Acta Scientific COMPUTER SCIENCES

Volume 6 Issue 6 June 2024

Review Article

Boolean Algebras and their Applications

Rasha Ahmed Hamid Ahmed* and Um Salama Ahmed Abd Alla Alemam

Department of Mathematics, University College in Addair, Jazan University, K.S.A

***Corresponding Author:** Rasha Ahmed Hamid Ahmed Department of Mathematics, University College in Addair, Jazan University, K.S.A.

Received: April 29, 2024

Published: May 07, 2024

Abstract

Expression Boolean algebra expressions are statements that make use of logical operators such as AND, OR, NOT, XOR, etc. These logical statements can only have two outputs, either true or false. In digital circuits and logic gates "1" and "0" are used to denote the input and output conditions. For example, if we write A OR B it becomes a Boolean.

Keywords: Boolean Algebra; Logical Operators; Binary Operations; Unary Operation; Digital Circuits; Logic Gates; Duality; Boolean Expression; Isomorphic; Simplification; Complement; Boolean Variables; Literal; Truth Table

Introduction

In the middle of the nineteenth century the English mathematician George Boole introduced the algebras which arenow named after him. These algebras give a mathematical basis to logical reasoning they are also the basis of electronic computer design via the physical implementation of Boolean operations. In this paper we introduce Boolean algebras and their operations, and we define Boolean functions which specify the operation performed by a Boolean circuit, or the truth value of a logical formula. Finally, we show how to represent a Boolean function by a polynomial expression built up using the basic operations of Boolean algebras.

Boolean algebra is a branch of algebra dealing with logical operations on variables. There can be only two possible values of variables in boolean algebra, i.e. either 1 or 0. In other words, the variables can only denote two options, true or false. The three main logical operations of boolean algebra are conjunction, disjunction, and negation.

In elementary algebra, mathematical expressions are used to mainly denote numbers whereas, in boolean algebra, expressions represent truth values. The truth values use binary variables or bits "1" and "0" to represent the status of the input as well as the output. The logical operators AND, OR, and NOT form the three basic boolean operators. In this article, we will learn more about the definition, laws, operations, and theorems of boolean algebra [6].

Very special Boolean algebra, denoted by IB, is the Boolean algebra containing only the two elements 0 and 1. (It is usually referred to as The Boolean algebra.) The sum, product and complement operations on this two element algebra are described in the 0 and 1, not forgetting that 1 + 1 = 1!

Boolean Algebras

Boolean algebra is defined as a set contain- ing two distinct elements 0 and 1, together with binary operations +, ·, and a unary operation, having the following properties: The definition of an abstract Boolean algebra gives the axioms for an 1.1 [4].

Definition (2-1):

The Boolean algebras so far have all been concrete, consisting of bit vectors or equivalently of subsets of some set. Such a Boolean algebra consists of a set and operations on that set which can be *shown* to satisfythe laws of Boolean algebra.

Instead of showing that the Boolean laws are satisfied, we can instead postulate a set *X*, two binary operations on *X*, and one unary operation, and *require* that those operations satisfy the laws of Boolean algebra. The elements of *X* need not be bit vectors or subsets but can be anything at all. This leads to the more general *abstract* definition.

A *Boolean algebra* is any set with binary operations Λ and ∨ and a unary operation ¬ thereonsatisfying the Boolean laws.

Definition (2-2)

We say the system mathematics (B, +, ., `, 0,1)

Such that B is set contains at least two different elements. defined two binary operations (+,.) and a unary operation (`).

0≠1 two elements.

B is Boolean Algebra if is satisfy Boolean lows

Boolean expression

A logical statement that results in a Boolean value, either be True or False, is a Boolean expression. Sometimes, synonyms are used to express the statement such as 'Yes' for 'True' and 'No' for 'False'. Also, 1 and 0 are used for digital circuits for True and False, respectively .

Boolean expressions are the statements that use logical operators, i.e., AND, OR, XOR and NOT. Thus, if we write X AND Y = True, then it is a Boolean expression.

Boolean algebra terminologies

Now, let us discuss the important terminologies covered in Boolean algebra.

