

## TRABAJO PRÁCTICO N° 2 - LÓGICA PROPOSICIONAL

### INTRODUCCIÓN:

Los objetos de estudio de la Lógica Proposicional son las proposiciones y los razonamientos proposicionales, proveyendo los métodos adecuados para establecer la verdad o falsedad de una proposición dada.

La **Proposición** es un enunciado declarativo que afirma o niega algo, tiene la propiedad de ser verdadero o falso. La característica principal de esta, radica en el hecho que, si el estado de cosas que la proposición enuncia o declara suceded tal y como está afirmado, la misma es **verdadera**, pero, si suceded de forma diferente al declarado por ella, la proposición es **falsa**.

La **Abstracción** e **Interpretación** son dos procesos inversos entre sí, constituyen rasgos característicos de la actividad inteligente del hombre. La primera implica siempre una generalización, en tanto que la interpretación lleva siempre a una particularización.

**Abstracción:** proceso que va del lenguaje descriptivo al lenguaje lógico; es una traducción consistente en una generalización cada vez mayor constituyendo un proceso de simplificación (Lenguaje Coloquial → Lenguaje Simbólico).

**Interpretación:** proceso por el cual pasamos de un lenguaje abstracto a un lenguaje descriptivo. Toda interpretación implica una particularización (Lenguaje Simbólico → Lenguaje coloquial)

Álgebra de Boole: constituye una formalización apropiada para representar información digital y proporciona un modelo matemático para determinar la respuesta de los circuitos lógicos, independientemente del tipo de dispositivos que constituyen dichos circuitos (transistores, relés, etc)

Algunas de las aplicaciones del Álgebra Booleana en conmutación son las siguientes:

Expresión algebraica de los requerimientos que debe cumplir un circuito lógico

Análisis y síntesis de circuitos combinacionales

Comparación de distintas realizaciones circuitales

Minimización del número de cables y dispositivos en los circuitos

Codificación de información digital.

Herramienta auxiliar para analizar y sintetizar circuitos secuenciales.

**OBJETIVOS:** Que el alumno reconozca una proposición lógica y opere con ella. Construya tablas de verdad y haga demostraciones con las mismas. Entienda y realice circuitos lógicos. Use funciones proposicionales y cuantificadores.

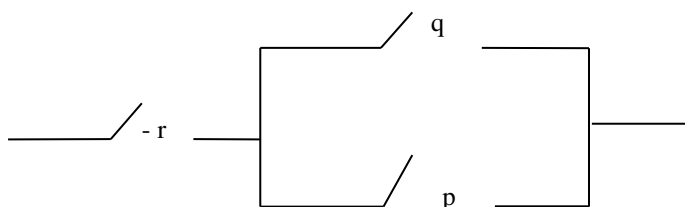
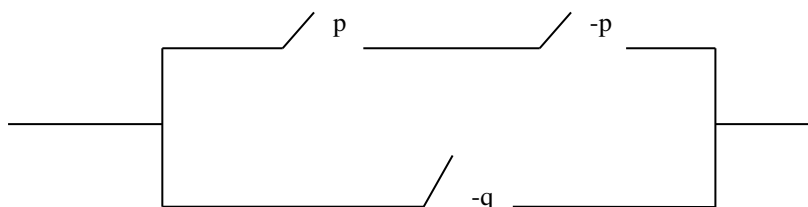
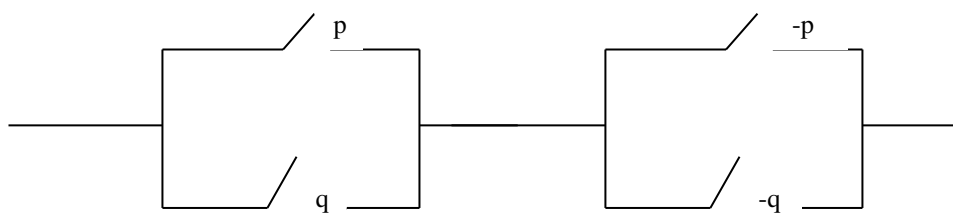
1.- Determinar en cada caso si los siguientes pares de enunciados son o no equivalentes:

- |   |   |
|---|---|
| a) $\neg(p \wedge q)$                           | con $\neg p \vee \neg q$                  |
| b) $p \vee (q \vee r)$                          | con $(p \vee q) \vee r$                   |
| c) $(p \rightarrow q) \vee r$                   | con $p \rightarrow (q \vee r)$            |
| d) $(p \rightarrow q) \wedge (q \rightarrow r)$ | con $p \rightarrow r$                     |
| e) $\neg(p \rightarrow q)$                      | con $(\neg p) \rightarrow (\neg q)$       |
| f) $p \rightarrow (q \leftrightarrow r)$        | con $(p \rightarrow q) \leftrightarrow r$ |

2.- Dadas las siguientes expresiones simbólicas, dibujar el circuito asociado:

- a)  $(p \wedge \neg q) \vee (\neg p \wedge q)$   
 b)  $[p \wedge (q \vee r)] \wedge (p \vee \neg p)$   
 c)  $(p \wedge \neg p) \wedge (q \vee \neg q)$   
 d)  $(p \wedge q) \wedge (\neg p \vee \neg q)$   
 e)  $(p \wedge \neg q) \vee (\neg p \wedge q) \vee (\neg p \wedge \neg q)$   
 f)  $(p \vee q \vee r) \wedge (\neg p \vee \neg q)$   
 g)  $p \wedge (\neg p \vee q) \wedge (r \vee q)$

3.- Dados los siguientes circuitos obtener en forma simbólica el enunciado asociado:



## Álgebra de Boole

George Boole (1815-1864) fue el creador de un sistema algebraico para el tratamiento sistemático de la lógica, que hoy utilizamos como soporte lógico de las técnicas digitales, el álgebra de conjuntos y la teoría de probabilidades. Desarrollada en 1847 en su obra “Análisis Matemático de la Lógica) y en 1854 en “Investigación de las Leyes del Pensamiento”.

