

Algorithms for Verifying Variants of Boolean Algebra Equations and Expressions

O.E. Oguike

Department of Computer Science
University of Nigeria
Nsukka, Nigeria
osondu.oguike@unn.edu.ng

ABSTRACT

The word, Boolean, was derived from the name of a British mathematician, George Boole, as a result of his classical work on logic. Boolean algebra can be defined as a set, whose members have two possible values, with two binary operators and one unary operator, satisfying the properties of commutativity, associativity, distributivity, existence of identity and complement. Boolean algebra has important applications to the design of computer hardware and software. Techniques, like Karnaugh maps, Boolean algebra theorems and laws can be used to simplify and reduce complex Boolean algebra expressions, while truth table can be used to confirm that the reduced Boolean algebra expression is the same as the original, complex Boolean algebra expression. Generating the truth table manually is tedious, especially when the Boolean algebra equation or expression has many Boolean variables. This paper presents three variants of Boolean algebra, and novel algorithms that can be used to verify Boolean algebra equations and evaluate Boolean algebra expressions.

Keywords: Boolean algebra, logic, boolean algebra expression, boolean algebra equation, truth table, logical values, binary numbers, set of sets, tautology.

African Journal of Computing & ICT Reference Format:

O.E. Oguike (2013). Algorithms for Verifying Variants of Boolean Algebra Equations and Expressions.
Afr J. of Comp & ICTs. Vol 6, No. 5. Pp1-8

1. INTRODUCTION

Any set that has two distinct values, which satisfies the following properties, commutativity, associativity, distributivity, existence of an identity element, and existence of a complement can be called a Boolean algebra. An example of Boolean algebra includes the following: the set of logical values, with two binary operators, $\wedge$, $\vee$ and one unary operator, $\sim$. Another example of Boolean algebra is the set of binary numbers, with one unary operator and two binary operators, $\cdot$ and $+$, defined in table 1 below.

Table 1: Binary Operators of Boolean Algebra

$\cdot$	0	1
0	0	0
1	0	1

$+$	0	1
0	0	1
1	1	1

The unary operator, $'$, is defined as: $'0 = 1$, $'1 = 0$.

Another example of Boolean algebra is a set, whose elements are the universal set and empty set, together with these two binary operators, $\cap$ and $\cup$, and one unary operator, c .

Each of the three sets satisfies all the properties of Boolean algebra.

Suppose X, Y and Z are members of the set of logical values, i.e. the two possible values of the variables are True and False, with these two binary operators, $\wedge$ denoting AND, $\vee$ denoting OR and $\sim$ denoting NOT, the following properties hold:

- $X \vee Y = Y \vee X$ Commutative
- $X \wedge Y = Y \wedge X$
- $X \vee (Y \vee Z) = (X \vee Y) \vee Z$ Associative
- $X \wedge (Y \wedge Z) = (X \wedge Y) \wedge Z$
- $X \vee (Y \wedge Z) = (X \vee Y) \wedge (X \vee Z)$ Distributive
- $X \wedge (Y \vee Z) = (X \wedge Y) \vee (X \wedge Z)$
- $X \vee F = X$ Identity
- $X \wedge T = X$
- $X \vee \sim X = T$ Complement
- $X \wedge \sim X = F$

Similarly, suppose X, Y and Z are members of the set of binary numbers, i.e. the two distinct values of the elements of the set are 0 and 1, with these two binary operators, $\cdot$ and $+$ as defined in table 1, and one unary operator, $'$, the following properties of Boolean algebra hold:

- $X \cdot Y = Y \cdot X$ Commutative
- $X + Y = Y + X$
- $X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z$ Associative

- $X+(Y+X) = (X+Y)+Z$
- $X.(Y+Z) = X.Y + X.Z$ Distributive
 $X+(Y.Z) = (X+Y).(X+Z)$
- $X+0 = X$ Identity
 $X.1 = X$
- $X+'X = 1$ Complement
 $X.'X = 0$

Furthermore, suppose X, Y and Z are members of a set, whose two distinct values are the universal set and the empty set, with these two binary operators, $\cap$ and $\cup$, and one unary operator, c , the following properties of Boolean algebra hold:

