Álgebra de Boole

El álgebra booleana fue desarrollada por George Boole y en su libro *An Investigation of the Laws of Thought*, publicado en 1854, muestra las herramientas para que las proposiciones lógicas sean manipuladas en forma algebraica. Debido al carácter abstracto de sus principios no tuvo una aplicación directa sino hasta 1938 en que la compañía de teléfonos Bell de Estados Unidos la utilizó para realizar un análisis de los circuitos de su red telefónica. En ese mismo año Claude E. Shannon, entonces estudiante de postgrado del Instituto Tecnológico de Massachussets, a partir del álgebra de Boole creó la llamada álgebra de conmutación para representar las propiedades de conmutación eléctrica biestables, demostrando con esto que el álgebra booleana se adapta perfectamente al diseño y representación de circuitos lógicos de control basados en relés e interruptores.

Se define un Algebra de Boole (A,+,*) como todo conjunto de elementos capaces de adoptar dos valores, designados por 1 y 0, y entre los cuales están definidas dos operaciones: suma lógica (+) y producto lógico (*). Cada uno de dichos elementos recibe el nombre de variable lógica o binaria.

Postulados del Ágebra de Boole

El álgebra de Boole cumple los siguientes postulados:

1.- <u>Propiedad conmutativa</u>: Dadas dos variables lógicas a,b $\in$ A (A= Algebra de Boole), se cumple:

$$a+b = b+a \quad y \quad a*b = b*a$$

2.- <u>Propiedad distributiva</u>: Dadas tres variables lógicas a,b,c $\in$ A se cumple:

$$a*(b+c) = a*b + a*c \quad y \quad a+(b*c) = (a+b)*(a+c)$$

3.- <u>Elemento neutro</u>: Existe un elemento neutro para cada una de las dos operaciones, designados por 0 para (+), 1 para (*). Así, dada la variable a $\in$ A, dichos elementos cumplen las siguientes condiciones:

$$a+0 = a \quad y \quad a*1 = a$$

4.- <u>Elemento simétrico</u> (complementario o inverso): Existe, para cada variable lógica a $\in$ A, su complementaria o inversa a´, definida para ambas operaciones (+) y (*), y tal que siempre se cumple,

$$a+a' = 1 \quad y \quad a*a' = 0$$

Teoremas del Algebra de Boole

Son 7 los Teoremas fundamentales de un Algebra de Boole, y se demuestran todos a partir de los cuatro postulados anteriores.

1.- <u>Ley de Dualidad</u>: Cualquier expresión o identidad en un Algebra de Boole tiene su expresión dual que se obtiene intercambiando (+) por (*) y 0 por 1.

Por ejemplo:

a+0 = a	a*1 = a
a+a´= 1	a*a´= 0
a*(b+c) = (a*b) + (a*c)	a+(b*c) = (a+b) * (a+c)

2.- <u>Ley de Acotación o elemento absorbente:</u> se cumplen, para toda variable a $\in$ A, las dos condiciones siguientes,

$$a+1 = 1 \quad y \quad a*0 = 0$$

Para demostrar esto:
1 = a+a´ = a + (a´ * 1) = (a+a´) * (a+1) = 1 * (a+1) = a+1
0 = a* a´= a * (a´+ 0) = (a*a´) + (a*0) = 0 + (a*0) = a*0

3.- Ley de Idempotencia: Se cumple, para toda variable a $\in$ A, lo siguiente,
a+a =a y a*a = a

Para demostrarlo:
a = a+0 = a + (a*a´) = (a+a) * (a+a´) = (a+a) *1 = a+a
a = a*1 = a * (a+a´) = (a*a) + (a*a´) = (a*a)+0 = a*a

4.- Ley de Absorción: Para todo par de variables a,b $\in$ A, se cumple,
a+(a*b) = a y a*(a+b) = a
Demostración:
a = 1*a = (1+b) * a = (1*a) + (b*a) = a +(a*b)
a = 0+a = (0*b) + a = (0+a) * (b+a) = a *(a+b)

4.1- Ley de Absorción parcial: Para todo par de variables a,b $\in$ A, se cumple,
a+(a´ *b) = a+b y a*(a´+b) = a*b

Demostración:
a+(a´ *b) = (a+a´) * (a+b) * = 1 + (a+b) = a+b
a*(a´+b) = (a*a´) + (a*b) * = 0 + (a*b) = a*b

