5. *Álgebra Booleana*

5.1. Introdução a Àlgebra de Boole

Em 1854, o matemático inglês George Boole (1815-1864) apresentou um sistema matemático de análise lógica conhecido como **Álgebra de Boole** ou **Álgebra Booleana**. Em 1938, o engenheiro americano Claude Shannon utilizou as teorias da álgebra de Boole para a solução de problemas de circuitos de telefonia com relés, introduzindo na área tecnológica o campo da eletrônica digital, que emprega em seus sistemas um pequeno grupo de circuitos básicos padronizados conhecidos como portas lógicas (IDOETA & CAPUANO, 1998). As portas lógicas são a base dos circuitos lógicos utilizados em sistemas digitais e têm por finalidade combinar as diferentes expressões booleanas de modo a realizar determinada função (DAGHLIAN, 1995). Assim, a Álgebra Booleana foi fundamental para o desenho dos circuitos dos computadores eletrônicos modernos.

Por Álgebra Booleana entende-se um conjunto **B={a, b, c , ..}** junto com duas operações binárias **+** e **.** em B, uma operação singular **'** em B e dois elementos distintos **0** e **1** de B (GERSTING, 1995), tais que valem as seguintes propriedades: (para todo a , b , c em B):

Associativa (ASSOC)	(a + b) + c = a + (b + c) = a + b + c	(a . b) . c = a . (b . c) = a . b . c
Comutativa (COMUT)	a + b = b + a	a . b = b . a
Distributiva (DISTRIB)	a + (b . c) = (a + b) . (a + c)	a . (b + c) = (a . b) + (a . c)
Identidade e Idempotente (IDE)	a + 0 = a	a . 0 = 0
	a + 1 = 1	a . 1 = a
	a + a' = 1	a . a' = 0
	a + a = a	a . a = a
Absorção (ABS)	a + (a . b) = a	a. (a + b) = a
	a + (a . b') = a	a. (a + b') = a
	a + (a' . b) = a+b	a. (a' + b) = a.b
De Morgan (MORGAN)	(a+b)' = a' . b'	(a.b)' = a' + b'

Aplicações da Álgebra Booleana

A Álgebra Booleana pode ser aplicada a circuitos de interruptores e a circuito de portas lógicas. A seguir, será apresentada a representação gráfica de funções booleanas através de circuito de portas lógicas. Após, a sua representação na Forma Normal Conjuntiva e Disjuntiva e um exemplo de aplicação envolvendo esses conceitos.

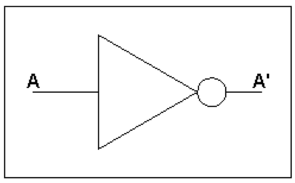
5.2. Portas Lógicas

A representação gráfica das funções booleanas é feita mediante símbolos padronizados por normas internacionais chamados blocos ou portas lógicas. As portas lógicas são a base dos circuitos lógicos utilizados em sistemas digitais e têm por finalidade combinar as diferentes expressões booleanas de modo a realizar determinada função (DAGHLIAN, 1995).

As seguintes são as portas lógicas padrão:

NOT

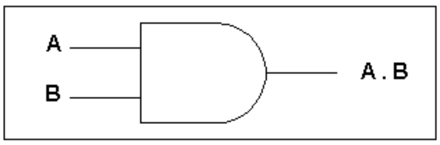
É a porta inversora. Seu símbolo e tabela-verdade são:



A	A'
0	1
1	0

AND

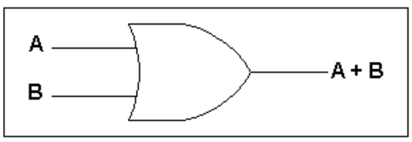
A porta **AND** mais simples possui 2 entradas e 1 saída.



A	B	A . B
0	0	0
0	1	0
1	0	0
1	1	1

OR

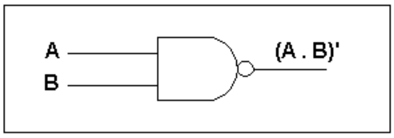
A porta **OR** mais simples possui, também, 2 entradas e 1 saída.



