

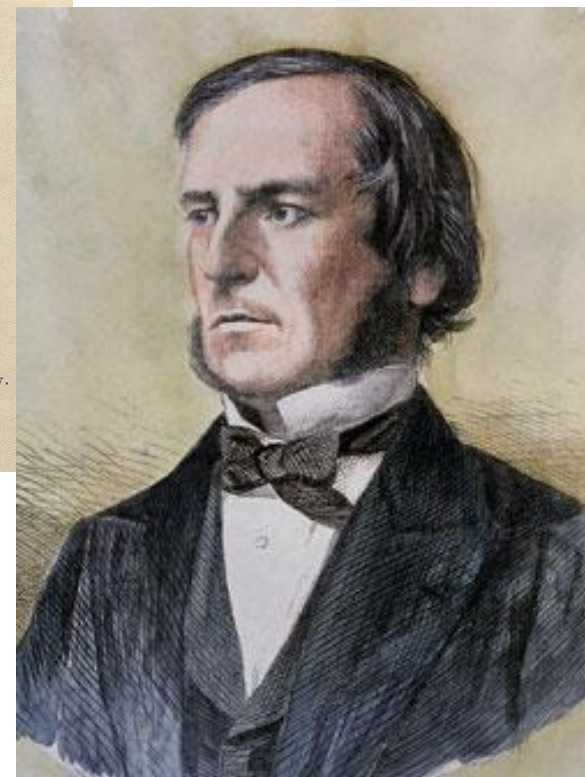
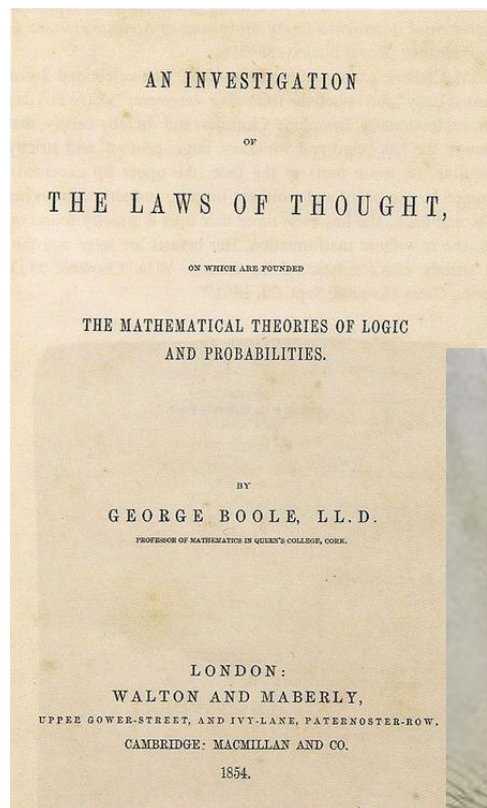
Elementos de Sistemas

estudo para aula 2 – Álgebra Booleana

*"A força motriz da invenção matemática não é o raciocínio, mas a imaginação."
"The moving power of mathematical invention is not reasoning but imagination."
Augustus De Morgan (1806 - 1871) matemático britânico*

George Boole

George Boole foi um matemático britânico, que desenvolveu os conceitos da álgebra booleana em 1847, que viria a se tornar uma das bases da computação moderna.

