizio di Haskell 29/10/2016

Si tratta di scrivere un programma che data una formula proposizionale decida se la formula è una tautologia oppure no. Una tautologia è una formula sempre vera.

Formule proposizionali sono per esempio A && B oppure (A && B) => B o anche (A && (A =>B)) =>B.

Per rappresentare le formula proposizionali definiamo il seguente tipo Haskell :

```
data Prop = Const Bool
          | Var Char
          | Not Prop
          | And Prop Prop
          |Imply Prop Prop
```

Le tre formule date prima sono rappresentate dai seguenti valori di tipo Prop:
And A B, Imply (And A B) B e Imply ((And A (Imply A B)) B).
Per semplicità e senza perdità di generalità non viene incluso in Prop l'operatore OR e l'equivalenza.
Il programma da fare deve essere scritto in un modulo che si chiama Solutions.hs che abbia la seguente prima riga:
module Solutions(vars, ttables,solutions, Prop(..)) where

Prop(..) esporta dal modulo i costruttori di Prop.
Oltre alla definizione di Prop questo modulo deve contenere le funzioni, vars, ttables e solutions con i seguenti significati:
vars :: Prop ->[Char], data una formula proposizionale F, rappresentata da un valore del tipo Prop deve restituire la lista delle variabili proposizionali distinte che sono contenute in F. Quindi
vars Imply ((And A (Imply A B)) B) = ['A','B']. Si richiede di mantenere la prima occorrenza (scorrendo la formula da sinistra a destra) di ogni variabile proposizionale. Quindi per la precedente formula, la risposta ['B','A'] sarebbe sbagliata.
ttables :: [Char] -> [[(Char,Bool)]] data una lista L di variabili proposizionali produce la lista di tutte le sostituzioni di valori di verità True/False per le variabili di L. Quindi per ['A','B'], ttables produce [[('A',True), ('B',True)], [('A',True),('B',False)], [('A',False),('B',True)], [('A',False),('B',False]]. Ogni sottolista è un'assegnazione di valori di verità per tutte le variabili della formula, e tutte le possibili assegnazioni sono prodotte.
solutions :: Prop -> [[(Char,Bool]]-> Bool considera ogni assegnazione di valori di verità per le variabili della formula (primo parametro) e valuta se la formula è vera o no per quell'assegnazione. Deve compiere questa operazione per ogni assegnazione e restituisce True sse la formula è sempre vera, cioè sse è una tautologia.

Osservate che nel modulo fornito, le 3 funzioni da fare sono invocate per una data formula (scritta come valore del tipo Prop). Se volete provare le vostre soluzioni anche per altre formule, basta che le sostituiate alla formula usata ora.