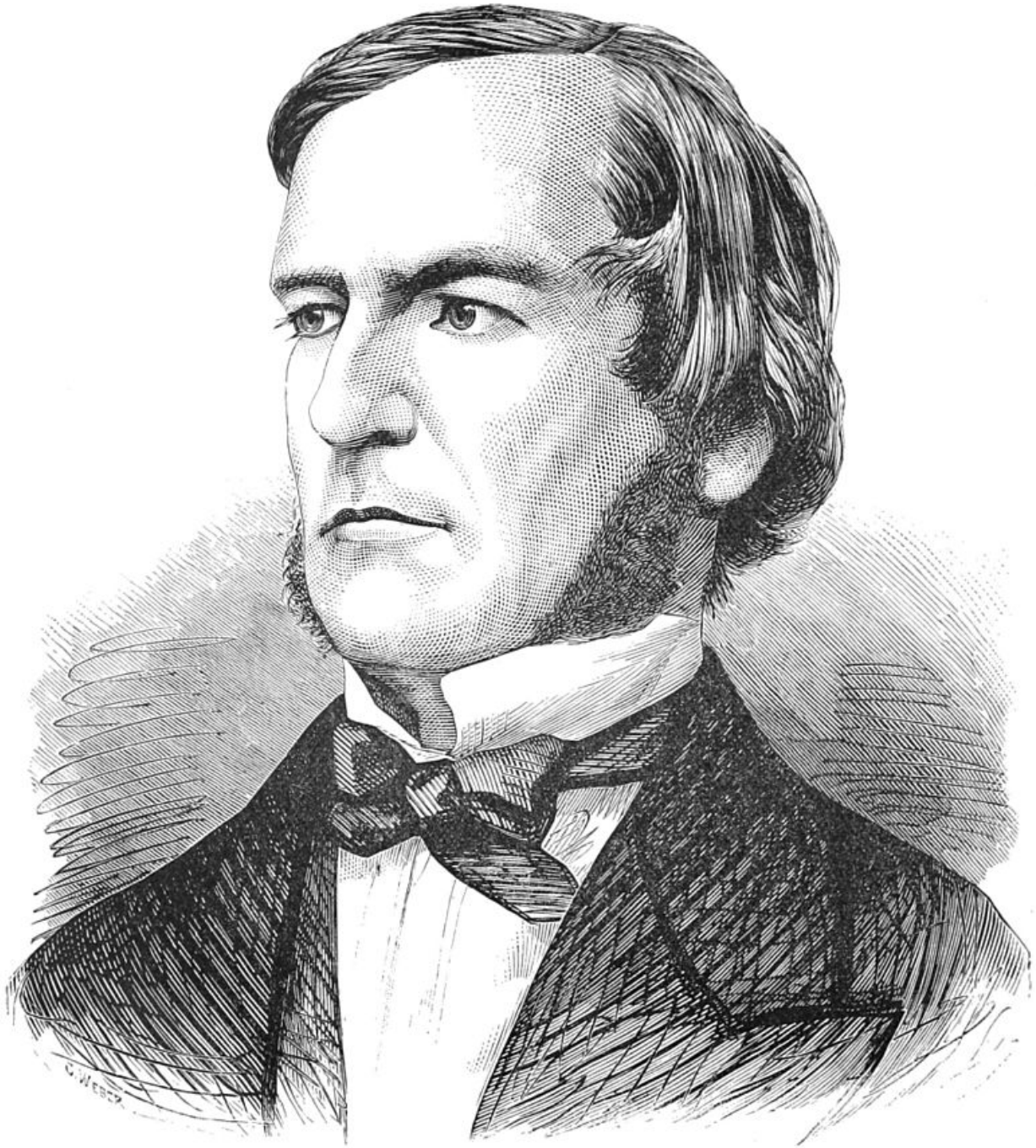
Álgebra Booleana

Por Ariel Parra

[Γα=Ω5]



El álgebra de Boole es una herramienta diseñada para poder representar proposiciones lógicas en forma algebraica. Desarrollada originalmente en 1847 por George Boole.

Actualmente, este tipo de álgebra es utilizada ampliamente en el diseño y simplificación de sistemas digitales binarios, es decir, sistemas que basan todo su comportamiento en los valores {1,0}.

Para nosotros nos sera útil para simplificar expresiónes lógicas y en operaciones de bits.

