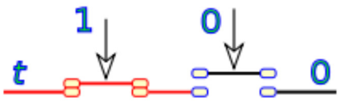
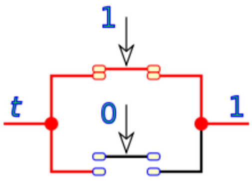
Álgebra de Boole

Álgebra de Boole (también llamada Retículas booleanas) en informática y matemática, es una estructura algebraica que rigorizan las operaciones lógicas Y, O y NO, así como el conjunto de operaciones unión, intersección y complemento.

Se denomina así en honor a George Boole, (2 de noviembre de 1815 a 8 de diciembre de 1864), matemático inglés que fue el primero en definirla como parte de un sistema lógico a mediados del siglo XIX. El álgebra de Boole fue un intento de utilizar las técnicas algebraicas para tratar expresiones de la lógica proposicional. En la actualidad, el álgebra de Boole se aplica de forma generalizada en el ámbito del diseño electrónico. Claude Shannon fue el primero en aplicarla en el diseño de circuitos de conmutación eléctrica biestables, en1948.

Definición:

El Álgebra de Boole está conformada sólo por dos elementos el 0, y 1. Es una estructura algebraica que establece las propiedades de las operaciones lógicas:

Operaciones:

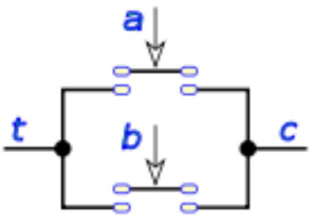
Hemos definido el conjunto **A = {1,0}** como el conjunto universal sobre el que se aplica el álgebra de Boole, sobre estos elementos se definen varias operaciones, veamos las más fundamentales:

<u>Operación suma:</u>

La operación suma (+) asigna a cada par de valores **a, b** de **A** un valor **c** de **A**:

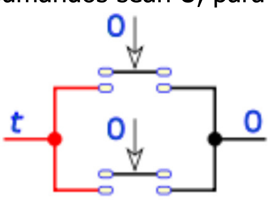
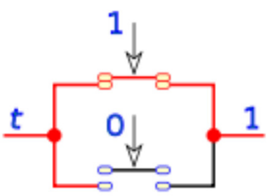
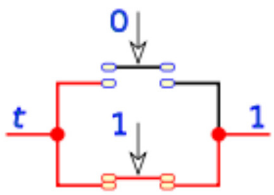
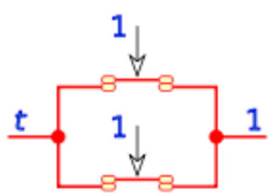
$$a + b = c$$

Su equivalencia en lógica de interruptores es un circuito de dos interruptores en paralelo.



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

Si uno de los valores de **a** o **b** es **1**, el resultado será **1**. Es necesario que los dos sumandos sean **0**, para que el resultado sea **0**.

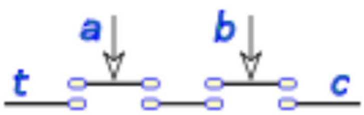





<u>Operación producto:</u>

La operación producto (**▪**) asigna a cada par de valores **a**, **b** de **A** un valor **c** de **A**:

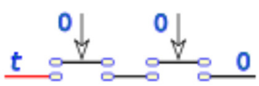
$$a \cdot b = c$$

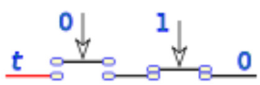
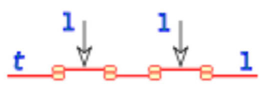
Esta operación en lógica de interruptores, es un circuito en serie de dos interruptores.



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

Sólo si los dos valores **a** y **b** son **1**, el resultado será **1**. Si uno solo de ellos es **0** el resultado será **0**.