- **Boolean Algebra:** Boolean algebra is the branch of algebra that deals with logical operations and binary variables .
- **Boolean Variables:** A Boolean variable is defined as a variable or a symbol defined as a variable or a symbol,generally an alphabet that represents the logical quantities such as 0 or 1 .
- **Boolean Function:** A Boolean function consists of binary variables, logical operators, constants such as 0 and1, equal to the operator, and the parenthesis symbols .
- **Literal:** A literal may be a variable or a complement of a variable .
- **Complement:** The complement is defined as the inverse of a variable, which is represented by a bar over thevariable .
- **Truth Table:** The truth table is a table that gives all the possible values of logical variables and the combination of the variables. It is possible to convert the Boolean equation into a truth table. The number of rows in the truth table should be equal to 2^n, where "n" is the number of variables in the equation. For example, if a Boolean equation consists of 3 variables, then the number of rows in the truth table is 8. (i.e.,) $2^3 = 8$.

Definition (2-5)

Duality

Duality is the quality or state of having two different or opposite parts or elements, or a difference betweentwo opposite things.

If E is Boolean expression the duality expressions $E^d$is expression find from change all adding operation ofmultiplying operation and O of I, I of O.

Example 2-1

If E = $x\acute{z} + x.0 + \acute{x}.1$. Find E^d

Solution

$E^d = (x + \acute{z}) + (x + 1) + (\acute{x} + o)$

Theorem 2-1: If B Boolean Algebra and a, b ∈ B such that

- (i) a+ a = a, (ii) a. a = a
- (i) a + 1 = a, (ii) a. 0 = 0
- (i) a + a b = a, (ii) a (a + b) = a
- (i) (a) = a, (ii) a + $\acute{a}$ = 1
- (i) $\acute{0}$ = 1, (ii) $\acute{1} = 0$
- (i) $(a \acute{+} b) = \acute{a}\acute{b}$, (ii) $(ab)\acute{} = \acute{a} + \acute{b}$

Theorem [1] (2-2):If B is Boolean Algebra and a ∈B → $\acute{a}$ is ungueneness.

Definition (2-6):

If B is Boolean Algebra a≠0 ∈B

We say a is a tom if ∀ b ∈B $(0 \le b \le a \ \rightarrow b = 0 \ v \ b = a)$

Definition (2-6):

If $B_1$and $B_2$are Boolean Algebra say $B_1$and B_2 are isomorphic if we find

$\emptyset: B_1 \rightarrow B_2 \ \forall \ a, b \in B_1$

$\emptyset(\acute{a}) = ((\emptyset(a)), \ \emptyset(ab) = \emptyset(a).\emptyset(b), \ \emptyset(a + b) = \emptyset(a) + \emptyset(b)$

Then say is isomorphic between $B_1$and B_2.

Boolean Functions

Definition (3-1)

If n ∈ $z^+$and $B_2^n = \{a_1, a_2, \dots \dots \dots, a_n : a_i \in B2\}$ called application $B_2^n \rightarrow B_2$ *is* Boolean Function.

Example (3-1)

Find truth table of Boolean function, f (x, y) = x y + $\acute{x}$

Solution

X	Y	$\acute{x}$	Xy	F (x, y) = x y + $\acute{x}$
1	1	0	1	1
1	0	0	0	0
0	1	1	0	1
0	0	1	0	1

Definition (3-2)

If f, g: $B_2{}^n \rightarrow B_2$ are Boolean Function we say f and g are equivalent and writes F = g, if we find sametruth table [4].

Example (3-2)

Prove that the Boolean functions

f (x, y, z) = x y, g (x, y, z) = x y (x $\acute{z}$ + y z) is equivalent.

Solution

g (x, y, z) = x y (x$\acute{z}$ + yz)

= x y x $\acute{z}$ + xyy

= x y $\acute{z}$ + xyz

= x y ($\acute{z}$ + z) = x y = f (x, y, z)

then f (x, y, z) = x y, g (x, y, z) = x y (x $\acute{z}$ + yz) are equivalent.

Definition (3-3)

If x_1, x_1, x_2,......................................, x_n → Boolean function

(i) ∀ 1 ≤ i ≤ n, x_i, $\acute{x}_i$ called literal.

(ii) y_1, ..., y_n, = 1 : x_i, = y_i or y_i = $\acute{x}_i$ ∀ 1 ≤ i ≤ n Called minterm.