Operaciones básicas

NOT (!)

a	¬a
0	1
1	0

AND (&&)

a	b	a · b
0	0	0
0	1	0
1	0	0
1	1	1

OR (||)

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

Propiedades y Axiomas

1. **Ley asociativa:**

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

2. **Existencia del elemento neutro:**

$$a + 0 = a$$
$$a \cdot 1 = a$$

3. **Ley conmutativa:**

$$a + b = b + a$$
$$a \cdot b = b \cdot a$$

4. **Ley distributiva:**

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

5. **Existencia del elemento complementario:**

$$a + \bar{a} = 1$$
$$a \cdot \bar{a} = 0$$

AND

- **Ley de idempotencia para el producto:**

$$a \cdot a = a$$

- **Ley de absorción para el producto:**

$$a \cdot 0 = 0$$

- **Ley de identidad para el producto:**

$$a \cdot 1 = a$$

OR

- **Ley de idempotencia para la suma:**

$$a + a = a$$

- **Ley de absorción para la suma:**

$$a + 1 = 1$$

- **Ley de identidad para la suma:**

$$a + 0 = a$$

NOT

- **Ley de involución:**

$$\bar{\bar{a}} = a$$

Leyes de De Morgan

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

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

Conectivas derivadas

NOR

Es la negada de la función “OR”:

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

En c++: `!(a&&b)`

Tabla NOR

a	b	y
0	0	1
0	1	0
1	0	0
1	1	0

NAND

Es la negada de la función “AND”:

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

En c++: `!(a && b)`

Tabla NAND

a	b	¬(a + b)
0	0	1
0	1	1
1	0	1
1	1	0

XOR

Es la función OR excluyente: uno u otro, pero no los dos. Se puede utilizar para detectar señales que son distintas.

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

En c++: `(a||b)&&!(a&&b)`

Tabla XOR

a	b	a ⊕ b
0	0	0
0	1	1
1	0	1
1	1	0

XNOR

Es la negada de la función “XOR”. Se puede utilizar para detectar señales que son iguales:

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

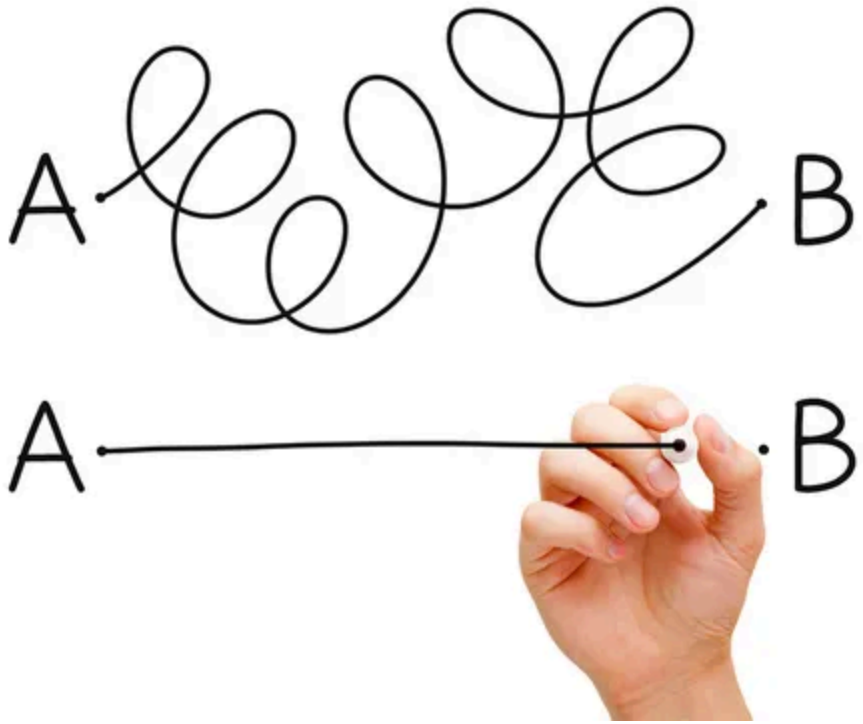
En c++: `!((a||b)&&!(a&&b))`

Tabla XNOR

a	b	¬(a ⊕ b)
0	0	1
0	1	0
1	0	0
1	1	1

https://www.boolean-algebra.com/

Prolemas de simplificación de expresiones



Problema 1

Expresión a simplificar:

$$F = (A + \overline{B}) \cdot (A + B)$$

Solución:

1. **Aplicar la Ley Distributiva**:

$$F = (A + \overline{B}) \cdot (A + B) = A \cdot (A + B) + \overline{B} \cdot (A + B)$$