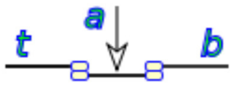



Operación negación:

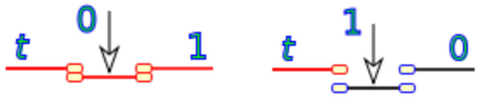
La operación negación presenta el opuesto del valor de **a**:

$$\overline{a} = b$$

Un interruptor inverso equivale a esta operación:



a	ā
0	1
1	0

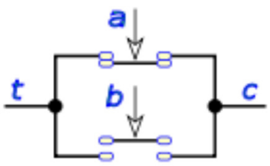


Operaciones combinadas:

Partiendo de estas tres operaciones elementales se pueden realizar otras más complejas, que podemos representar como ecuaciones booleanas, por ejemplo:

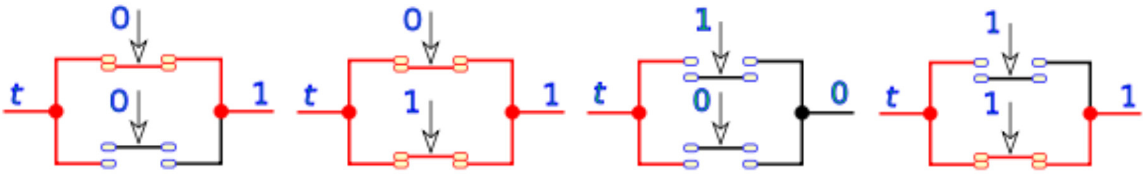
$$\overline{a} + b = c$$

Que representado en lógica de interruptores es un circuito de dos interruptores en paralelo, siendo el primero de ellos inverso.



a	b	ā + b
0	0	1
0	1	1
1	0	0
1	1	1

La distinta secuencia de valores de **a** y **b** da los resultados vistos en la tabla de verdad.



Leyes fundamentales:

1. Ley de idempotencia: $a \cdot a = a$
   $a + a = a$

2. Elemento unidad: $a \cdot 0 = 0$
   $a + 1 = 1$

3. Elemento neutro: $a + 0 = a$
   $a \cdot 1 = a$

4. Elemento simétrico: $a + \overline{a} = 1$
   $a \cdot \overline{a} = 0$

5. Ley de involución: $\overline{\overline{a}} = a$

6. Ley conmutativa: $a \cdot b = b \cdot a$
   $a + b = b + a$

7. Ley asociativa: $a \cdot (b \cdot c) = (a \cdot b) \cdot c$
   $a + (b + c) = (a + b) + c$

8. Ley distributiva: $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$
   $(a + b) \cdot c = (a \cdot c) + (b \cdot c)$

$$a+(b\cdot c)=(a+b)\cdot(a+c)$$
$$(a\cdot b)+c=(a+c)\cdot(b+c)$$
$$a+\overline{a}\cdot b=a+b$$

9. Ley de cancelación: $(a\cdot b)+a=a$

$$(a+b)\cdot a=a$$

10. Leyes de De Morgan: $\overline{a+b}=\overline{a}\cdot\overline{b}$

$$\overline{a\cdot b}=\overline{a}+\overline{b}$$

Principio de dualidad

El concepto de dualidad permite formalizar este hecho: a toda relación o ley lógica le corresponderá su dual, formada mediante el intercambio de los operadores unión (suma lógica) con los de intersección (producto lógico), y de los 1 con los 0.

Además hay que cambiar cada variable por su negada. Esto causa confusión al aplicarlo en los teoremas básicos, pero es totalmente necesario para la correcta aplicación del principio de dualidad. Véase que esto no modifica la tabla adjunta.

	Adición	Producto
1	$a+\bar{a}=1$	$a\cdot\bar{a}=0$
2	$a+0=a$	$a\cdot 1=a$
3	$a+1=1$	$a\cdot 0=0$
4	$a+a=a$	$a\cdot a=a$
5	$a+b=b+a$	$a\cdot b=b\cdot a$
6	$a+(b+c)=(a+b)+c$	$a\cdot(b\cdot c)=(a\cdot b)\cdot c$
7	$a+(b\cdot c)=(a+b)\cdot(a+c)$	$a\cdot(b+c)=a\cdot b+a\cdot c$
8	$a+a\cdot b=a$	$a\cdot(a+b)=a$
9	$\overline{(a+b)}=\bar{a}\cdot\bar{b}$	$\overline{(a\cdot b)}=\bar{a}+\bar{b}$