(iii) If f written the form sum minterms. This from is called (complete sum of products) denoted (C S P).

Algorithm [4] (3 – 1)

If x_1, x_1, x_2, ..., x_n → Boolean function we put f (x, y, z) in the from complete sum of products C SP per from the following steps.

- Find truth table off.
- Find where value of f = 1.
- Find the minterm of $y_1 y_2$, ... y_n where x_i, = y_i = 1 (Or) $\acute{x}_i$ = y_i = 0.
- Find C S P is sum minterms of step (3)

Example (3-3)

Write the function, f (x, y, z) = (x + y) $\acute{z}$ in the from complete sum of products C S P. Solution

X	Y	Z	x + y	$\acute{z}$	(x + y) $\acute{z}$	Minterm
1	1	1	1	0	0	
1	1	0	1	1	1	x y $\acute{z}$
1	0	1	1	0	0	
1	0	0	1	1	1	x $\acute{y}\acute{z}$
0	1	1	1	0	0	
0	1	0	1	1	1	$\acute{x}$ y $\acute{z}$
0	0	1	0	0	0	
0	0	0	0	1	0	

f (x, y, z) = (x + y) $\acute{z}$ in the from C S P = x y $\acute{z}$ + x $\acute{y}\acute{z}$ + $\acute{x}$ y $\acute{z}$

Logarithm [4] (3 – 2)

If x_1, x_1, x_2, ..., x_n → Boolean function we put f (x, y, z) in the from C S P perform the followingsteps

Find truth table of f . 2.Find where value of f = 0.

Find the maxterm of y_1 + y_2 +, ...+y_n

∀ x_i, = y_i, if x_i, = 0 Or $\acute{x}_i$ = y_i, if x_i, = 1.

Find C P S is product maxterm of step (3)

Example (3-4)

Write the function, f (x, y, z) = (x + y) $\acute{z}$ in the from complete products of sum C P S Solution.

X	Y	Z	x + y	$\acute{z}$	(x + y) $\acute{z}$	Maxterm
1	1	1	1	0	0	$\acute{x}$ + $\acute{y}$ + $\acute{z}$
1	1	0	1	1	1	
1	0	1	1	0	0	$\acute{x}$ + y + $\acute{z}$
1	0	0	1	1	1	
0	1	1	1	0	0	x + $\acute{y}$ + $\acute{z}$
0	1	0	1	1	1	
0	0	1	0	0	0	x + y + $\acute{z}$
0	0	0	0	1	0	x + y +z

f (x, y, z) = (x + y) $\acute{z}$ in the from C P S = ($\acute{x}$ + $\acute{y}$ + $\acute{z}$) ($\acute{x}$ + y + $\acute{z}$) (x + $\acute{y}$ + $\acute{z}$) (x + y + $\acute{z}$) (x + y +z)

The minimal Boolean algebra (3-2)

A very special Boolean algebra, denoted by IB, is the Boolean algebra containing only the two elements 0and 1. (It is usually referred to as The Boolean algebra.) The sum, product and complement operations on this two element algebra are described in the following table.

x	y	$x+y$	xy	$\overline{x}$
0	0	0	0	1
0	1	1	0	1
1	0	1	0	0
1	1	1	1	0

The function f (x, y) = $\bar{x}y + \bar{y}$ = is a polynomial function. Its values are given by the following table

x	y	$\overline{x}$	$\overline{x}y$	$\overline{y}$	$\overline{x}y+\overline{y}$
0	0	1	0	1	1
0	1	1	1	0	1
1	0	0	0	1	1
1	1	0	0	0	0

Laws of Boolean Algebra [7]

There are six types of Boolean algebra laws.

They are

- Commutative law
- Associative law
- Distributive law
- AND law
- OR law
- Inversion law

Those six laws are explained in detail here.

Commutative law

Any binary operation which satisfies the following expression is referred to as a commutative operation. Commutative law states that changing the sequence of the variables does not have any effect on the output of a logic circuit [7].

- A. B = B. A
- A + B = B + A

Associative law

It states that the order in which the logic operations are performed is irrelevant as their effect is the same [7].