2. **Simplificar**:

   - Para (A \cdot (A + B)):

$$A \cdot (A + B) = A$$

     (por la Ley de Absorción)

   - Para (\overline{B} \cdot (A + B)):

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

$$\overline{B} \cdot B = 0$$

     Entonces:

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

Entonces:

$$F = A + \overline{B} \cdot A$$

3. **Aplicar la Ley de Absorción**:

$$A + \overline{B} \cdot A = A$$

Entonces la expresión simplificada es:

$$F = A$$

Problema 2

Expresión a simplificar:

$$F = (A \cdot B) + (\overline{A} \cdot C) + (B \cdot \overline{C})$$

Solución:

1. **Simplificar por agrupación**:
   - Agrupa

$$((A \cdot B))((\overline{A} \cdot C))$$

   con la Ley de Distributiva:

$$(A \cdot B) + (\overline{A} \cdot C) = (A \cdot B) + (\overline{A} \cdot C)$$

   - Reescribe la expresión:

$$F = (A \cdot B) + (\overline{A} \cdot C) + (B \cdot \overline{C})$$

2. **Aplicar la Ley de Absorción**:
   - Nota que ((A \cdot B) + (B \cdot \overline{C}) = B) (porque (A \cdot B) está contenido en (B)).
   - Entonces:

$$F = B + (\overline{A} \cdot C)$$

   La expresión no se simplifica más. Entonces:

$$F = B + (\overline{A} \cdot C)$$

Ejemplo Práctico: Sistema de Alarma en una Casa Inteligente

Imaginemos que queremos que el sistema de alarma se active bajo ciertas condiciones:

1. **Condiciones**:
   - La puerta de entrada está abierta (`P`).
   - El sensor de movimiento detecta movimiento (`M`).
   - El sistema de alarma está encendido (`A`).

Queremos que la alarma suene si:

- **La puerta está abierta y hay movimiento**, o
- **La puerta está abierta y el sistema de alarma está encendido**.

Expresión Booleana Inicial

La lógica para activar la alarma se puede expresar como:

$$\text{Alarma} = (P \land M) \lor (P \land A)$$

o en sintaxis de c++

```
(P || M) && (P || A)
```

Donde:

- P = Puerta abierta (0 = cerrada, 1 = abierta)
- M = Movimiento detectado (0 = no, 1 = sí)
- A = Alarma encendida (0 = apagada, 1 = encendida)

Simplificación de la Expresión

1. **Expresión Inicial**:

$$\text{Alarma} = (P \land M) \lor (P \land A)$$

2. **Aplicar Leyes de Booleanos**:

   - **Factorización**:

$$\text{Alarma} = P \land (M \lor A)$$

3. **Resultado Simplificado**:

   - La expresión simplificada es:

$$\text{Alarma} = P \land (M \lor A)$$

Ejemplo Simplificado en C++

```
bool shouldActivateAlarm(bool doorOpen, bool motionDetected, bool alarmOn) {
    return doorOpen && (motionDetected || alarmOn);
}
int main() {
    bool doorOpen = true;         // Puerta abierta
    bool motionDetected = false; // No hay movimiento
    bool alarmOn = true;          // Alarma encendida

    if (shouldActivateAlarm(doorOpen, motionDetected, alarmOn)) {
        cout << "La alarma  se activa.";
    } else {
        cout << "La alarma NO se activa.";
    }
    return 0;
}
```

Problemas

> para subir el problema I, deben ingresar manualmente al gym, darle en submit y elegir el problema https://codeforces.com/gym/102890/submit

- Problema I. Is this the best deal?, del ICPC Mexico ronda 3 del 2020

Referencias

- boolean-algebra. (s.f.). *Boolean Algebra Simplifier*. Recuperado de https://www.boolean-algebra.com/
- Brunete, A. (s.f.). *3.1 Álgebra de Boole*. Recuperado de https://bookdown.org/alberto_brunete/intro_automatica/algebraboole.html
- khandelwal, S. (2024) *Boolean Algebra*. Recuperado de https://www.geeksforgeeks.org/boolean-algebra/
- UNAM. (s.f.) *Algebra de Boole*. Recuperado de http://ecampus.fca.unam.mx/ebook/imprimibles/informatica/arquitectura_computadoras/Unidad_4.pdf
- UNAM. (s.f). *Álgebra Booleana*. Recuperado de https://virtual.cuautitlan.unam.mx/intar/sistdig/algebra-booleana/