A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

NAND

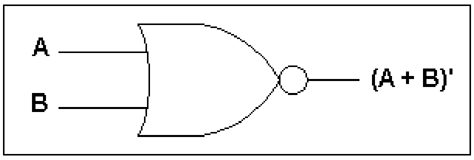
É equivalente à 1 (uma) porta **AND** seguida de 1 (uma) porta **NOT**.



A	B	(A . B)'
0	0	1
0	1	1
1	0	1
1	1	0

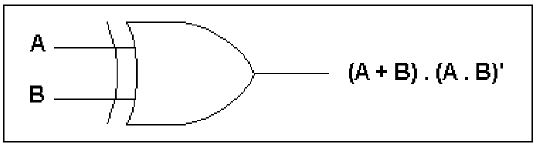
NOR

É equivalente à 1 (uma) porta **OR** seguida de 1 (uma) porta **NOT**.



A	B	(A + B)'
0	0	1
0	1	0
1	0	0
1	1	0

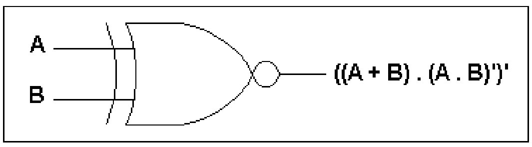
XOR é o "OU exclusivo"



A	B	XOR
0	0	0
0	1	1
1	0	1
1	1	0

XNOR

Equivalente à porta **XOR** seguida da porta **NOT**.



A	B	XNOR
0	0	1
0	1	0

	1	0	0
	1	1	1

Exemplos:

a) f= ab’ + c

b) f= a+b’

c) f= a.(b+c)’

d) f= (ab’)’

Exercício1 – Construir a tabela verdade das seguintes funções booleanas:

a) a’+b
b) f= ab’ + c
c) f= a.(b+c)’
d) f= (ab’)’

Exercício2 – Represente o diagrama lógico das seguintes funções booleanas:

a) f= (a+b+c).(d.e’)’

b) f= abc’ + a’cd+(a’+b)’

c) f= abc + d

5.3. Formas Normais

A partir de uma tabela verdade, pode-se obter uma expressão booleana que a represente. Para isso, usam-se as Formas Normais Disjuntiva (FND) ou Conjuntiva (FNC).

Para se obter uma **Forma Normal Disjuntiva** (FND ou soma de produtos):

1. Observamos todas as linhas da tabela que possuem **1** na última coluna;
2. Construímos para cada uma destas linhas as conjunções (produtos) correspondentes, sendo que quando ocorrer a situação 0, a variável fica negada;
3. Fazemos a disjunção (soma) destas conjunções obtendo uma fórmula em **FND** que satisfaz a tabela verdade.

Exemplo: Determine uma fórmula que satisfaça a tabela verdade abaixo:

A	**B**	**?**	
0	0	1	**A'.B'**
0	1	0	
1	0	0	
1	1	1	**A.B**

Resposta: Expressão obtida **(A'.B') + (A.B)**

Exercício3 – Encontre a Forma Normal Disjuntiva (FND) das seguintes funções representadas pelas tabelas verdade abaixo:

a)

A	**B**	**?**
0	0	0
0	1	1
1	0	0
1	1	1

b)

A	**B**	**?**
0	0	0
0	1	0
1	0	1
1	1	1

c)

A	**B**	**?**
0	0	1
0	1	0
1	0	1
1	1	1

d)

A	**B**	**C**	**?**
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Exercício4 – Construir o diagrama lógico das funções representadas pelas tabelas verdades do exercício anterior.

Para se obter uma **Forma Normal Conjuntiva** (FNC ou produto das somas):

1. Observamos todas as linhas da tabela que possuem **0** na última coluna;
2. Construímos para cada uma destas linhas as disjunções (somas) correspondentes, sendo que quando ocorrer a situação 1, a variável fica negada;
3. Fazemos a conjunção (produto) destas disjunções obtendo uma fórmula em **FNC** que satisfaz a tabela verdade.