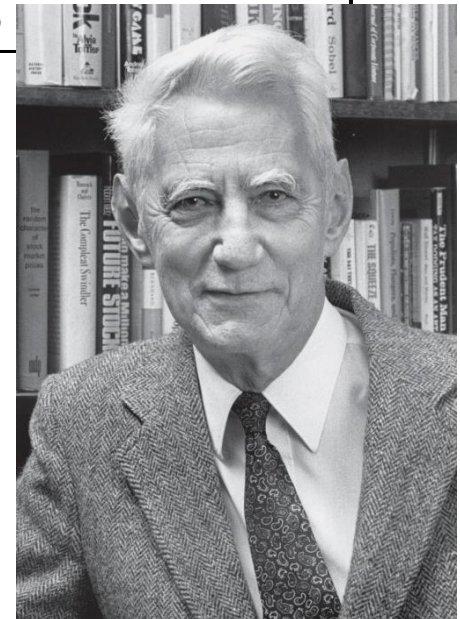
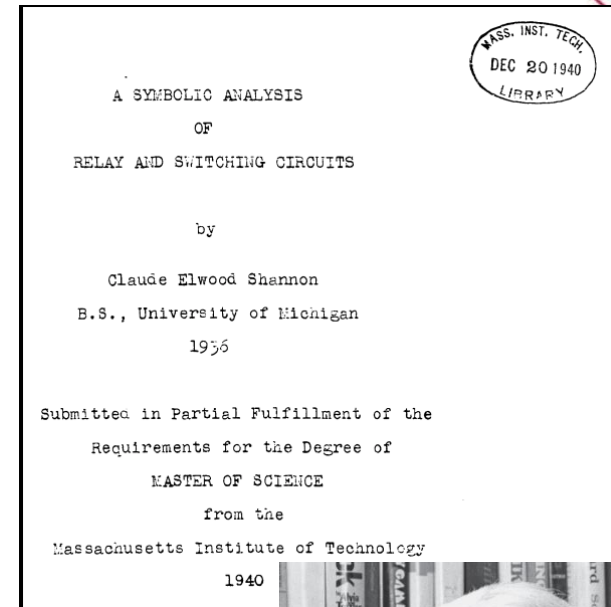
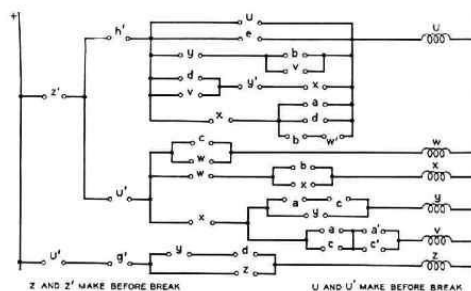


George Boole (1815 — 1864)

Na álgebra booleana as variáveis só possuem dois valores e as operações fundamentais são a $\wedge$ (conjunção), $\vee$ (disjunção) e $\neg$ (negação).

Claude Shannon

O engenheiro americano Claude Shannon (EUA) aplicou as ideias da álgebra booleana em circuitos elétricos. Enquanto trabalhava no Analisador Diferencial de Vannevar Bush, ele percebeu uma relação entre um relé ser ligado/desligado e a lógica booleana verdadeiro/falso.



Claude Shannon (1916 – 2001)

Álgebra Booleana

As variáveis (dígitos) só podem assumir dois valores.

0

desligado

falso

0 volt

branco

1

ligado

verdadeiro

5 volts

preto

Em computação chamamos isso de **bit**.

Operações Básicas em Álgebra Booleana

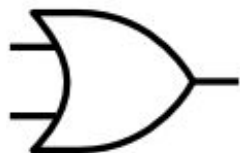
OR

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 1$$



AND

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

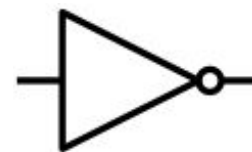
$$1 \times 1 = 1$$



NOT

$$\bar{0} = 1$$

$$\bar{1} = 0$$



símbolos ANSI

Funções Booleanas e Tabelas Verdade

Lógicas Booleanas podem ser expressas com:

