Universidad de Carabobo - Facultad de Ingenieria - Direccion de Postgrado
Programa: Maestria Matematica y Computacion - **Asignatura**: Introduccion al Calculo
Alumno: Ronald Medina - **Cedula**: V-16291029
Lapso: 03 - 2022 - **Fecha**: 14 - 03 - 2023
Titulo: Asignacion - **Parte 8, 9, 10)** Logica Proposicional

```
> with(Logic):
```

Ejercicio 8.1 $po1 = (\neg p \vee q) \wedge \neg q \Rightarrow \neg p$

Se crea la proposicion y se guarda en la variable *po1*

```
> po1 := Import(((¬p ∨ q) ∧¬q) ⇒¬p);
```

$$po1 := Logic:\text{-}\&implies((\&not(p)\ \&or\ q)\ \&and\ \&not(q), \&not(p)) \quad (1.1)$$

Se usa la funcion que verifica, si la proposicion es una tautologia

```
> Tautology(po1);
```

$$true \quad (1.2)$$

Como el resultado es verdadero, se concluye que po1 es una tautologia

```
> Contradiction(po1);
```

$$false \quad (1.3)$$

Como adicional, al ejecutar po1 en la funcion Contradiccion, nos arroja valor falso

Ejercicio 8.2 $po2 = ((p \wedge q) \vee q) \wedge \neg q$

Se crea la proposicion y se guarda en la variable *po2*

```
> po2 := Import(((p ∧ q) ∨ q) ∧¬q);
```

$$po2 := (p\ \&and\ q\ \&or\ q)\ \&and\ \&not(q) \quad (2.1)$$

Se usa la funcion que verifica, si la proposicion es una tautologia

```
> Tautology(po2);
```

$$false \quad (2.2)$$

Como el resultado es falso, se utiliza la funcion logica Contradiccion, como siguiente paso

```
> Contradiction(po2);
```

$$true \quad (2.3)$$

Como el resultado es verdadero, se concluye que po2 es una contradiccion

Ejercicio 9.1 $p \Rightarrow q \equiv \neg p \vee q$

Se escribe el lado izquierdo de la equivalencia y se guarda en la variable po3

```
> po3 := Import(p ⇒ q);
```

$$po3 := Logic:\text{-}\&implies(p, q) \quad (3.1)$$

Se escribe el lado derecho de la equivalencia y se guarda en la variable po4

```
> po4 := Import(¬p ∨ q);
```

$$po4 := \&not(p)\ \&or\ q \quad (3.2)$$

Se utliza la funcion que verifica la equivalencia de expresiones logicas

```
> Equivalent(po3, po4);
```

$$true \quad (3.3)$$

Al dar como resultado verdadero, se verifica la equivalencia de la expresion. En este caso, por definicion, una implicacion, es equivalente a la negacion del antecedente disyuncion con el consecuente.

Ejercicio 9.2 $\neg(p \vee q) \equiv \neg p \wedge \neg q$

Se escribe el lado izquierdo de la equivalencia y se guarda en la variable po5

```
> po5 := Import(¬(p ∨ q));
```

$$po5 := \&not(p\ \&or\ q) \quad (4.1)$$

Se escribe el lado derecho de la equivalencia y se guarda en la variable po6

> $po6 := Import(\neg p \wedge \neg q);$

$$po6 := \&not(p)\ \&and\ \&not(q) \quad \textbf{(4.2)}$$

Se utliza la funcion que verifica la equivalencia de expresiones logicas

> $Equivalent(po5, po6);$

$$true \quad \textbf{(4.3)}$$

Al dar como resultado verdadero, se verifica la equivalencia de la expresion. Ley de DeMorgan.

Ejercicio 10.1 $\neg((p \Rightarrow \neg q) \vee \neg q) \Rightarrow ((\neg p \Rightarrow (\neg p \Rightarrow q)) \wedge ((\neg p \Rightarrow q) \Rightarrow \neg p))$

Se divide la proposicion, en dos partes, lado izquierdo y lado derecho, lado referidos al simbolo de implicacion.

Se escribe el lado izquierdo de la proposicion y se guarda en la variable li.

> $li := Import(\neg((p \Rightarrow \neg q) \vee \neg q));$

$$li := \&not(Logic:-\&implies(p, \&not(q))\ \&or\ \&not(q)) \quad \textbf{(5.1)}$$

Se escribe el lado derecho de la proposicion y se guarda en la variable ld.

> $ld := Import((\neg p \Rightarrow (\neg p \Rightarrow q)) \wedge ((\neg p \Rightarrow q) \Rightarrow \neg p));$

$$ld := Logic:-\&implies(\&not(p), Logic:-\&implies(\&not(p), q))\ \&and\ Logic:-\&implies(Logic:-\&implies(\&not(p), q), \&not(p)) \quad \textbf{(5.2)}$$

Se conjforma la proposicion del ejercio, a traves del simbolo de implicacion y se guarda la expresion logica en la variable *po7*.

> $po7 := Import(li \Rightarrow ld);$

$$po7 := Logic:-\&implies(\&not(Logic:-\&implies(p, \&not(q)))\ \&or\ \&not(q)), Logic:-\&implies(\&not(p), Logic:-\&implies(\&not(p), q))\ \&and\ Logic:-\&implies(Logic:-\&implies(\&not(p), q), \&not(p))) \quad \textbf{(5.3)}$$

Se utiliza la funcion que permite simplificar la expresion logica.

> $BooleanSimplify(po7);$

$$\&not(p)\ \&or\ \&not(q) \quad \textbf{(5.4)}$$

Se genera el resultado de la simplificacion de la proposicion *po7*.