Exemplo: Determine uma fórmula que satisfaça a tabela verdade abaixo:

A	B	?	
0	0	1	
0	1	0	**A+B’**
1	0	0	**A’+B**
1	1	1	

Resposta: Expressão obtida **(A+B’) . (A’+B)**

Observação: as expressões obtidas **(A’.B’) + (A.B)** e **(A+B’) . (A’+B)** são equivalentes, isto é, possuem o mesmo resultado lógico: (A’.B’) + (A.B) = (A+B’) . (A’+B)

Exercício5 – Encontre a Forma Normal Conjuntiva (FNC) das seguintes funções representadas pelas tabelas verdade abaixo:

a)

A	B	?
0	0	0
0	1	0
1	0	1
1	1	1

b)

A	B	C	?
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Exercício6 – Encontre a FNC das funções do exercício 1. Após verifique a equivalência entre a FND e a FNC de cada uma das funções booleanas.

5.4. Exemplo de aplicação da Álgebra Booleana

Um exemplo de uma aplicação da Álgebra de Boole: considere um sistema de segurança de uma loja em um shopping. Há um sensor de contato que, ligado, (**on**, **V** ou **1**), indica que a porta está fechada; e outro sensor infravermelho que, ligado, indica que não há pessoas ou coisas se movendo no interior da loja. Há, também, um alarme que é acionado quando um dos dois sensores é desligado. Isto é, basta um único sensor ser desativado para soar o alarme. Denomine cada sensor pelos símbolos **A** e **B**,

A = "sensor de contato"

B = "sensor infravermelho"

A tabela-verdade para a função alarme, **f(A,B)**, é dada por

A	B	f(A,B)
0	**0**	1
0	**1**	1
1	**0**	1
1	**1**	0

onde **0** e **1** significam **desligado** e **ligado**, respectivamente.

A tabela-verdade é um excelente instrumento para a especificação da função alarme, em particular, e de funções lógicas, em geral. Mas essa função também pode ser representada através de uma expressão booleana na sua forma normal, ou através de portas lógicas.

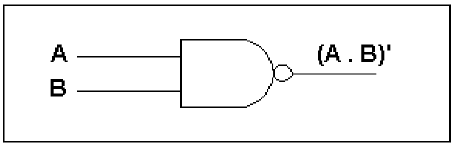
A função alarme, acima, pode ser escrita na sua forma normal disjuntiva:

f(A,B) = A'.B'+ A'.B + A.B'

Sua tabela-verdade é construída da seguinte maneira:

A	B	A.B	A'.B'+ A'.B + A.B'
0	**0**	0	1
0	**1**	0	1
1	**0**	0	1
1	**1**	1	0

Simplificando essa função com o uso das propriedades da Álgebra Booleana, obtemos a seguinte função **f(A,B) = (A.B)'** , que pode ser especificada através do seguinte diagrama lógico:



isto é, através da porta lógica **NAND**.

5.5. Simplificação de Funções Booleanas

Podemos usamos as propriedades da Álgebra Booleana para **minimizar ou simplificar** algebricamente uma função.

Associativa (ASSOC)	(a + b) + c = a + (b + c) = a + b + c	(a . b) . c = a . (b . c) = a . b . c
Comutativa (COMUT)	a + b = b + a	a . b = b . a
Distributiva (DISTRIB)	a + (b . c) = (a + b) . (a + c)	a . (b + c) = (a . b) + (a . c)
Identidade e Idempotente (IDE)	a + 0 = a	a . 0 = 0
	a + 1 = 1	a . 1 = a
	a + a' = 1	a . a' = 0
	a + a = a	a . a = a
Absorção (ABS)	a + (a . b) = a	a. (a + b) = a
	a + (a . b') = a	a. (a + b') = a
	a + (a' . b) = a+b	a. (a' + b) = a.b
De Morgan (MORGAN)	(a+b)' = a' . b'	(a.b)' = a' + b'