Otras formas de notación del álgebra de Boole

En matemática se emplea la notación empleada hasta ahora ({0,1}, +, ▪) siendo la forma más usual y la más cómoda de representar. Por ejemplo las leyes de De Morgan se representan así:

$$\overline{a+b}=\bar{a}\cdot\bar{b}$$
$$\overline{a\cdot b}=\bar{a}+\bar{b}$$

Cuando el álgebra de Boole se emplea en electrónica, suele emplearse la misma denominación que para las puerta lógica AND (Y), OR (O) y NOT (NO), ampliándose en ocasiones con X-OR (O exclusiva) y su negadas NAND (NO Y), NOR (NO O) y X-NOR (equivalencia). las variables pueden representarse con letras mayúsculas o minúsculas, y pueden tomar los valores {0, 1}. Empleando esta notación las leyes de De Morgan se representan:

$$\text{NOT }(a\text{ OR }b)=\text{NOT }a\text{ AND NOT }b$$
$$\text{NOT }(a\text{ AND }b)=\text{NOT }a\text{ OR NOT }b$$

En su aplicación a la lógica se emplea la notación $(\vee,\wedge,\neg)$ y las variables pueden tomar los valores {F, V}, falso o verdadero, equivalentes a {0, 1}. Con la notación lógica las leyes de Morgan serían así:

$$\neg(a\vee b)=\neg a\wedge\neg b$$
$$\neg(a\wedge b)=\neg a\vee\neg b$$

En el formato de Teoría de conjuntos el Álgebra de Boole toma el aspecto: $(\cup, \cap \sim, \{0,1\})$

En esta notación las leyes de De Morgan serían así:

$$\sim(a \cup b) = \sim a \cap \sim b$$
$$\sim(a \cap b) = \sim a \cup \sim b$$

Desde el punto de vista práctico existe una forma simplificada de representar expresiones booleanas. Se emplean apóstrofos (') para indicar la negación, la operación suma (+) se representa de la forma normal en álgebra, y para el producto no se emplea ningún signo, las variables se representan, normalmente con una letra mayúscula, la sucesión de dos variables indica el producto entre ellas, no una variable nombrada con dos letras.

La representación de las leyes de De Morgan con este sistema quedaría así, con letra minúsculas para las variables:

$$(a+b)' = a'b'$$
$$(ab)' = a' + b'$$

y así, empleando letras mayúsculas para representar las variables:

$$(A+B)' = A'B'$$
$$(AB)' = A' + B'$$

Todas estas formas de representación son correctas, se utilizan de hecho, y pueden verse al consultar bibliografía. La utilización de una u otra notación no modifica el álgebra de Boole, sólo su aspecto, y depende de la rama de las matemáticas o la tecnología en la que se esté utilizando para emplear una u otra notación.

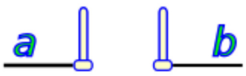
Álgebra de Boole aplicada a la informática

Se dice que una variable tiene valor booleano cuando, en general, la variable contiene un 0 lógico o un 1 lógico. Esto, en la mayoría de los lenguajes de programación, se traduce en false (falso) o true (verdadero), respectivamente.

Una variable puede no ser de tipo booleano, y guardar valores que, en principio, no son booleanos; ya que, globalmente, los compiladores trabajan con esos otros valores, numéricos normalmente, aunque también algunos permiten cambios desde, incluso, caracteres, finalizando en valor booleano. ..

El 0 lógico

El valor booleano de negación suele ser representado como **false**, aunque también permite y equivale al valor natural, entero y decimal (exacto) 0, así como la cadena "false", e incluso la cadena "0".



El 1 lógico

En cambio, el resto de valores apuntan al valor booleano de afirmación, representado normalmente como true, ya que, por definición, el valor 1 se tiene cuando no es 0. Cualquier número distinto de cero se comporta como un 1 lógico, y lo mismo sucede con casi cualquier cadena (menos la "false", en caso de ser ésta la correspondiente al 0 lógico).