Práctica 02
Lógica Computacional
Universidad Nacional Autónoma de México

Dr. Miguel Carrillo Barajas
Sara Doris Montes Incin
Mauricio Esquivel Reyes

1. Lógica proposicional

Consideremos la siguiente representación de la lógica proposicional:

```
-- Tipo de datos para índices de variables
type Indice = Int

-- Tipo de datos para formulas de la PL
data PL = Top | Bot | Var Indice
        | Oneg PL | Oand PL PL
        | Oor PL PL | Oimp PL PL deriving (Eq,Show,Ord)

-- Tipo de datos para modelos
type Modelo = [Indice]
```

1.1. Elimina implicaciones

1.1.1. eliminaImp :: PL ->PL

Dada una fórmula cambiar las instancias de las implicaciones.

```
* Main> eliminaImp (Oimp (Var 1) (Var 2))
Oor (Oneg $ Var 1) (Var 2)
* Main> eliminaImp (Oand Top Bot)
Oand Top Bot
```

1.2. Forma normal de negación

1.2.1. aNNF :: PL ->PL

Dada una fórmula de la lógica proposicional se debe regresar su traducción a forma normal de negación.

```
* Main> aNNF (Oimp (Oneg $ (Oand (Var 1)(Var 2))) (Oor (Var 3)(Oneg $ Top)))
Oor (Oand (Var 1) (Var 2)) (Oor (Var 3) Bot)
* Main> aNNF (Oand (Oneg $ (Oand (Var 1)(Var 2))) (Oneg $ Oor (Var 3)(Oneg $ Top)))
Oand (Oor (Oneg (Var 1)) (Oneg (Var 2))) (Oand (Oneg (Var 3)) Top)
```

1.2.2. esNNF :: PL ->PL

Dada una fórmula de la lógica proposicional verificar si se encuentra en forma normal de negación.

```
* Main> esNNF (Oneg $ (Oand (Var 1) (Var 2)))
False
* Main> esNNF (Oor (Oneg $ Var 1)(Oneg $ Var 2))
True
```

1.3. Forma normal conjuntiva

1.3.1. aCNF :: PL ->PL

Dada una fórmula en forma normal de negación, dar su forma normal conjuntiva, tal que sean lógicamente equivalentes. Ejemplos:

```
* Main> aCNF $ Oimp (Oneg $ Oand (Var 1)(Var 2)) (Oor (Var 3)(Oneg $ Top))
Oand (Oor (Var 1) (Oor (Var 3) Bot)) (Oor (Var 2) (Oor (Var 3) Bot))
* Main> aCNF $ Oneg $ Oor (Oimp (Var 1) (Var 2)) (Oor (Var 3) (Var 4))
Oand (Oand (Var 1) (Oneg (Var 2))) (Oand (Oneg (Var 3)) (Oneg (Var 4)))
```

Punto extra por la función $cnf :: PL- > PL$ la cual toma cualquier fórmula de la lógica proposicional y regresa su forma normal conjuntiva.

1.3.2. esCNF :: PL ->PL

Dada una fórmula de la lógica proposicional revisa si se encuentra en forma normal conjuntiva. Ejemplos:

```
* Main> esCNF (Oimp (Var 1) (Oand (Var 2)(Oor (Var 3)(Var 4))))
False
* Main> esCNF (Oand (Oor (Var 2) (Oneg $ Var 2))(Oor (Var 4)(Oor (Var 5)(Var 7))))
True
```

1.4. Formal normal disyuntiva

1.4.1. esTermino :: PL ->PL

Dada una fórmula de la lógica proposicional revisa si es un término. Ejemplos:

```
* Main> esTermino $ Oand (Var 1) (Oor (Var 2) (Var 3))
False
* Main> esTermino $ Oand (Oand (Var 1) (Var 2)) (Oneg $ Var 3)
True
```

1.4.2. esDNF :: PL ->PL

Dada una fórmula de la lógica proposicional revisa si se encuentra en forma normal disyuntiva. Ejemplos:

```
* Main> esDNF (Oor (Oand (Var 2) (Oneg $ Var 2))(Oand (Var 4)(Oand (Var 5)(Var 7))))
True
* Main> esDNF (Oor (Oand (Var 2) (Oneg $ Var 2))(Oand (Var 4)(Oor (Var 5)(Var 7))))
False
```

1.5. Semántica

1.5.1. satMod :: Modelo ->PL ->Bool

Dado un modelo y una fórmula de la lógica proposicional se debe indicar si es satisfacible. Ejemplos:

```
* Main> satMod [1] (Oimp (Var 1)(Var 2))
False
* Main> satMod [1,2] (Oand (Var 1) (Oor (Var 2) (Var 3)))
True
```

1.5.2. esSatPL :: PL ->Bool

Dada una formula de la lógica proposicional indicar si es satisfacible. Ejemplos:

```
* Main> esSatPL $ Oand (Var 1) (Oneg $ Var 1)
False
* Main> esSatPL $ Oimp (Oand (Var 1)(Var 2)) (Oor (Var 3) (Oneg $ Var 4))
True
```

1.5.3. esValPL :: PL ->Bool

Dada una fórmula de la lógica proposicional indicar si es valida. Ejemplos:

```
* Main> esValPL (Oand (Var 1) (Oor (Var 2) (Var 3)))
False
* Main> esValPL (Oor (Var 1) (Oor (Oneg $ Var 2) (Var 3)))
True
```

1.5.4. Preguntas

Las respuestas deben estar en el archivo README.

1. Describe la diferencia entre una fórmula satisfacible y una valida.
2. Explica como se utiliza el algoritmo de índices complementarios para comprobar que las fórmulas en forma normal disyuntiva son satisfacibles.
3. Indica si las siguientes fórmulas son satisfacibles y explica como se llego a dicha conclusión:
   - $(p \land q \land s \to p) \land (p \land r \to p) \land (p \land s \to s)$
   - $(p \land q \land s \to \bot) \land (q \land r \to \bot) \land (s \to \bot)$

1.6. Deducción Natural

1.6.1. Reglas faltantes

En el archivo DeduccionNatural.hs implementar las siguientes funciones:

1. checkIdis1
2. checkIdis2
3. checkE2neg

1.6.2. Ejercicios

Dar la deducción de los siguientes ejercicios:

1. $p, \neg\neg(q \land r) \vdash \neg\neg p \land r$
2. $p \land q \to r \vdash p \to (q \to r)$
3. $q \to r \vdash p \lor q \to p \lor r$
4. $p \land (q \lor r) \vdash (p \land q) \lor (p \land r)$
5. $(p \land q) \lor (p \land r) \vdash p \land (q \lor r)$