5.- Ley asociativa: dadas tres varibles lógicas a,b,c $\in$ A, se cumple,
a+(b+c) = (a+b)+c y a*(b*c) = (a*b)*c

Demostración con la tabla de verdad

c b a	b+c	a+(b+c)	a+b	(a+b)+c
0 0 0	0	0	0	0
0 0 1	1	1	0	1
0 1 0	1	1	1	1
0 1 1	1	1	1	1
1 0 0	0	1	1	1
1 0 1	1	1	1	1
1 1 0	1	1	1	1
1 1 1	1	1	1	1

6.- Ley de la doble negación o Ley Involutiva: Para toda variable a $\in$ A, se cumple,

(a´)´ = a

7.- Leyes de Morgan: Para todo par de variables lógicas a,b $\in$A, se cumple,
(a+b)´ = a´*b´ y (a*b)´ =a´+b´

Estas dos leyes son muy importantes, ya que permiten pasar de expresiones en sumas lógicas a expresiones equivalentes en productos lógicos.

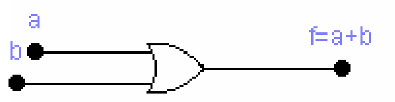
Puertas Lógicas

En un primer momento el álgebra de Boole se utilizó en conmutadores, cuya realización práctica se llevó a cabo mediante relés para construir los primeros ordenadores de la historia.

El avance de la tecnología electrónica ha llevado a la realización física de otros elementos, las "puertas lógicas", que configuran también un álgebra de Boole. En este caso las variables binarias son señales eléctricas de tensión alta (H≡1) o baja (L≡0).

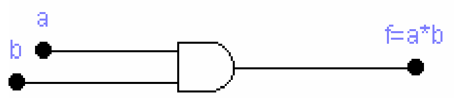
Las puertas básicas son las correspondientes a las tres operaciones lógicas básicas: suma, producto y complementación.

1.- Puerta **OR**: Salida 1 si hay algún 1 en las entradas.



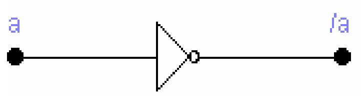
a	b	a+b
0	0	0
0	1	1
1	0	1
1	1	1

2.- Puerta **AND**: Salida 1 si todas las entradas son 1.



a	b	a*b
0	0	0
0	1	0
1	0	0
1	1	1

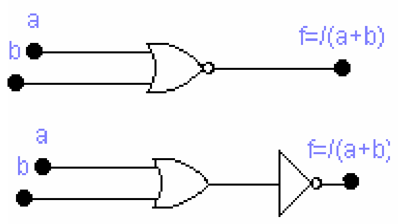
3.- Puerta **NOT**:



a	~a
0	1
1	0

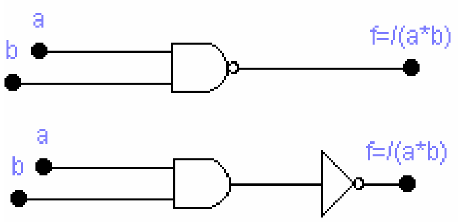
Mediante combinaciones de estas se obtienen otras dos de muy amplio uso:

4.- Puerta **NOR**:



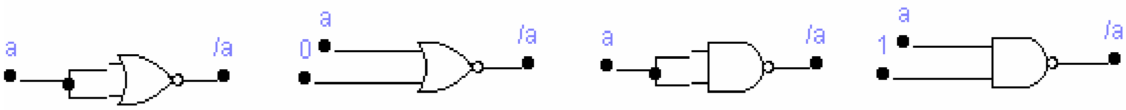
a	b	~(a+b)
0	0	1
0	1	0
1	0	0
1	1	0

5.- Puerta **NAND**:



a	b	~(a*b)
0	0	1
0	1	1
1	0	1
1	1	0

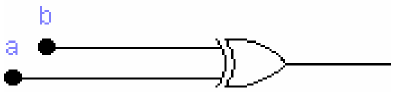
Las puertas más utilizadas son la NOT, NOR y NAND. A su vez, las puertas NOR y NAND pueden funcionar como puertas inversoras, conectando sus entradas apropiadamente:



Además existen dos tipos de compuertas muy utilizadas

6.- Función **OR-EXCLUSIVA**

Es la función que verifica la siguiente tabla de verdad:
Esta función está construida en la práctica en forma de una puerta



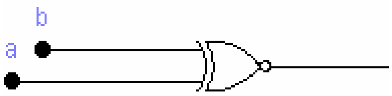
a	b	a⊕b
0	0	0
0	1	1
1	0	1
1	1	0

lógica especial (puerta OR-EXCLUSIVA), cuyo símbolo se muestra en la figura (⊕).
a⊕b = ~a.b+a.~b

La función OR-Exclusiva vale 1 cuando hay un número impar de variables de entrada a 1, y vale 0 cuando dicho número es par. Este criterio es aplicable tanto a puertas OR-Exclusiva de dos entradas como a puertas con mayor número de entradas.