Exemplo - Simplificar a função abaixo:

f= a'b'+a'b+ab'
= b'(a'+a) + a'b DISTRIB
= b'.(1) + a'b IDE
= b' + a'b IDE
= b'+a' ABS
= (b.a)' MORGAN

Exercício7 – Simplifique as funções abaixo usando a propriedade **Identidade e Idempotente (IDE)**:

a) b.0.1 =
b) a+1+a' =
c) b'.1 =
d) a.1.a' =
e) b+0+b'=
f) c'+0+c=

Exercício8 – Simplifique as funções abaixo usando a propriedade **Distributiva (DISTRIB)**:

a) x(y+z)=

b) a(b+c')=

c) a'(b'+a)=

d) ab(c+d)

e) bc+ac=

f) a'b' + ab'=

g) abc+ade=

h) xz+x'z

i) a'b'c+a'b'c'=

Exercício9 – Simplificar as funções abaixo. Após, verificar a equivalência com a tabela verdade ou Diagramas de Venn.

a) f= ab+ab'

b) f= ab+a'b

c) f=a'b+a'b'

d) f=abc+ab'c

e) f=ab+a'b+a'b'

Exercício10 – Simplificar as funções abaixo usando a propriedade **DeMorgan**:

a) (a+b)'.(b+c)'=

b) (ab+c)'=

c) (ac+c')'=

d) [(p+q)'.rp'] '=

e) [(p+q)'+r'+q]'=

f) [(pq)'.r.p']'=

Exercício11 – Simplificar as funções abaixo usando a propriedade **Absorção (ABS):**

a) $x.(x+y) =$

b) $x'.(x'+y) =$

c) $(y+x').x' =$

d) $x.(x+y') =$

e) $x'.(x'+y') =$

f) $xy'.(xy'+z) =$

g) $y'x.(x'z+xy')=$

h) $x.(x'+y) =$

i) $x'.(x+y) =$

j) $x.(x'+y') =$

k) $x'.(x+y') =$

l) $x.y.((x.y)'+z) =$

m) $x'.y.(z.w+(x'.y)') =$

n) $(x+y)'.(x+y+z) =$

o) $(x'+y).((x'+y)'+z) =$

Exercício12 – Simplificar as funções abaixo usando a propriedade **Absorção (ABS):**

a) x+xy =

b) xy+x =

c) x+xy’ =

d) xz+xzy =

e) xz’+xz’y =

f) x+xzy =

g) x+xz’y =

h) x’+x’y =

i) x’y+x’yz =

j) yx+xyz =

k) y’x+zxy’ =

l) x+x’y =

m) x’+xy =

n) x’+xy’ =

o) (xy)’+xyz =

p) xy+(xy)’zw =

Exercício13 – Simplificar as funções abaixo usando as propriedades booleanas:

a) f= ab+ab’+a’b

b) f= ab+a’b’+ab’

c) f=ab’c+abc’+abc

d) f= abc+ab’c+a’bc+abc’+a’bc’

e) f= a’b’c’+a’bc+a’bc’+ab’c’+abc’

f) f= abc+ab’c’+ab’c

g) f= ab’+a’b+a’b’

h) f= ab’c+ab’c’+a

Exercício14 – Encontre a FND correspondente a seguinte tabela verdade e a simplifique. Após, desenhe o circuito lógico e verifique a equivalência:

a)

A	B	C	?
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

b)

A	B	C	?
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

6. Equivalências Lógicas com Diagramas de Venn

É possível estabelecer uma relação entre a Lógica Proposicional, a Teoria dos Conjuntos e a Álgebra Booleana, no que tange as operações de conjunção, disjunção e negação.

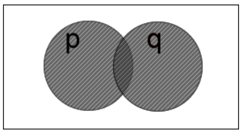
Uma ***fórmula proposicional*** e uma ***expressão na álgebra dos conjuntos***, são correspondentes se substituirmos " $\underline{\sim , \vee , \wedge , \Leftrightarrow , \mathbf{F} , \mathbf{V}}$ " por " **'**, $\underline{\cup , \cap , = , \varnothing , \mathbf{U}}$ " ou, na álgebra booleana " $\underline{', + , \bullet , = , \mathbf{0} , \mathbf{1}}$ ", Considerando-se **a**, **b** e **c** como variáveis proposicionais ou de conjuntos ou booleanas, respectivamente, tem-se o exemplo abaixo.

