

# README

## PRÁCTICA 03

Liprandi Cortes Rodrigo: 317275605  
Tinoco Miguel Laura Itzel: 316020189

November 11, 2020

### 1 Observaciones:

#### Funciones auxiliares de la práctica 3

- **distrDNF**: Función auxiliar que distribuye disyunciones de una fórmula  
`distrDNF :: Prop -> Prop`
- **uneAux**: Función auxiliar para unit, une una lista de clausulas con la formula de una solucion  
`uneAux :: [Clausula] -> Solucion -> Solucion`
- **elimAux**: Auxiliar para elim. Función que dada una literal y una fórmula elimina las clausulas donde aparezca esa literal.  
`elimAux :: Literal -> [Clausula] -> [Clausula]`
- **elimLiteral**: Función auxiliar que elimina una literal de una clausula. Como no queremos una clausula vacia, (conflict no checa si hay clausulas vacias), entonces si la clausula sólo tiene una literal, regresamos la misma clausula.  
`elimLiteral :: Literal -> Clausula -> Clausula`
- **redAux**: Auxiliar para red. Dada una literal y una fórmula, quita la contraria de la literal de todas las clausulas de la fórmula.  
`redAux :: Literal -> [Clausula] -> [Clausula]`

#### Funciones auxiliares de la práctica 2

- **esta**: Función auxiliar que nos dice si un elemento está en una lista.  
`esta :: Eq a => a -> [a] -> Bool`

- **equalListA**: Función auxiliar que nos dice si dos listas son iguales por sus elementos.  
`equalListA :: (Eq a) => [a] -> [a] -> Bool`
- **contieneEstado**: Función que nos dice si una lista de estados contiene un cierto estado.  
`contieneEstado :: Estado -> [Estado] -> Bool`
- **andM**: Función auxiliar que i un estado de todos los posibles estados de p satisfacen a p, entonces lo agrega a una lista.  
`modelosAux :: [[String]] -> Prop -> [Estado]`
- **orM**: Función que dada una lista de valores booleanos devuelve true si alguno de los valores es true, false si todos los valores son false, es decir hace una disyunción con todos los booleanos de la lista.  
`orM :: [Bool] -> Bool`
- **aConjunto**: Función auxiliar que elimina los duplicados es una lista.  
`aConjunto :: Eq a => [a] -> [a]`
- **conjuntoEstados**: Función auxiliar que elimina los estados duplicados de una lista de estados.  
`conjuntoEstados :: [Estado] -> [Estado]`
- **modelosAux**: Función auxiliar para modelos, si un estado de todos los posibles estados de p satisface a p, entonces lo agrega a una lista.  
`modelosAux :: [[String]] -> Prop -> [Estado]`
- **diferencia**: Función auxiliar para equivalencias, si ambas proposiciones son satisfechas por los mismo modelos entonces la diferencia de las listas de sus modelos es igual a [], si no entonces no son satisfechas por los mismos modelos y por lo tanto no son equivalentes `diferencia :: Eq a => [a] -> [a] -> [a]`
- **simplificaNeg**: Simplifica todas las negaciones de una fórmula, quita dobles negaciones.  
`simplificaNeg :: Prop -> Prop`