funções

ou

tabelas verdade

$$f(x, y, z) = (x + y) z$$

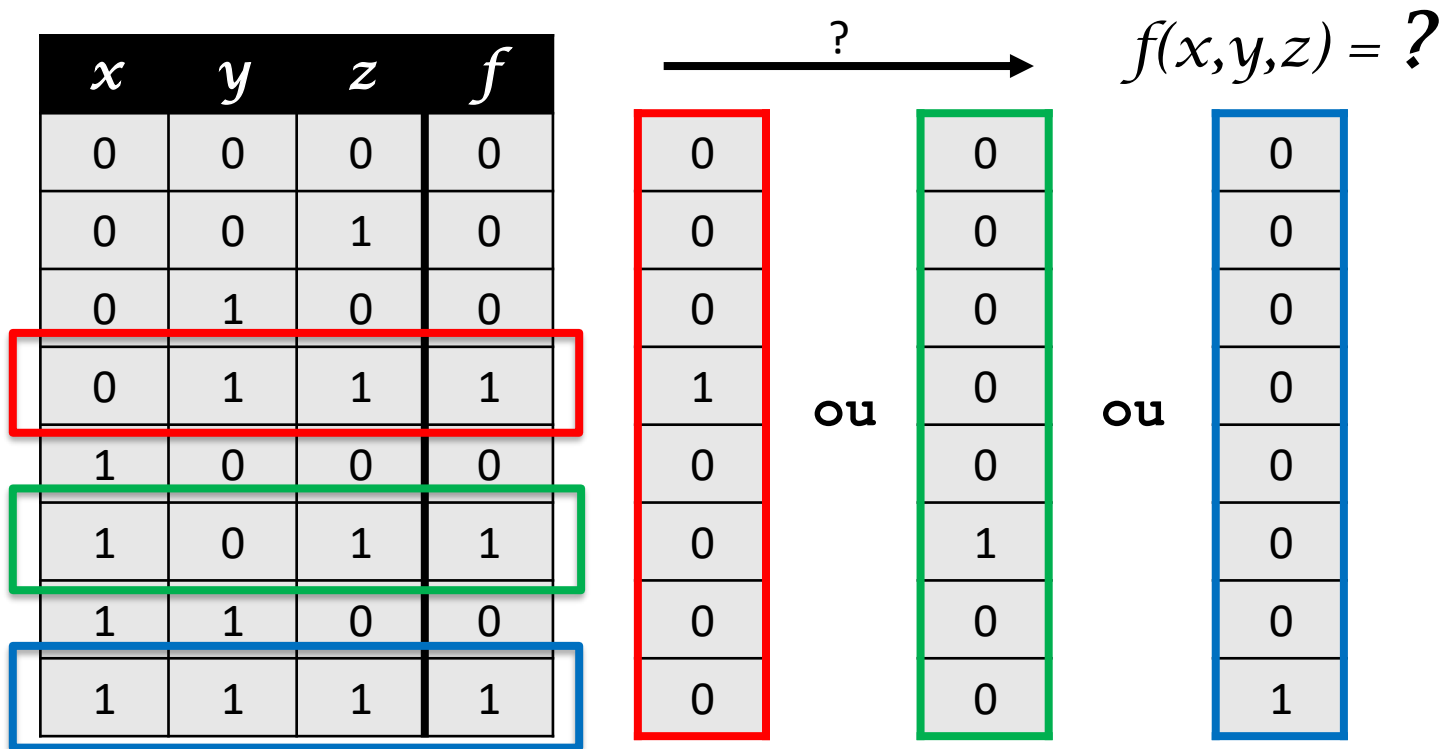
$$f(1, 0, 1) = (1 + 0) 1 = 1$$



x	y	z	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Funções geradas por Tabelas Verdade

Como a partir de uma tabela verdade gerar sua função Booleana?



Formas Canônicas

Toda expressão booleana pode ser escrita em uma forma padronizada, chamada de forma canônica:

- Soma de Produtos (Forma Normal Disjuntiva ou Soma de Mintermos)
Representar as condições de entrada que produzirão a saída 1 (e portanto as demais condições produzirão saída 0).
- Produto de Somas (Forma Normal Conjuntiva ou Produto de Maxtermos)
Representar as condições de entrada que produzirão a saída 0 (e portanto as demais condições produzirão saída 1).

Soma de Produtos

A	B	C	Q
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Identifique os valores de entrada para qual a saída é **1** (Mintermos)

Multiplique os termos (E) de forma que o resultado de cada expressão seja 1

Some (OU) todas as expressões

$$Q = \bar{A} * \bar{B} * C + \bar{A} * B * C + A * \bar{B} * \bar{C} + A * B * \bar{C}$$

Produto de Somas

A	B	C	Q
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Identifique os valores de entrada para qual a saída é **0** (Maxtermos)

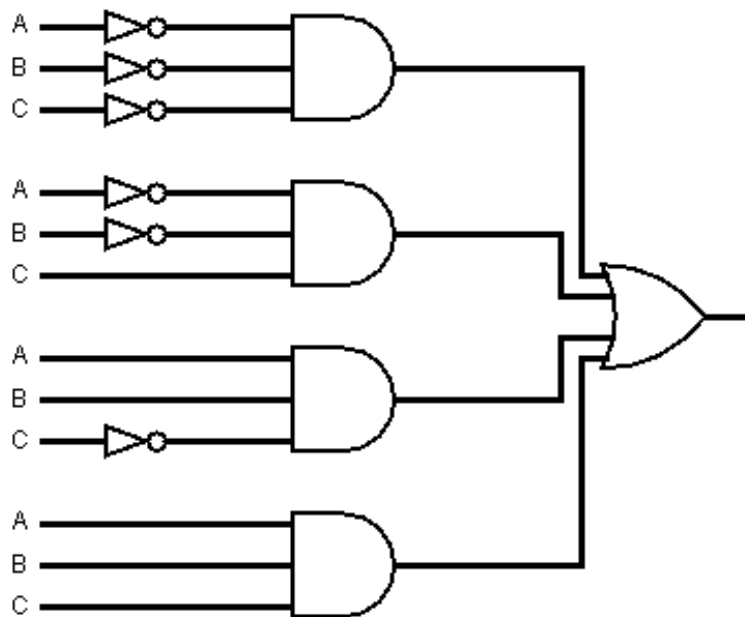
Some os termos (OU) de forma que o resultado de cada expressão seja 0

