ÁLGEBRA BOOLEANA

POR: DOMENICA NARVAEZ

¿QUÉ ES?

El álgebra booleana es un sistema matemático creado por George Boole en el siglo XIX, que se basa en la lógica binaria. Trata con variables que solo pueden tener dos valores: verdadero (representado como 1) o falso (representado como 0). Se usa ampliamente en informática, electrónica digital (como circuitos lógicos) y razonamiento lógico. En esencia, modela operaciones lógicas como "y", "o" y "no" de manera algebraica, permitiendo simplificar expresiones y resolver problemas de lógica.

OPERACIONES BÁSICAS DEL ÁLGEBRA BOOLEANA

AND (Y, conjunción): Denotado como A ∧ B o A · B. Resultado es 1 solo si ambas entradas son 1.

A	B	A ∧ B
0	0	0
0	1	0
1	0	0
1	1	1

OR (O, disyunción): Denotado como A ∨ B o A + B. Resultado es 1 si al menos una entrada es 1.

A	B	A ∨ B
0	0	0
0	1	1
1	0	1
1	1	1

NOT (Negación): Denotado como ¬A o A'. Invierte el valor.

A	¬A
0	1
1	0

REGLAS DE LA LÓGICA MATEMÁTICA APLICABLES (LEYES BOOLEANAS)

El álgebra booleana sigue reglas similares a la aritmética, pero adaptadas a la lógica binaria. Estas leyes permiten simplificar expresiones booleanas y son fundamentales en la lógica matemática.

NOMBRE DE LA LEY	LOGICA DE PROPOSICIONES
1. Idempotencia	• $x.\ x=x$ • $x+ x=x$
2. Identidad	• $x.1=x$ • $x + 0=x$
3. Dominación	• $x.0=0$ • $x+1=1$
4. Conmutativa	• $x.\ y=y.\ x$ • $x+ y=y+ x$
5. Asociativa	• $x.\ y.\ z \Leftrightarrow (x.\ y).z \Leftrightarrow x.(y.\ z)$ • $x + y + z=(x + y) + z= x + (y + z)$
6. Distributiva	• $x.(y + z)= (x.\ y)+(x.\ z)$ • $x+(y.z)= (x + y).(x + z)$
7. Complementación: • **Contradicción** • **Tercero excluido**	• $x.\ \overline{x}= 0$ • $x + \overline{x}=1$
8. Involución	• $\overline{\overline{x}}=x$ doble negación • $\overline{\overline{\overline{x}}}=\overline{x}$ triple negación
9. D' Morgan	• $\overline{x.y}= \overline{x} + \overline{y}$ • $\overline{x+y}= \overline{x} . \overline{y}$
10. Absorción	• $x.(x + y)= x$ • $x + (x.\ y)= x$

LEYES BOOLEANAS

1. Ley de Identidad:

- A . 1 = A (AND con 1 no cambia A).
- A + 0 = A (OR con 0 no cambia A).
- Ejemplo:

Si A=1, 1 . 1 = 1 (verdadero).
Si A=0, 0 . 1 = 0 (falso).

2. Ley de Dominio:

- A . 0 = 0 (AND con 0 siempre es 0).
- A + 1 = 1 (OR con 1 siempre es 1).
- Ejemplo:

Cualquier cosa AND falso es falso.

3. Ley de Complemento:

- A . A' = 0 (A y su negación es imposible).
- A + A' = 1 (A o su negación es siempre cierto).
- Ejemplo:

"Es de día y no es de día" es falso.

LEYES BOOLEANAS

4. Ley Conmutativa:
- A . B = B . A (orden no importa en AND).
- A + B = B + A (orden no importa en OR).
- Ejemplo:

"Comprar pan y leche" es lo mismo que "comprar leche y pan"; el orden de las condiciones no afecta el resultado.

5. Ley Asociativa:
- (A . B) . C = A . (B . C) (agrupación no afecta en AND).
- (A + B) + C = A + (B + C) (agrupación no afecta en OR).
- Ejemplo:

En una cadena de suministro, verificar "proveedor A, B y C" es igual a "proveedor A y (B y C)"; la agrupación no cambia la verificación total.

6. Ley Distributiva:
- A . (B + C) = (A . B) + (A . C) (AND se distribuye sobre OR).
- A + (B . C) = (A + B) . (A + C) (OR se distribuye sobre AND).

LEYES BOOLEANAS

7. Ley de Absorción:

- A . (A + B) = A (A absorbe A + B).
- A + (A . B) = A (A absorbe A . B).
- Ejemplo:

Si ya tienes "llueve" (A), entonces "llueve y (llueve o nieva)" se simplifica a solo "llueve", ya que la condición extra no añade nada.

8. Ley de De Morgan (muy útil en lógica):

- (A . B)' = A' + B' (negación de AND es OR de negaciones).
- (A + B)' = A' . B' (negación de OR es AND de negaciones).