$\sim(\sim \mathbf{a} \vee (\mathbf{b} \wedge \mathbf{c}))$ corresponde a $(\mathbf{a}' \cup (\mathbf{b} \cap \mathbf{c}))'$ ou a $(\mathbf{a}' + (\mathbf{b} . \mathbf{c}))'$

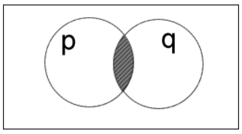
Assim, os DIAGRAMAS DE VENN podem ser usados para verificar a equivalência entre fórmulas proposicionais.

Exemplos de representação gráfica de funções booleanas:

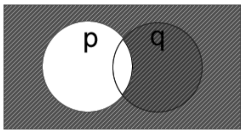
a) p + q



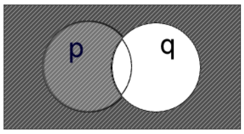
b) p . q



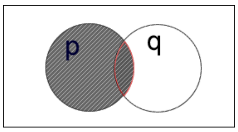
c) p´



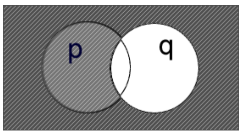
d) q´



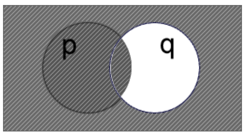
e) p + q´



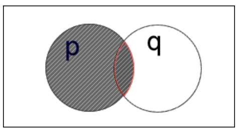
ou



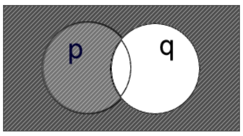
=



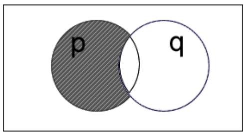
f) p . q´



e



=



Exercício15 – Desenhar o Diagrama de Venn das seguintes expressões:

a) P + (P.Q)
b) P´ . Q´
c) Q´ + (P.Q)
d) P + (Q´.P´)
e) P´. Q

Exercício16 - Verifique a equivalência lógica das seguintes expressões, usando Diagramas de Venn:

a) (P + Q) . Q´ ≡ Q´ . P
b) (P + Q) . P´ ≡ P´ . Q
c) P . Q + Q´ ≡ Q´ + P
d) (P´+Q) . Q´ ≡ Q´ . P´
e) (P . Q) + P´ ≡ P´ + Q
f) (P+Q) . (P+Q´) ≡ P

Exercício17 – Desenhar o Diagrama de Venn das seguintes expressões:

a) A . B . C
b) A . (B+C)
c) A + B + C´
d) A + (B´.C)
e) (A . B . C)´
f) (P´ + Q´) . R
g) (P´.Q) + P + Q´ + R
h) (P´.Q´) + (R.P) + (Q.R´)

Exercício18 - Verifique a equivalência lógica das seguintes expressões, usando Diagramas de Venn:

a) A . (B+C) ≡ (A.B)+(A.C)
b) A + (B.C) ≡ (A+B).(A+C)

7. Referências

Esse material foi elaborado baseado na seguinte bibliografia:

1) SOUZA, João Nunes. **Lógica para Ciência da Computação: uma introdução concisa.** Rio de janeiro: Elsevier, 2008.

2) NOLT, John; ROHATYN, Dennis. ***Lógica*.** Makron Books, 1991.

3) GERSTING, Judith L. ***Fundamentos Matemáticos para Ciência da Computação***. 3.ed. LTC, 1995.

4) POFFAL, Cristiana Andrade; RENZ, Sandra Pacheco. **Fundamentos de Lógica Matemática.** Porto Alegre: La Salle, 2003.

5) COPI, Irving M. **Introdução à lógica**. 2 ed. São Paulo:Mestre Jou, 1978.

6) DAGHLIAN, Jacob. **Lógica e álgebra de Boole**. 4. ed. São Paulo: Atlas, 1995