Multiplique (E) todas as expressões

$$Q = (A+B+C) * (A+\bar{B}+C) * (\bar{A}+B+\bar{C}) * (\bar{A}+\bar{B}+\bar{C})$$

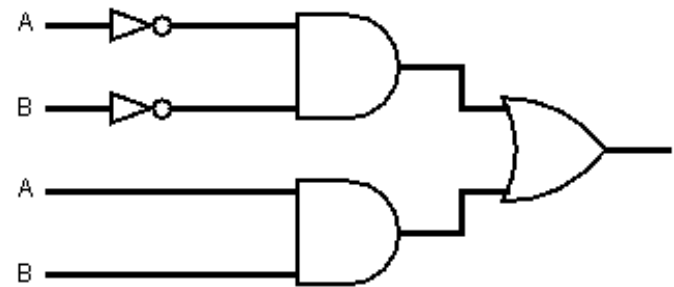
Para que álgebra booleana

Desejamos criar circuitos, e o mais eficiente possível



$$\bar{A} \cdot \bar{B} \cdot \bar{C} + \bar{A} \cdot \bar{B} \cdot C + A \cdot B \cdot \bar{C} + A \cdot B \cdot C$$

≡



$$\bar{A} \cdot \bar{B} + A \cdot B$$

Propriedades da Álgebra Booleana

Lei da Identidade	$A = A$ $\bar{A} = \bar{A}$
Lei da Comutatividade	$A \cdot B = B \cdot A$ $A + B = B + A$
Lei da Associatividade	$A \cdot (B \cdot C) = A \cdot B \cdot C$ $A + (B + C) = A + B + C$
Lei da Distributividade	$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$ $A + (B \cdot C) = (A + B) \cdot (A + C)$
Lei da Idempotência	$A \cdot A = A$ $A + A = A$
Lei do Complemento Duplo	$\bar{\bar{A}} = A$
Lei da Complementariedade	$A \cdot \bar{A} = 0$ $A + \bar{A} = 1$
Lei da Intersecção	$A \cdot 1 = A$ $A \cdot 0 = 0$
Lei da União	$A + 1 = 1$ $A + 0 = A$
Teorema de DeMorgan	$\overline{A \cdot B} = \bar{A} + \bar{B}$ $\overline{A + B} = \bar{A} \cdot \bar{B}$

Simplificação de Expressões Booleanas

Usando as propriedades da álgebra booleana é possível simplificar expressões complexas. Por exemplo, podemos usar simplificações para deixarmos as representações canônicas mais simples.

Exemplo:

$$(\bar{A} + B) * (A + B) =$$

$$B + (A * \bar{A}) =$$

$$B + (0) =$$

$$B$$

Lei da Distributividade

Lei da Complementariedade

Lei da União

Simplificação de Expressões Booleanas

Outro Exemplo:

$$A * B * C + A * \overline{C} + A * \overline{B} =$$

$$A * (B * C + \overline{C} + \overline{B}) = \leftarrow \text{Lei da Distributividade}$$

$$A * (B * C + (\overline{C} + \overline{B})) = \leftarrow \text{Lei da Associatividade}$$

$$A * (B * C + \overline{C * B}) = \leftarrow \text{Teorema de DeMorgan}$$

$$A * (1) = \leftarrow \text{Lei da Complementariedade}$$

$$A \leftarrow \text{Lei da Intersecção}$$

Mapas de Veitch-Karnaugh

- Uma opção a simplificação de lógicas é a aplicação de Mapas de Veitch-Karnaugh. Que muitas vezes pode acabar mostrando soluções não facilmente identificáveis usando álgebra booleana.
- O Mapa de Karnaugh usa representações gráficas e onde regiões identificam simplificações. Quando múltiplas regiões estão presentes, basta somar elas