7.- Función **NOR-EXCLUSIVA**

Es la inversa de la OR-Exclusiva, y su tabla de verdad y símbolo son los siguientes:



a	b	~(α⊕b)
0	0	1
0	1	0
1	0	0
1	1	1

Esta función vale 1 cuando hay un número par de "unos" en las entradas (considerando el cero par). No suele encontrarse como puerta lógica individual, sino que se obtiene con una OR-Exclusiva más una puerta inversora.

~(a⊕b) = ~a.~b+a.b

Funciones o expresiones Booleanas

El álgebra booleana trabaja con señales binarias. Al mismo tiempo una gran cantidad de sistemas de control, también conocidos como digitales, usan señales binarias y éstas son un verdadero o un falso (una señal de tensión alta o baja) que proviene de dispositivos que mandan la información al circuito de control que lleva a cabo la evaluación para obtener un valor que indicará si se realiza o no una determinada actividad, como encender un foco, arrancar un equipo de ventilación en un cine o ejecutar una operación matemática en una computadora etc.
Las funciones o expresiones booleanas poseen las siguientes propiedades:

- Están compuestas de literales A, B, C, D…y sus complementos y cada una de ellas representa la señal de un dispositivo.
Un ejemplo es **F = A´BD + AB´CD**

- El valor de las señales o de la función sólo puede ser 0 o 1, falso o verdadero.

- Además de literales, en la expresión booleana se puede tener el valor de 0 o 1.
Por ejemplo: **F = ABD1 + AB´CD + 0**

- Las literales de las expresiones booleanas pueden estar conectadas por medio de los operadores lógicos And, Or y Not.

- Además de las operaciones básicas, también es posible aplicar la ley de De Morgan de forma semejante a como se aplica en teoría de conjuntos. El siguiente ejemplo muestra la aplicación de esta propiedad:

$\mathbf{(ABCD)'} = \mathbf{A'} + \mathbf{B'} + \mathbf{C'} + \mathbf{D'}$ y también $\mathbf{(A + B + C + D)'} = \mathbf{A'\ B'\ C'\ D'}$

- Un conjunto de literales que están relacionadas mediante la operación AND se denominan términos producto y si se relacionan mediante la operación OR son denominados términos suma

Por ejemplo: Término producto **ABD** término suma **B+C+D**

Formas Canónicas de una Función Booleana

Una función booleana puede representarse por sus 2 formas canónicas

Primera forma canónica

Es la expresión de una función que contiene solamente suma de productos canónicos. Un producto canónico es aquel que contiene todas las variables de las que depende la función. Este tipo de términos se denominan Mintérminos y cada término representa un **1** en el valor de salida de la función. Se representan con la variable si corresponde a un **1** y con la variable complementada si corresponde un **0.**
Por ejemplo, tomemos la salida de la siguiente función de tres variables:

X	Y	Z	F = X + Y´ · Z	Mintérmino
0	0	0	0	
0	0	1	1	$X' \cdot Y' \cdot Z = m_1$
0	1	0	0	
0	1	1	0	
1	0	0	1	$X \cdot Y' \cdot Z' = m_4$
1	0	1	1	$X \cdot Y' \cdot Z = m_5$
1	1	0	1	$X \cdot Y \cdot Z' = m_6$
1	1	1	1	$X \cdot Y \cdot Z = m_7$

La función tiene un 1 a la salida en los mintérminos 1,4,5,6 y 7 es decir la función se puede expresar como

$F = \sum m(1, 4,5,6,7)$

$F = (X' \cdot Y' \cdot Z)+(X \cdot Y' \cdot Z')+(X \cdot Y' \cdot Z)+(X \cdot Y \cdot Z')+$
$+(X \cdot Y \cdot Z)$