- (A. B). C = A. (B. C)
- (A + B) + C = A + (B + C)

Distributive law

Distributive law states the following conditions [7]

- A. (B + C) = (A. B) + (A. C)
- A + (B. C) = (A + B). (A + C)

and law

These laws use the AND operation.
Therefore they are called AND laws.

- A. 0 = 0
- A. 1 = A
- A. A = A

Associative Law

It states that the order in which the logic operations are performed is irrelevant as their effect is the same.

- (A. B). C = A . (B . C)
- (A + B) + C = A + (B + C)

Distributive Law

Distributive law states the following conditions:

- A. (B + C) = (A. B) + (A. C)
- A + (B. C) = (A + B) . (A + C)

AND Law

These laws use the AND operation. Therefore they are called AND laws.

- A .0 = 0
- A .1 = A
- A. A = A
- $A.\bar{A} = 0$

OR Law

These laws use the OR operation. Therefore they are called OR laws.

- A + 0 = A
- A + 1 = 1
- A + A = A
- $A + \bar{A} = 1$

Inversion Law

In Boolean algebra, the inversion law states that double inversion of variable results in the original variable itself.

- $\bar{\bar{A}} = A$

Simplification boolean function

Boolean Algebra Simplification and how to simplify Boolean algebra expressions using some basic rules applied to their variables, literals and terms. Boolean Algebra simplification is not that difficult to understand if you realise that the use of the symbols or signs of: "+" and "." represent the operation of logical functions.

Logical functions test whether a condition or state is either *TRUE* or *FALSE* but not both at the same time. Sodepending on the result of that test, a digital circuit can then decide to do one thing or another [3].

Definition (5-1)

Let f, g two Boolean functions w-ritten in the form product of sums say that f is simplest of g if

- Number of term f less than number of term g
- Number of literal f less than number of literal gDefinition (5-2)

Let S the set of all equivalent functions each of which in the form of sum products . If f is simplest function say that f in the form (minimial sum of products) [4].

Having established the switching operation of the AND, OR, and NOT functions. We can now look at simplifying some basic Boolean Algebra expressions to obtain a final expression that has the minimum number of terms [3].

Example (5-1)

First let us start with something simple such as

The Boolean Expression: A. (A + B) Multiplying out the brackets gives us

	A. (A+B)	Start
Multiply:	A.A + A. B	Distributive law
But:	A.A = A	Idempotent Law
Then:	A + A. B	Reduction
Thus:	A. (1 + B)	Annulment Law
Equals to:	A	Absorption Law

Example (5-2)

This time we will use three Boolean terms, A, B, and C and use the same Boolean algebra simplification rules as before.

Boolean Expression: (A + B) (A + C)

Again, multiplying out the brackets gives us

	((A+B) (A+C)	
Multiply:	A.A + A.C + A.B + B.C	Distributive law
But:	A.A = A	Idempotent Law
Then:	A + A.C + A.B + B.C	Reduction
However:	A + A.C = A	Absorption Law

Boolean algebra theorems.

The two important theorems which are extremely used in Boolean algebra are De Morgan's First law and DeMorgan's second law. These two theorems are used to change the Boolean expression. This theorem basically helps to reduce the given Boolean expression in the simplified form. These two De Morgan's laws are used to change the expression from one form to another form. Now, let us discuss these two theorems in detail.

De Morgan's First Law (6-1)

De Morgan's First Law states that (A.B)' = A'+B'.

The first law states that the complement of the product of the variables is equal to the sum of their individual complements of a variable.

The truth table that shows the verification of De Morgan's First law is given as follows

A	B	A′	B′	(A.B)′	A′+B′
0	0	1	1	1	1
0	1	1	0	1	1
1	0	0	1	1	1
1	1	0	0	0	0

The last two columns show that (A.B)′ = A′+B′.

Hence, De Morgan's First Law is proved.

De Morgan's Second Law:

De Morgan's Second law states that (A+B)′ = A′. B′.

The second law states that the complement of the sum of variables is equal to the product of their individual complements of a variable.

The following truth table shows the proof for De Morgan's second law.

A	B	A′	B′	(A+B)′	A′. B′
0	0	1	1	1	1
0	1	1	0	0	0
1	0	0	1	0	0
1	1	0	0	0	0

The last two columns show that (A+B)′ = A′. B′.Hence, De Morgan's second law is proved.