El Álgebra de Boole, constituye una formalización apropiada para representar información digital y proporciona un modelo matemático para determinar la respuesta de los circuitos lógicos, independientemente del tipo de dispositivos que constituyen dichos circuitos (transistores, relés, etc)







Algunas de las aplicaciones del Álgebra Booleana en conmutación son las siguientes:

- Expresión algebraica de los requerimientos que debe cumplir un circuito lógico
- Análisis y síntesis de circuitos combinacionales
- Comparación de distintas realizaciones circuitales
- Minimización del número de cables y dispositivos en los circuitos
- Codificación de información digital.
- Herramienta auxiliar para analizar y sintetizar circuitos secuenciales.

En 1938 Shannon aplica el Álgebra de Boole al análisis de los circuitos de conmutación

### Elementos del Álgebra de Boole:

- ✚ **Variables lógicas:** variables binarias que solo pueden tener un valor entre dos distintos, por convención las llamaremos 0 (cero): proposición falsa y 1 (uno): Proposición verdadera; A, B, C, ....{0, 1}
- ✚ **Funciones lógicas o booleanas:** son funciones de n variables lógicas que toman valores en el conjunto 0, 1;  $F(A_1, A_2, A_3, \dots, A_n) = \{0,1\}$
- ✚ **Función inversa o complementación:** permite invertir cualquier secuencia de ceros y unos ( $0 \rightarrow 1$  ;  $1 \rightarrow 0$ ), tal como lo hace la compuerta NOT (único operador que posee una sola entrada).
- ✚ **Función intersección o Producto Lógico:** esta operación se materializa mediante la compuerta AND
- ✚ **Función reunión o Suma Lógica:** operación que se materializa con la compuerta OR

Name	Graphical Symbol	Algebraic Function	Truth Table															
AND		$F = A \cdot B$ or $F = AB$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	0	1	0	0	1	1	1
A	B	F																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR		$F = A + B$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	1
A	B	F																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
NOT		$F = \overline{A}$ or $F = A'$	<table><tr><th>A</th><th>F</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	F	0	1	1	0									
A	F																	
0	1																	
1	0																	
NAND		$F = \overline{AB}$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	1	0	1	1	1	0	1	1	1	0
A	B	F																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR		$F = \overline{A + B}$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	1	0	1	0	1	0	0	1	1	0
A	B	F																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
XOR		$F = A \oplus B$	<table><tr><th>A</th><th>B</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	F	0	0	0	0	1	1	1	0	1	1	1	0
A	B	F																
0	0	0																
0	1	1																
1	0	1																
1	1	0																

### Basic Logic Gates

**Operaciones en el álgebra de Boole:** definidas las operaciones  $+$  (suma) y  $*$  (multiplicación) y la operación unitaria  $-$ ,  $'$ ; en el álgebra de Boole se cumplen los siguientes Axiomas y Teoremas:

Axiomas del Álgebra de Boole	
<i>Leyes Conmutativas</i>	
$a + b = b + a$	$a * b = b * a$
<i>Leyes Distributivas</i>	
$a + (b * c) = (a + b) * (a + c)$	$a * (b + c) = (a * b) + (a * c)$
<i>Leyes de Identidad</i>	
$a + 0 = a$	$a * 1 = a$
<i>Leyes de Complemento</i>	
$a + a' = 1$	$a * a' = 0$
<i>Leyes de Idempotencia</i>	
$a + a = a$	$a * a = a$
<i>Leyes de Acotamiento</i>	
$a + 1 = 1$	$a * 0 = 0$
<i>Leyes de Absorción</i>	
$a + (a * b) = a$	$a * (a + b) = a$
<i>Leyes Asociativas</i>	
$(a + b) + c = a + (b + c)$	$(a * b) * c = a * (b * c)$
<i>Unicidad del Complemento</i>	
Si $a + x = 1$ y $a * x = 0$ , entonces $x = a'$	
<i>Ley de Involución</i>	
$(a')' = a$	
<i>Teoremas</i>	
$0' = 1$	$1' = 0$
<i>Leyes de DeMorgan</i>	
$(a + b)' = a' * b'$	$(a * b)' = a' + b'$

Nota: A no ser que se indique entre paréntesis el complemento “-”, “’” tiene precedencia sobre el producto \* y este tiene precedencia sobre la suma +

Fuente: SISTEMAS NUMÉRICOS Y ÁLGEBRA DE BOOLE- Ing. Sebastián Vicente Martín

4.- ¿Cómo procesaría una compuerta NOT las siguientes sucesiones?

(a) 101101      (b) 110100011      (c) 0100101

5.- Dados A = 1101011; B = 1000110; C = 1100100, Hallar el valor de las siguientes operaciones

a)  $A + B + C$       b)  $A.C$       c)  $\overline{B + C}$