- $X \cap Y = Y \cap X$
 $X \cup Y = Y \cup X$ Commutative
- $X \cup (Y \cap Z) = (X \cup Y) \cap Z$ Associative
 $X \cap (Y \cup Z) = (X \cap Y) \cup Z$
- $X \cup (Y \cap Z) = (X \cup Y) \cap (X \cup Z)$ Distributive
 $X \cap (Y \cup Z) = (X \cap Y) \cup (X \cap Z)$
- $X \cup \phi = X$
 $X \cap U = X$ Identity
- $X \cup X^c = U$ Complement
 $X \cap X^c = \phi$

2. SURVEY OF RELATED LITERATURE

In [1], the author presented Boolean algebra as a set of binary numbers, with two binary operators and one unary operator, satisfying some properties. It did not identify Boolean algebra as any of the other two sets that this paper has identified. On the applications of Boolean algebra, [2] identified an application outside computer science, psychiatry. The author pointed out that the operations of Boolean algebra could be seen to correspond to the operations of nervous system. On the other hand, [3] emphasized that the rules and laws of Boolean algebra were essential for the simplification of long and complex logic equations that were based on the AND and OR logic equations. In [4], the author identified a variant of Boolean algebra, as a set with two binary operators, union and intersection operators, the author argued that such set could be fuzzy sets; therefore, the argument was that fuzzy sets could satisfy the properties of Boolean algebra, and thus could be Boolean algebra. The author of [5] used truth table to evaluate logical expressions with the aim of comparing the result of the truth table with the arithmetic version. Two variants of Boolean algebra were identified in [6], the set of logical values and the set of binary numbers.

3. STANDARD BOOLEAN ALGEBRA THEOREMS

Consider the set of logical values, suppose X, Y and Z are members of the set, the following Boolean algebra theorems/equations hold:

- $X \vee (X \wedge Y) = X$
- $X \vee (\sim X \wedge Y) = X \vee Y$
- $(X \wedge Y) \vee (\sim X \wedge Z) \vee (Y \wedge Z) = (X \wedge Y) \vee (\sim X \wedge Z)$
- $X \wedge (X \vee Y) = X$
- $X \wedge (\sim X \vee Y) = (X \wedge Y)$
- $(X \vee Y) \wedge (X \vee \sim Y) = X$
- $(X \vee Y) \wedge (\sim X \vee Z) = X \wedge Z \vee \sim X \wedge Y$
- $(X \vee Y) \wedge (\sim X \vee Z) \wedge (Y \vee Z) = (X \wedge Y) \wedge (\sim X \vee Z)$
- $\sim (X \wedge Y) = \sim X \vee \sim Y$
- $\sim (X \vee Y) = \sim X \wedge \sim Y$

Similarly, consider the set of binary numbers, suppose X, Y and Z are members of the set, the following Boolean algebra theorems/equations hold:

- $X + (X.Y) = X$
- $X + ('X.Y) = X + Y$
- $(X.Y) + ('X.Z) + (Y.Z) = (X.Y) + ('X.Z)$
- $X.(X+Y) = X$
- $X.('X+Y) = X.Y$
- $(X+Y).(X+'Y) = X$
- $(X+Y).('X+Z) = (X.Z) + ('X.Y)$
- $(X+Y).('X+Z).(Y+Z) = (X+Y).('X+Z)$
- $'(X.Y) = 'X+'Y$
- $'(X+Y) = 'X.'Y$

Furthermore, consider the set, whose elements are the universal set and empty set. Suppose X, Y and Z are members of the set, the following Boolean algebra theorems/equations hold:

- $X \cup (X \cap Y) = X$
- $X \cup (X^c \cap Y) = X \cup Y$
- $(X \cap Y) \cup (X^c \cap Z) \cup (Y \cap Z) = (X \cap Y) \cup (X^c \cap Z)$
- $X \cap (X \cup Y) = X$
- $X \cap (X^c \cup Y) = X \cap Y$
- $(X \cup Y) \cap (X \cup Y^c) = X$
- $(X \cup Y) \cap (X^c \cup Z) = (X \cap Z) \cup (X^c \cap Y)$
- $(X \cup Y) \cap (X^c \cup Z) \cap (Y \cup Z) = (X \cap Y) \cap (X^c \cup Z)$
- $(X \cap Y)^c = X^c \cup Y^c$
- $(X \cup Y)^c = X^c \cap Y^c$

4. OTHER OPERATORS OF BOOLEAN ALGEBRA

Boolean algebra has other operators, apart from the basic two binary operators, and one unary operator. The other operators of Boolean algebra are the imply operator, $\rightarrow$, and the equivalent operator, $\leftrightarrow$.