The other theorems in Boolean algebra are complementary theorem, duality theorem, transposition theorem, redundancy theorem and so on. All these theorems are used to simplify the given Boolean expression. The reduced Boolean expression should be equivalent to the given Boolean expression.

Question: Simplify the following expression:

$c + \bar{BC}$

Solution:

Given:

$C + \bar{BC}$

According to Demorgan's law, we can write the ab

$C + (\bar{B} + \bar{C})$

From Commutative law:

$(C + \bar{C}) + \bar{B}$

From Complement law

$1 + \bar{B} = 1$

Therefore,

$C + \bar{BC} = 1$

Question 2: Draw a truth table for A(B+D).

Solution: Given expression A(B+D).

A	B	D	B+D	A(B+D)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

Logic network

Logic network is switching network but we use gates.

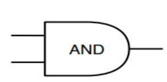
We find some gates:

- NOT gate
- OR gate
- AND gate
- NOR gate
- NAND gate

Core boolean operators [5] (7 – 1)

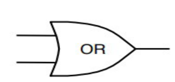
AND $\wedge$

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1



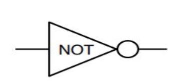
OR $\vee$

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1



NOT $\neg$

A	NOT A
0	1
1	0



Universal gates (7-2)

Universality of nortruth table

A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

- NOT: A nor A
- AND: (A nor A) nor (B nor B)
- OR: (A nor B) nor (A nor B)Universality of NAND

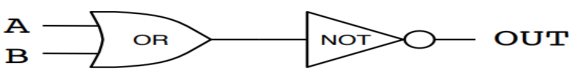
Truth table

A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

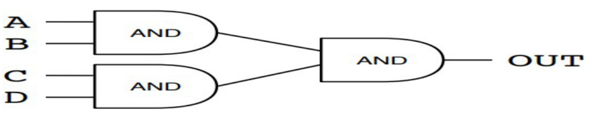
- NOT: A nand A
- AND: (A nand B) nand (A nand B)
- OR: (A nand A) nand (B nand B)

From Boolean expressions to circuits (4-4)

- Operation: not (A or B)Circuit :



- 4 inputs (A, B, C, D), output 1 iff all inputs are 1
- Operation: (A and B) and (C and D)
- Circuit:



- **FIRST LAW**

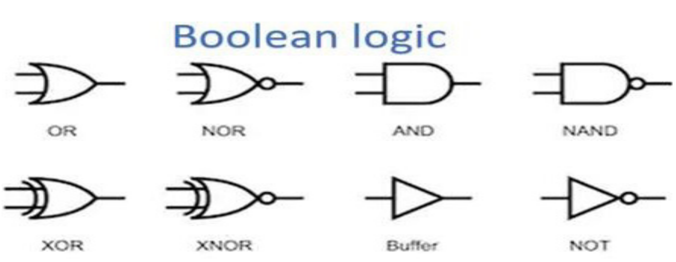
$\overline{A.B} = \overline{A} + \overline{B}$

Figure: De Morgan's First Law

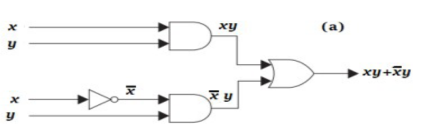
- **SECOND LAW**

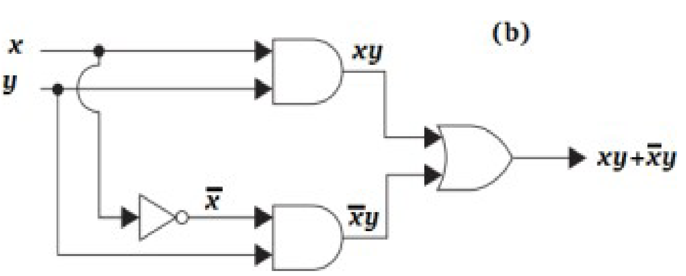
$\overline{A + B} = \overline{A}.\overline{B}$

Figure: De Morgan's Second Law



Note that, output from a gate may be used as input by one or more other elements, as shown in the followibg. Both drawings (a) and (b) depict the circuit that produces the output $xy + \bar{x}y$.