9. Ley de Doble Negación:

- (A')' = A (negar dos veces vuelve al original).

LEYES BOOLEANAS

10. Ley de Idempotencia (agregada como solicitaste):

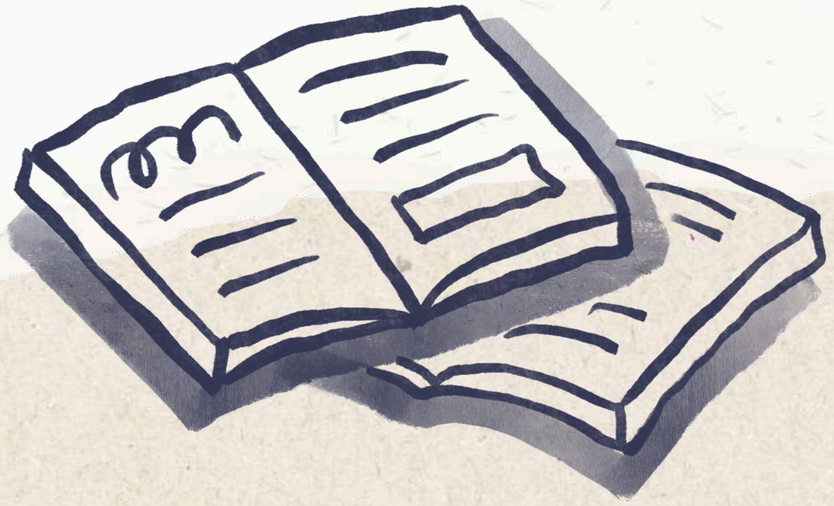
- A . A = A (AND con sí mismo no cambia).
- A + A = A (OR con sí mismo no cambia).
- Ejemplo:

En un sistema de votación, "votar sí y sí" es igual a "votar sí"; repetir la acción no altera el resultado.

EJERCICIO PRÁCTICO

Simplifica la expresión booleana: $A . (A' + B) + A . B$.

- **Paso 1**: Aplica la ley distributiva para expandir: $A . (A' + B) = (A . A') + (A . B)$. (Distributiva: $A . (X + Y) = A.X + A.Y$).
- **Paso 2**: Simplifica $A . A'$ usando la ley de complemento: $A . A' = 0$. Así, la expresión queda: $0 + (A . B) + A . B$.
- **Paso 3**: Combina términos: $(A . B) + (A . B) = A . B + A . B$. Aplica la ley de idempotencia: $A . B + A . B = A . B$.
- **Resultado final**: $A . B$.



APLICACIONES

1. Diseño de Circuitos Digitales (Hardware)
   - **Circuitos Aritméticos:** El Álgebra de Boole se usa para diseñar la Unidad Aritmético-Lógica (ALU) del procesador, que es la encargada de realizar sumas, restas y otras operaciones matemáticas. Un circuito sumador (como un full adder) se diseña y simplifica enteramente con ecuaciones booleanas.
   - **Memoria (Flip-flops y Latches):** Las unidades básicas de memoria (como los flip-flops) que almacenan un solo bit (0 o 1) se construyen con compuertas lógicas y se analizan usando Álgebra de Boole.

2. Estructuras de Datos y Bases de Datos
   - **Búsquedas Lógicas:** Cuando realizas una búsqueda avanzada en internet, una base de datos o en el explorador de archivos, utilizas operadores booleanos. Por ejemplo, buscar ("perros" AND "gatos") NOT "domésticos" aplica directamente las operaciones booleanas de Conjunción (AND) y Negación (NOT).
   - **Consultas SQL:** En el lenguaje de consulta estructurada (SQL) para bases de datos, la cláusula WHERE utiliza operadores lógicos (AND, OR, NOT) que son una aplicación directa del Álgebra de Boole para seleccionar los registros.

APLICACIONES

3. Programación y Control de Flujo

- **Estructuras de Control:** Las sentencias condicionales (if/else, while, for) se basan en la evaluación de expresiones booleanas. El programa solo ejecuta un bloque de código si la condición compuesta es verdadera (1). Ejemplo: if (A > 5 AND B == 10)
- **Tipos de Datos Booleanos:** Casi todos los lenguajes de programación tienen un tipo de dato llamado bool que solo puede ser True o False, la representación directa de los valores 1 y 0 del álgebra.

4. Redes y Telecomunicaciones

- **Máscaras de Red:** En redes IP, las máscaras de subred utilizan la operación AND booleana para determinar la dirección de red a la que pertenece un dispositivo, un proceso llamado subnetting.
- **Protocolos de Control:** Muchos protocolos de red y comunicación utilizan la lógica booleana para establecer estados de conexión y verificar la integridad de los datos.

¡MUCHAS GRACIAS!