Boolean algebra

The logical symbol 0 and 1 are used for representing the digital input or output. The symbols "1" and "0" can also be used for a permanently open and closed digital circuit. The digital circuit can be made up of several logic gates. To perform the logical operation with minimum logic gates, a set of rules were invented, known as the **Laws of Boolean Algebra**. These rules are used to reduce the number of logic gates for performing logic operations.

The Boolean algebra is mainly used for simplifying and analyzing the complex Boolean expression. It is also known as **Binary algebra** because we only use binary numbers in this. **George Boole** developed the binary algebra in **1854**.

Rules in Boolean algebra

1. Only two values(1 for high and 0 for low) are possible for the variable used in Boolean algebra.
2. The overbar(-) is used for representing the complement variable. So, the complement of variable C is represented as .
3. The plus(+) operator is used to represent the ORing of the variables.
4. The dot(.) operator is used to represent the ANDing of the variables.

Properties of Boolean algebra

These are the following properties of Boolean algebra:

Annulment Law

When the variable is AND with 0, it will give the result 0, and when the variable is OR with 1, it will give the result 1, i.e.,

B.0 = 0

B+1 = 1

Identity Law

When the variable is AND with 1 and OR with 0, the variable remains the same, i.e.,

B.1 = B

B+0 = B

Idempotent Law

When the variable is AND and OR with itself, the variable remains same or unchanged, i.e.,

B.B = B

B+B = B

Complement Law

When the variable is AND and OR with its complement, it will give the result 0 and 1 respectively.

B.B' = 0

B+B' = 1

Double Negation Law

This law states that, when the variable comes with two negations, the symbol gets removed and the original variable is obtained.

((A)')' = A

Commutative Law

This law states that no matter in which order we use the variables. It means that the order of variables doesn't matter in this law.

A.B = B.A

A+B = B+A

Associative Law

This law states that the operation can be performed in any order when the variables priority is of same as '*' and '/'.

(A.B).C = A.(B.C)

(A+B)+C = A+(B+C)

Distributive Law

This law allows us to open up of brackets. Simply, we can open the brackets in the Boolean expressions.

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

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

Absorption Law

This law allows us for absorbing the similar variables.

$B+(B.A) = B$

$B.(B+A) = B$

De Morgan Law

The operation of an OR and AND logic circuit will remain same if we invert all the inputs, change operators from AND to OR and OR to AND, and invert the output.

$(A.B)' = A'+B'$

$(A+B)' = A'.B'$