Consider the set of logical values as Boolean algebra, suppose X and Y are members of the set, the imply operator, which is written as $X \rightarrow Y = \text{imply}(X, Y)$ can be defined as:

$$\begin{cases} FALSE, & \text{If } X = TRUE, Y = FALSE \\ TRUE, & \text{Otherwise} \end{cases} \quad (1)$$

In a similar manner, consider the set of binary numbers as Boolean algebra, suppose X and Y are members of the set, the imply operator, which is written as $X \rightarrow Y = \text{imply}(X, Y)$ can be defined as:

$$X \rightarrow Y = \begin{cases} 0, & \text{If } X = 1, Y = 0 \\ 1, & \text{Otherwise} \end{cases} \quad (2)$$

Furthermore, consider the set, whose elements are the universal set and empty set as Boolean algebra. Suppose X and Y are members of the set, the imply operator, which is written as $X \rightarrow Y = \text{imply}(X, Y)$ can be defined as:

$$X \rightarrow Y = \begin{cases} \phi, & \text{If } X = U, Y = \phi \\ U, & \text{Otherwise} \end{cases} \quad (3)$$

On the otherhand, consider the set of logical values as Boolean algebra, suppose X and Y are members of the set, the equivalent operator, which is written as $X \leftrightarrow Y$ can be defined as:

$$X \leftrightarrow Y = \begin{cases} True, & \text{If } X = Y \\ False, & \text{Otherwise} \end{cases} \quad (4)$$

Similarly, consider the set of binary numbers as Boolean algebra, suppose X and Y are members of the set, the equivalent operator, which is written as $X \leftrightarrow Y$, can be defined as:

$$X \leftrightarrow Y = \begin{cases} 1, & \text{If } X = Y \\ 0, & \text{Otherwise} \end{cases} \quad (5)$$

Finally, consider the set that consists of universal set and empty set as Boolean algebra. Suppose X and Y are members of the set, the equivalent operator, which is written as $X \leftrightarrow Y$, can be defined as:

$$X \leftrightarrow Y = \begin{cases} U, & \text{If } X = Y \\ \phi, & \text{Otherwise} \end{cases} \quad (6)$$

5. ALGORITHMS FOR EVALUATING COMPLEX BOOLEAN ALGEBRA THEOREMS/EQUATIONS

Having stated the basic Boolean algebra theorems/equations, and other Boolean algebra operators, like, imply and equivalent operators, for each type of Boolean algebra, these Boolean algebra operators will be combined with the basic Boolean algebra theorems/equations to state complex Boolean algebra theorems/equations, and novel algorithms for verifying these theorems/equations will be stated.

5.1 Theorem 5.1

Consider the set of logical values as Boolean algebra, suppose X , Y and Z are members of the set, the imply and equivalent operators can be used to state another version of Boolean algebra theorem as:

$$((X \rightarrow Y) \leftrightarrow Z) \vee (((X \rightarrow Y) \leftrightarrow Z) \wedge Y) = ((X \rightarrow Y) \leftrightarrow Z)$$

The algorithm that can be used to verify the boolean algebra theorem/equation follows below:

```

Boolean theorem5.1a(boolean X, Y, Z)
1  Read data
   1.1 Read X
   1.2 Read Y
   1.3 Read Z
2  Determine theorem5.1a
   2.1 IF X = TRUE AND Y = FALSE
       THEN
         2.1.1 first = FALSE
         ELSE
           2.1.2 first = TRUE
       2.2 IF first = Z THEN
           2.2.1 third = TRUE
           ELSE
             2.2.2 third = FALSE
       2.3 fourth = third AND Y
       2.4 IF third OR fourth = third THEN
           2.4.1 Result = "The Theorem Holds"
           ELSE
             2.4.2 result = "The Theorem does not hold"
3.  Display result