Segunda forma canónica

Es la expresión de una función que contiene solamente el producto de sumas canónicas. Una suma canónica es aquella que contiene todas las variables de las que depende la función. Este tipo de términos se denominan Maxtérminos y cada término representa un **0** en el valor de salida de la función. Se representan con la variable si corresponde a un **0** y con la variable complementada si corresponde un **1.**

Tomemos la salida de la misma función de tres variables del ejemplo anterior

X	Y	Z	$F = X + Y' \cdot Z$	Maxtérmino
0	0	0	0	$X + Y + Z = M_0$
0	0	1	1	
0	1	0	0	$X + Y' + Z = M_2$
0	1	1	0	$X + Y' + Z' = M_3$
1	0	0	1	
1	0	1	1	
1	1	0	1	
1	1	1	1	

La función tiene un 0 a la salida en los maxtérminos 0, 2 y 3 es decir la función se puede expresar como

$F = \prod M(0, 2,3)$

$F = (X + Y + Z).(X + Y' + Z).(X + Y' + Z')$

En resumen para tres variables se representan los mintérminos y los maxtérminos en la siguiente tabla:

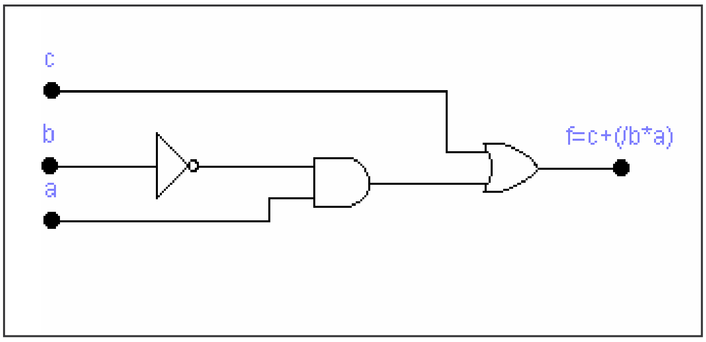
Valor decimal	X Y Z	Mintérmino	Maxtérmino
0	0 0 0	$X' \cdot Y' \cdot Z' = m_0$	$X + Y + Z = M_0$
1	0 0 1	$X' \cdot Y' \cdot Z = m_1$	$X + Y + Z' = M_1$
2	0 1 0	$X' \cdot Y \cdot Z' = m_2$	$X + Y' + Z = M_2$
3	0 1 1	$X' \cdot Y \cdot Z = m_3$	$X + Y' + Z' = M_3$
4	1 0 0	$X \cdot Y' \cdot Z' = m_4$	$X' + Y + Z = M_4$
5	1 0 1	$X \cdot Y' \cdot Z = m_5$	$X' + Y + Z' = M_5$
6	1 1 0	$X \cdot Y \cdot Z' = m_6$	$X' + Y' + Z = M_6$
7	1 1 1	$X \cdot Y \cdot Z = m_7$	$X' + Y' + Z' = M_7$

Implementación de Funciones Booleanas con puertas lógicas

Una función booleana puede representarse por una expresión algebraica de literales, pero no es única. Dos maneras de representarlas son las formas canónicas y a partir de ellas se puede obtener la función óptima (mínima cantidad de puertas lógicas) simplificando esas formas canónicas, ya sea por medio de los postulados y teoremas del **Álgebra de Boole** o por medio de métodos gráficos como son los **mapas o diagramas de Karnaugh**.

Un ejemplo sería representar con puertas lógicas la operación compuesta o función booleana:

F = c + (~b*a)



<u>Simbología</u>:

En una puerta lógica, una entrada con un círculo significa entrada invertida (a través de un inversor), e igualmente, una salida con círculo significará normalmente salida a través de un inversor.

Mapas de Karnaugh

Las variables se colocan en una tabla, de forma que cualquier columna (fila) contigua, solo difiera de la anterior en una sola variable. Las variables se consideran ordenadas, con la más significativa a la izquierda. La primera y última columna (fila) también se consideran contiguas.