A		
B	0	1
0	0	1
1	0	1

Região A

A		
B	0	1
0	0	0
1	1	1

Região B

A		
B	0	1
0	0	0
1	0	1

Região $A \cdot B$

A		
B	0	1
0	0	1
1	0	0

Região $A \cdot \bar{B}$

A		
B	0	1
0	1	0
1	1	0

Região $\bar{A}$

A		
B	0	1
0	1	1
1	0	0

Região $\bar{B}$

A		
B	0	1
0	0	0
1	1	0

Região $\bar{A} \cdot B$

A		
B	0	1
0	1	0
1	0	0

Região $\bar{A} \cdot \bar{B}$

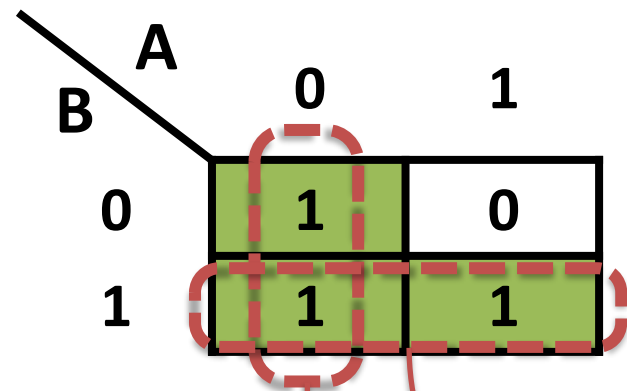
Exemplo de Mapa de Karnaugh

Tabela Verdade

A	B	Q
0	0	1
0	1	1
1	0	0
1	1	1

$$Q = \bar{A} * \bar{B} + \bar{A} * B + A * B$$

Mapa de Karnaugh



$$Q = \bar{A} + B$$

Mapas de Veitch-Karnaugh Maiores

Os mapas de Karnaugh podem ter dimensões maiores, nesses casos as regiões passam de um lado para o outro.

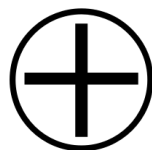
CD \ AB	00	01	11	10
00	0	0	0	0
01	1	0	0	1
11	0	0	0	0
10	0	0	0	0

$$Q = \overline{B} * \overline{C} * D$$

Outras Portas XOR e XNOR

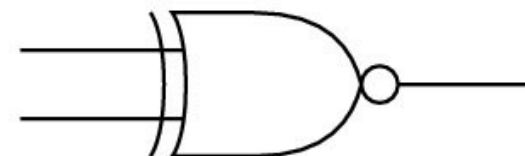
Um outro conjunto de portas lógicas muito usadas

XOR



A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

XNOR



A	B	Q
0	0	1
0	1	0
1	0	0
1	1	1

Não Importa (Don't Care)

O X indica que é indiferente o valor da variável naquela situação para a análise da tabela verdade.

A	B	C	Q
0	X	0	0
0	0	1	1
X	1	1	0
1	0	X	1
1	1	0	1

Simplificação por Python (PyEDA)

Para os mais curiosos:

<http://pyeda.readthedocs.io/en/latest/overview.html>

```
lpsoares — python — 101×32
>>> from pyeda.inter import *
>>> tabela = ttvars('x', 3)
>>> soma = truthtable(tabela, "01101001")
>>> carry = truthtable(tabela, "00010111")
>>> soma
x[2] x[1] x[0]
0 0 0 : 0
0 0 1 : 1
0 1 0 : 1
0 1 1 : 0
1 0 0 : 1
1 0 1 : 0
1 1 0 : 0
1 1 1 : 1
>>> carry
x[2] x[1] x[0]
0 0 0 : 0
0 0 1 : 0
0 1 0 : 0
0 1 1 : 1
1 0 0 : 0
1 0 1 : 1
1 1 0 : 1
1 1 1 : 1
>>> somam, carrym = espresso_tts(soma, carry)
>>> somam
Or(And(~x[0], x[1], ~x[2]), And(x[0], x[1], x[2]), And(x[0], ~x[1], ~x[2]), And(~x[0], ~x[1], x[2]))
>>> carrym
Or(And(x[1], x[2]), And(x[0], x[1]), And(x[0], x[2]))
>>>
```



Insper

www.insper.edu.br