```

Similarly, consider the set of binary numbers as Boolean algebra, suppose X , Y and Z are members of the set, theorem5.1, which has been stated above can be written as follows:

$$((X \rightarrow Y) \leftrightarrow Z) + (((X \rightarrow Y) \leftrightarrow Z).Y) = ((X \rightarrow Y) \leftrightarrow Z)$$

The algorithm that can be used to verify the above boolean algebra theorem/equation follows below:

```

Boolean theorem5.1b(boolean X, Y, Z)
1.  Read data
   1.1 Read X
   1.2 Read Y

```

```

1.3 Read Z
2. Determine theorem5.1b
2.1 IF (X = 1).(Y = 0) THEN
2.1.1 first = 0
    ELSE
2.1.2 first = 1
2.2 IF first = Z THEN
2.2.1 third = 1
    ELSE
2.2.2 third = 0
2.3 fourth = third.Y
2.4 IF third + fourth = third THEN
2.4.1 Result = "The Theorem Holds"
    ELSE
2.4.2 result = "The Theorem does not hold"
3 Display result

```

Similarly, consider the set, whose elements are universal set and empty set as Boolean algebra, suppose X, Y and Z are members of the set, theorem 5.1 that uses the imply and equivalent operators can be written as:

$$((X \rightarrow Y) \leftrightarrow Z) \cup (((X \rightarrow Y) \leftrightarrow Z) \cap Y) = ((X \rightarrow Y) \leftrightarrow Z)$$

The algorithm that can be used to prove this theorem follows:

Boolean theorem5.1c(boolean X, Y, Z)

```

1. Read data
1.1 Read X
1.2 Read Y
1.3 Read Z
2. Determine theorem5.1c
2.1 IF (X = U).(Y = φ) THEN
2.1.1 first = φ
    ELSE
2.1.2 first = U
2.2 IF first = Z THEN
2.2.1 third = U
    ELSE
2.2.2 third = φ
2.3 fourth = third.Y
2.4 IF third ∪ fourth = third THEN
2.4.1 Result = "The Theorem Holds"
    ELSE
2.4.2 result = "The Theorem does not hold"
3. Display result

```

5.2 Theorem 5.2

Consider the set of logical values as Boolean algebra, suppose X, Y and Z are members of the set, the imply

and equivalent operators can be used to state another version of Boolean algebra theorem as:

$$((X \rightarrow Y) \leftrightarrow Z) \vee \neg((X \rightarrow Y) \leftrightarrow Z) \wedge Y = ((X \rightarrow Y) \leftrightarrow Z) \vee Y$$

The algorithm that will prove this theorem can be stated below:

Boolean theorem5.2a(boolean X, Y, Z)

```

1. Read data
1.1 Read X
1.2 Read Y
1.3 Read Z
2. Determine theorem5.2a
2.1 IF X = TRUE AND Y = FALSE THEN
2.1.1 first = FALSE
    ELSE
2.1.2 first = TRUE
2.2 IF first = Z THEN
2.2.1 third = TRUE
    ELSE
2.2.2 third = FALSE
2.3 fourth = third OR (NOT third AND Y)
2.4 fifth = (third OR Y)
2.5 IF fourth = fifth THEN
2.5.1 Result = "The Theorem holds"
    ELSE
2.5.2 result = "The Theorem does not hold"
3 Display result

```

Consider another variant of Boolean algebra, the set of binary numbers, suppose X, Y and Z are members of the set, theorem 5.2 can be stated as:

$$((X \rightarrow Y) \leftrightarrow Z) + ((X \rightarrow Y) \leftrightarrow Z).Y = ((X \rightarrow Y) \leftrightarrow Z) + Y$$

The novel algorithm that can be used to prove this Boolean theorem/equation can be stated as follows:

Boolean theorem5.2b(boolean X, Y, Z)

```

1. Read data
1.1 Read X
1.2 Read Y
1.3 Read Z
2. Determine theorem5.2b
2.1 IF (X = 1).(Y = 0) THEN
2.1.1 first = 0
    ELSE
2.1.2 first = 1
2.2 IF first = Z THEN
2.2.1 third = 1
    ELSE
2.2.2 third = 0
2.3 fourth = third + (third.Y)
2.4 fifth = third + Y
2.5 IF fourth = fifth