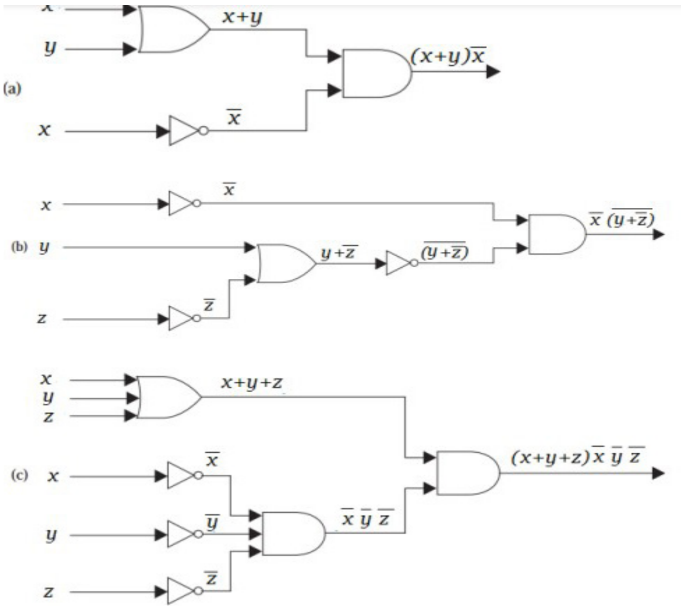


Example (7-1)

Construct circuits that produce the following output

(a) $(x + y)\bar{x}$ (b) $\bar{x}\overline{(y + \bar{z})}$ (c) $(x + y + z)(\bar{x}\bar{y}\bar{z})$

Solution



Frequently Asked Questions on Boolean Algebra [7].

Q1 What is meant by Boolean algebra?

In Mathematics, Boolean algebra is called logical algebra consisting of binary variables that hold the values 0 or 1, and logical operations [17].

Q2 What are some applications of Boolean algebra?

In electrical and electronic circuits, Boolean algebra is used to simplify and analyze the logical or digitalcircuits.

Q3 What are the three main Boolean operators?

The three important Boolean operators are:AND (Conjunction) OR (Disjunction)NOT (Negation)

Q4 Is the value 0 true or false?

In Boolean logic, zero (0) represents false and one (1) represents true. In many applications, zero is interpreted as false and a non-zero value is interpreted as true.

Q5 Mention the six important laws of Boolean algebra.The six important laws of Boolean algebra are:

Commutative lawAssociative law Distributive law Inversion law AND law OR law

Main results

- Boolean algebra is a branch of algebra. It differs from elementary algebra in two ways. First, the values of the variables are the truth values *true* and *false*, usually denoted 1 and 0, whereas in elementary algebra the values of the variables are numbers.
- Second, Boolean algebra uses logical operators such as conjunction (*and*) denoted as $\wedge$, disjunction (*or*) denoted as $\vee$, and the negation (*not*) denoted as $\neg$. Elementary algebra, on the other hand
- Uses arithmetic operators such as addition, multiplication, subtraction, and division. Boolean algebra is therefore a formal way of describing logical operations, in the same way that elementaryalgebra describes numerical operations.

Conclusion

In this scientific paper, I discussed the concept of Boolean Algebras and its characteristics and some of its applications.

And, I have sought to provide assistance to students through the explanation in the text and examples to understand the definition of Boolean algebra, its laws, and some of its applications.

Bibliography

1. KH Rosen. "Discrete Mathematics And Its Applications, 7th Edition, McGraw-Hill (2007).
2. RP Grimaldi. "Discrete and Combinatorial Mathematics: AnappliedIntroduction". Addison-Waesly (1989).
3. Web sites dedicated to Graph Theory And Its Applications on the intern4- Discrete Mathematics, written by Marouf Samhan and Ahmed Sharari.
4. J Eldon Whitesitt. "Boolean Algebra and Its Applications.
5. Stone Marshall H. "Boolean algebras and their application to topology". *Proceedings of the National Academy of Sciences* 20.3 (1934): 197-202.
6. Nies, André. "Effectively dense Boolean algebras and their applications". *Transactions of the American Mathematical Society* 352.11 (2000):4989-5012.
7. Whitesitt J Eldon. "Boolean algebra and its applications". Courier Corporation (2012).