Simplificación con Mintérminos

Las tablas para 2, 3 y 4 variables son de la forma siguiente:

B \ A	0	1
0	m_0	m_1
1	m_2	m_3

C \ BA	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

DC \ BA	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

La distribución de los minitérminos, según los valores de las variables, siguen la pauta que se puede observar en las figuras, obedecen al código Gray donde solo una variable cambia de estado

El principio de simplificación se basa en una de las leyes del álgebra de Boole:

$$B\,A + B\,\overline{A} = B\,(A + \overline{A}) = B$$

B \ A	0	1
0	0	0
1	1	1

El algoritmo de simplificación, con minitérminos, es el siguiente:

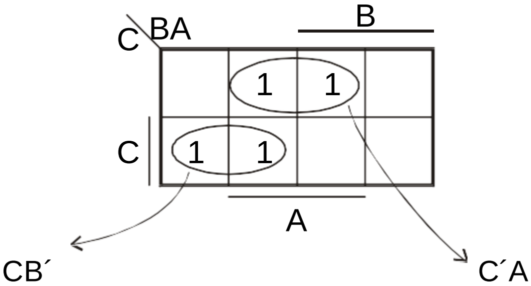
- Dada una función en forma canónica, o mediante la tabla de verdad, marcamos con un "1" las casillas en que las que el correspondiente minitérmino vale "1".
- Identificamos y señalamos grupos de 16, 8, 4, o 2 "1" adyacentes, lo mayor posibles, aun- que algún "1" ya pertenezca a otro grupo. Un "1" puede pertenecer a más de un grupo.
- Cada grupo dará lugar a un producto en que solo figuran las variables comunes a dicho grupo. Un grupo de 2 "unos" simplificará la variable que aparezca en forma normal y ne- gada. Un grupo de 4 "unos" simplificará dos variables, y uno de 8, eliminará 3 variables.
- La función será la suma de las expresiones de cada grupo, más la de los "1" aislados que no se puedan simplificar.

Ejemplo: **F = $\sum$ m(1,3,4,5)** sumatoria de mintérminos, donde la función presenta **1** a la salida

Primera forma canónica $\boldsymbol{F = \overline{C}\,\overline{B}A + \overline{C}BA + C\overline{B}\,\overline{A} + C\overline{B}A}$

Luego de la simplificación

$$\boldsymbol{F = C\overline{B} + \overline{C}A}$$



Como obtener una forma Canónica a partir de una expresión Booleana

Si se tiene una sumatoria de términos producto, para obtener la primera forma canónica, a los términos que les falta alguna variable se lo multiplica por la suma de la variable y su complemento, ya que esa suma es un 1 lógico y multiplicar por 1 no modifica la expresión. Si se forman más de un término idéntico solo se toma uno y los otros se eliminan.

Ejemplo: Una expresión F en tres variables A; B y C, es decir $F(A;B;C) = AB + BC + C$

Hallar la primera forma canónica.

$F(A;B;C) = AB\mathbf{.(C+C')} + BC\mathbf{.(A+A')} + C\mathbf{.(A+A').(B+B')}$

$F(A;B;C) = AB\mathbf{C} + AB\mathbf{C'} + \cancel{BC\mathbf{A}} + BC\mathbf{A'} + \cancel{C\mathbf{AB}} + C\mathbf{AB'} + \cancel{C\mathbf{A'B}} + C\mathbf{A'B'}$

Se eliminan los términos que se repiten quedando solo uno de ellos

La Primera forma Canónica de la expresión será:

$\mathbf{F(A;B;C) = ABC + ABC' + A'BC + AB'C + A'B'C}$

Si se tiene una productoria de términos suma, para obtener la segunda forma canónica, a los términos que les falta alguna variable se le suma la variable multiplicada por su complemento, ya que esa multiplicación es un 0 lógico y sumar un 0 no modifica la expresión. Si se forman más de un término idéntico solo se toma uno y los otros se eliminan.

Ejemplo: Una expresión F en tres variables A; B y C, es decir $F(A;B;C) = (A+C) . (B+C). A$

Hallar la segunda forma canónica.

$F(A;B;C) = [A+\mathbf{(B*B')}+\mathbf{(C*C')}]\ \mathbf{.}\ [(A+C)+\mathbf{(B*B')]}.\ [(B+C)+\mathbf{(A*A')}]$

$F(A;B;C) = (A+B+C).(A+B+C').(A+B'+C).(A+B'+C').(\cancel{A+B+C}).(\cancel{A+B'+C}).(\cancel{A+B+C}).(A'+B+C)$

Se eliminan los términos que se repiten quedando solo uno de ellos

La Segunda forma Canónica de la expresión será:

$\mathbf{F(A;B;C) = (A+B+C).(A+B+C').(A+B'+C).(A+B'+C').(A'+B+C)}$