```

2.5.1 Result = “The Theorem holds”
ELSE
2.5.2 result = “The Theorem does not hold”
3 Display result

Furthermore, consider another variant of Boolean algebra, set, whose two elements are the universal set and the empty set. Suppose X, Y and Z are members of the set, theorem 5.2 can be stated as:

$$((X \rightarrow Y) \leftrightarrow Z) \cup (((X \rightarrow Y) \leftrightarrow Z) \cap Y) = ((X \rightarrow Y) \leftrightarrow Z) \cup Y$$

The algorithm that will be used to prove this theorem can be stated below:

Boolean theorem5.2c(boolean X, Y, Z)

```
1. Read data
   1.1 Read X
   1.2 Read Y
   1.3 Read Z
2. Determine theorem5.2c
   2.1. IF (X = U) ∩ (Y = ϕ) THEN
       2.1.1 first = ϕ
       ELSE
       2.1.2 first = U
   2.2 IF first = Z THEN
       2.2.1. third = U
       ELSE
       2.2.2. third = ϕ
   2.3. fourth = third ∪ (third ∩ Y)
   2.4. fifth = third ∪ Y
   2.5. IF fourth = fifth
       2.5.1 Result = “The Theorem holds”
       ELSE
       2.5.2. result = “The Theorem does not hold”
3. Display result
```

5.3 Theorem 5.3

In a similar manner, consider the set of logical values as Boolean algebra, suppose X, Y and Z are members of the set, the imply and equivalent operators can be used to state another version of Boolean algebra theorem as:

$$(((X \rightarrow Y) \leftrightarrow Z) \wedge Y) \vee (\sim((X \rightarrow Y) \leftrightarrow Z) \wedge Z) \vee (Y \wedge Z) = (((X \rightarrow Y) \leftrightarrow Z) \wedge Y) \vee (\sim((X \rightarrow Y) \leftrightarrow Z) \wedge Z)$$

The algorithm that can be used to prove the above theorem can be stated below.

Boolean theorem5.3a(boolean X, Y, Z)

```
1. Read data
   1.1 Read X
   1.2 Read Y
   1.3 Read Z
2. Determine theorem5.3a
   2.1. IF X = TRUE AND Y = FALSE THEN
       2.1.1 first = FALSE
       ELSE
       2.1.2 first = TRUE
   2.2 IF first = Z THEN
       2.2.1 third = TRUE
       ELSE
       2.2.2. third = FALSE
   2.3 fourth = third AND Y
   2.4 fifth = NOT third AND Z
   2.5 sixth = Y AND Z
   2.6 left = fourth OR fifth OR sixth
   2.7 right = fourth OR fifth
   2.8 IF left = right THEN
       2.8.1 result = “The Theorem Holds”
       ELSE
       2.8.2 result = “The Theorem does not hold”
3. Display result
```

In a similar manner, consider another variant of Boolean algebra, the set of binary numbers, suppose X, Y and Z are members of the set, theorem5.3 can be written as:

$$(((X \rightarrow Y) \leftrightarrow Z).Y) + (\sim((X \rightarrow Y) \leftrightarrow Z).Z) + (Y.Z) = (((X \rightarrow Y) \leftrightarrow Z).Y) + (\sim((X \rightarrow Y) \leftrightarrow Z).Z)$$

The algorithm that can be used to prove the above theorem follows:

Boolean theorem5.3b(boolean X, Y, Z)

```
1. Read data
   1.1 Read X
   1.2 Read Y
   1.3 Read Z
2. Determine theorem5.3b
   2.1. IF (X = 1).(Y = 0) THEN
       2.1.1 first = 0
       ELSE
       2.1.2 first = 1
   2.2 IF first = Z THEN
       2.2.1 third = 1
       ELSE
       2.2.2. third = 0
   2.3. fourth = third.Y
   2.4. fifth = 'third.Z
   2.5. sixth = Y.Z
   2.6. left = fourth+fifth+sixth
   2.7. right = fourth+fifth
   2.8. IF left = right THEN
       2.8.1 result = “The Theorem Holds”
       ELSE
```

