# Practica 2

*Domingo 1 de Marzo, 2020*

*Escamilla Soto Cristopher Alejandro*
*Montiel Manriquez Ricardo*

## Procedimiento:

La practica se realizo mediante la implementación de las siguientes funciones:

**1.-** variables: Es una función que nos regresa una lista con las variables proposicionales dentro de ella y ocupamos otra función auxiliar de nombre repetidos para que elimine repetidos dentro de una lista.

**2.-** conjPotencia: Es la mera definición del conjunto potencia en la teoría de conjuntos y utilizamos listas por comprensión para definir de forma mas fácil un conjunto de elementos.

**3.-** interpretaciones: Es la definición de cada operador lógico, se utilizaron equivalencias lógicas para poder simular al operador de implicación y del syss.

**4.-** estadosPosibles: Es la función de conjunto potencia aplicada a una cadena de variables, así obtenemos las combinaciones posibles de estados y aparte eliminamos los elementos repetidos.

**5.-** tautologia: Usamos la definición de comparar para que todos los modelos sean iguales a las variables proposicionales, eso nos dirá que si están todas las variables entonces es una tautología, lo que hacemos con ayuda de una función auxiliar de nombre elementosIguales.

**6.-** contradiccion: Aplicamos la función tautología en caso de cumplir los requisitos entonces es una contradicción.

**7.-** esModelo: No es mas que una interpretación de cuando nos dan un modelo en especifico y nos da su interpretación por supuesto.

**8.-** modelos: Son todos los posibles modelos, utilizamos listas por comprensión para definirlos, por supuesto son todos los conjuntos tales que pertenecen al conjunto de estados posibles y a la vez a alguna interpretación que es verdadera.

**9.-** esValida: Es un función booleana sencilla pues solo tenemos que saber si los modelos son iguales al conjunto potencia de las variables de una proposición.

**10.-** esInsatisfacible: Si una proposición es insatisfacible implica que no tiene modelos

**11.-** esSatisfacible: Si una proposición es satisfacible implica que tiene algun modelo.

**12.-** Funciones Auxiliares:

- repetidos:Usamos recurción y la definición de una lista para poder eliminar los elementos repetidos.

- elementosIguales: Compara los elementos de dos listas y nos dice si el elemento de la primera esta contenida en la segunda.

## Forma de Ejecución del Programa:

Primero compilamos el programa como: ghci Practica2.hs

Para la ejecución de las funciones:

**1.-** Prelude> variables Prop
Prelude> variables (Impl (Conj p q) p)

**2.-** Prelude> conjPotencia []
Prelude> conjPotencia [1,2]

**3.-** Prelude> interpretacion Prop Prop
Prelude> interpretacion (Conj (Var "q") (Disy (Var "r") (Var "p"))) ["p"]

**4.-** Prelude> estadosPosibles Prop
Prelude> estadosPosibles (Disy (Var "q") (Conj (Var "r") (Var "q")))

**5.-** Prelude> tautologia Prop
Prelude> tautologia (Disy (Var "p") (Neg (Var "p")))

**6.-** Prelude> contradiccion Prop
Prelude> contradiccion (Disy (Var "p") (Neg (Var "p")))

**7.-** Prelude> esModelo Estado Prop
Prelude> esModelo ["r"] (Conj (Disy p q) (Disy (Neg q) r))

**8.-** Prelude> modelos Prop
Prelude> modelos (Conj (Disy p q) (Disy (Neg q) r))

**9.-** Prelude> esValida Prop
Prelude> esValida (Impl p p)

**10.-** Prelude> esInsatisfacible Prop
Prelude> esInsatisfacible (Conj p (Neg p))

**11.-** Prelude> esSatisfacible Prop
Prelude> esSatisfacible (Conj p (Neg p))

Los ejemplos de ejecución anteriores fueron tomados del PDF mandado por el ayudante de Laboratorio.

## Conclusiones:

La lógica proposicional se puede definir y así poder hacer el trabajo mas fácil pues también podemos modelar si una proposición es satisfacible o no, podemos decir que ahora podemos interpretar de forma mas eficiente.

## Observaciones:

Cambiamos
variables :: Prop -> [Estado]
por :
variables :: Prop -> [String]
Ya que en los comentarios y el PDF esta definida de esa forma

En varias formulas comentadas para probar el programa falta agregar paréntesis al inicio de la formula para que pueda funcionar correctamente.