- 2.8.2 result = “The Theorem does not hold”
3. Display result

Furthermore, consider the set that contains of the universal set and empty set as Boolean algebra, suppose X, Y and Z are members of the set, theorem 5.3 can be written as:

$$(((X \rightarrow Y) \leftrightarrow Z) \cap Y) \cup (((X \rightarrow Y) \leftrightarrow Z)^c \cap Z) \cup (Y \cap Z) = (((X \rightarrow Y) \leftrightarrow Z) \cap Y) \cup (((X \rightarrow Y) \leftrightarrow Z)^c \cap Z)$$

The novel algorithm that can be used to prove the theorem follows below:

Boolean theorem5.3c(boolean X, Y, Z)

1. Read data
 - 1.1 Read X
 - 1.2 Read Y
 - 1.3 Read Z
2. Determine theorem5.3c
 - 2.1. IF $(X = U) \cap (Y = \phi)$ THEN
 - 2.1.1 first = ϕ
 - ELSE
 - 2.1.2 first = U
 - 2.2 IF first = Z THEN
 - 2.2.1. third = U
 - ELSE
 - 2.2.2 third = ϕ
 - 2.3 fourth = $\text{third} \cap Y$
 - 2.4 fifth = $\text{third}^c \cap Z$
 - 2.5 sixth = $Y \cap Z$
 - 2.6 left = $\text{fourth} \cup \text{fifth} \cup \text{sixth}$
 - 2.7 right = $\text{fourth} \cup \text{fifth}$
 - 2.8 IF left = right THEN
 - 2.8.1 result = “The Theorem Holds”
 - ELSE
 - 2.8.2 result = “The Theorem does not hold”
3. Display result

Apart from using the algorithms to prove or verify Boolean algebra theorems/equations, the algorithms can also be used to evaluate Boolean expressions. Eaxmple, consider the set of logical values as Boolean algebra, suooose X, Y and Z are members of the set, the evaluation of the Boolean expression shown below gives TRUE result for the various values of Boolean variables.

$$(X \rightarrow Y) \wedge (Y \rightarrow Z) \rightarrow (X \rightarrow Z)$$

Such a Boolean expression that gives TRUE result always is called a tautology.

Similarly, Consider the set of binary numbers as Boolean algebra, suppose X, Y and Z are members of the set, the above tautology can be written as:

$$(X \rightarrow Y) \cdot (Y \rightarrow Z) \rightarrow (X \rightarrow Z)$$

The evaluation of this Boolean expression always gives 1 result for different values of the Boolean variables.

Furthermore, consider the set that contains the universal set and empty set as Boolean algebra, suppose X, Y and Z are members of the set, the tautology can be written as:

$$(X \rightarrow Y) \cap (Y \rightarrow Z) \rightarrow (X \rightarrow Z)$$

The evaluation of this Boolean expression always gives U result for different values of the Boolean variables.

Algorithms that are similar to the algorithms used to verify Boolean theorems/equations can be designed to evaluate each of the tautologies, for each variant of Boolean algebra.

Consider the set of logical values as Boolean algebra, suppose X, Y and Z are members of the set, the novel algorithms that will evaluate the tautology follows below:

Boolean tautology1(boolean X, Y, Z)

1. Read data
 - 1.1. Read X
 - 1.2. Read Y
 - 1.3. Read Z
2. Determine tautology1
 - 2.1. IF $(X = \text{TRUE}) \text{ AND } (Y = \text{FALSE})$ THEN
 - 2.1.1. first = FALSE
 - ELSE
 - 2.1.2 first = TRUE
 - 2.2 IF $(Y = \text{TRUE}) \text{ AND } (Z = \text{FALSE})$ THEN
 - 2.2.1 second = FALSE
 - ELSE
 - 2.2.2 second = TRUE
 - 2.3. F $(X = \text{TRUE}) \text{ AND } (Z = \text{FALSE})$ THEN
 - 2.3.1 third = FALSE
 - ELSE
 - 2.3.2 third = TRUE
 - 2.4 fourth = first AND second
 - 2.5 IF fourth = TRUE AND third = FALSE THEN
 - 2.5.1 result = “Not Tautology”
 - ELSE
 - 2.5.2 result = “TRUE”
3. Display Result

In a similar manner, consider the set of binary numbers as Boolean algebra, suppose X, Y and Z are members of the

set, the tautology can be evaluated using the algorithm below:

Boolean tautology2(boolean X, Y, Z)

1. Read data
 - 1.1 Read X
 - 1.2 Read Y
 - 1.3 Read Z
2. Determine tautology2
 - 2.1 IF $(X = 1).(Y = 0)$ THEN
 - 2.1.1. first = 0
 - ELSE
 - 2.1.2 first = 1
 - 2.2 IF $(Y = 1).(Z = 0)$ THEN
 - 2.2.1 second = 0
 - ELSE
 - 2.2.2 second = 1
 - 2.3. F $(X = 1).(Z = 0)$ THEN
 - 2.3.1 third = 0
 - ELSE
 - 2.3.2 third = 1
 - 2.4 fourth = first.second
 - 2.5 IF $(fourth = 1).(third = 0)$ THEN
 - 2.5.1 result = "Not Tautology"
 - ELSE
 - 2.5.2 result = "TRUE"
3. Display Result

Furthermore, consider another variant of Boolean algebra, set, whose elements are the universal set and empty set. Suppose X, Y and Z are members of the set, the algorithm that can be used to evaluate the tautology follows below:

Boolean tautology3(boolean X, Y, Z)

1. Read data
 - 1.1 Read X
 - 1.2 Read Y
 - 1.3 Read Z
2. Determine tautology3
 - 2.1 IF $(X = U) \cap (Y = \phi)$ THEN
 - 2.1.1. first = ϕ
 - ELSE
 - 2.1.2 first = U
 - 2.2 IF $(Y = U) \cap (Z = \phi)$ THEN
 - 2.2.1 second = ϕ
 - ELSE
 - 2.2.2 second = U
 - 2.3. F $(X = U) \cap (Z = \phi)$ THEN
 - 2.3.1 third = ϕ
 - ELSE
 - 2.3.2 third = U

2.4 fourth = first $\cap$ second

2.5 IF $(fourth = U) \cap (third = \phi)$ THEN

2.5.1 result = "Not Tautology"

ELSE

2.5.2 result = U

3. Display Result

6. IMPLEMENTATION OF THE ALGORITHMS

Java programming language can be used to implement all the algorithms that have been designed in this paper. Java provides Boolean data type, together with all the relevant Boolean operators that can be used to implement the algorithms. Each algorithm has been implemented in a Java class, and tested separately.

7. RESULT OF THE IMPLEMENTED ALGORITHM

All the Boolean theorems are correct and the tautology gives TRUE or 1 or U result always, for the appropriate variants of Boolean algebra.

8. CONCLUSION

This paper has been able to identify three variants of Boolean algebra. It has also designed algorithms that can be used to verify the Boolean algebra theorems/equations and evaluate Boolean algebra expressions for each variant of Boolean algebra. The algorithms have been implemented and tested using Java programming language.

REFERENCES

- [1] Delin Tan, "On the Structure of Finite Boolean Algebra", *Journal of Mathematical Sciences & Mathematics Education*, Vol. 5 No. 1.
- [2] Paul W., Dale M., D., "An Introduction to the Application of Boolean Algebra to Psychiatry", *Psychiatry Quarterly*, January 1955, Volume 29, Issue 1-4, pp 48 – 59, Springer.
- [3] Lianly Rompis, "Boolean Algebra for Xor Gates" *Journal of Computer Science and Applications*, 2013, Volume 1 No 1, pp 14 - 16.
- [4] Mamoni Dhar, "Fuzzy Sets towards forming Boolean Algebra", *International Journal of Energy, Information and Communications*, Volume 2 Issue 4, November 2011.
- [5] Azram M., "Arithmetic Version of Boolean Algebra", *Proc. of 2nd IEEE International Conference on Computer Science and Information Technology*, 2009, pp 79 – 80.
- [6] Bernard Kolman, Robert C. Busby, "Discrete Mathematical Structures for Computer Science", Prentice Hall Inc., 1984.

Author's Brief



Dr. Oguike, Osondu Everestus is a Senior Lecturer in the Department of Computer Science, University of Nigeria, Nsukka, Enugu State, Nigeria. He has received many academic prizes and scholarships as a result of his outstanding academic performance. He is interested in modeling the performance of parallel computer system. He can be reached by phone on +2348035405100 and through E-mail osondu.oguike@unn.edu.ng