

Introduction to Boolean Algebra

These notes form an introduction to Boolean algebra and cover the knowledge required by the BCS Certificate Exam. We explain what Boolean algebra is and how it is used to manipulate simple expressions. A separate article looks at truth tables and gates.

Boolean algebra is the algebra of logic (technically speaking, there are an infinite number of Boolean algebras but computer scientists are normally concerned only with the Boolean algebra of a two-value variable).

Boolean algebra is defined as a mathematical system with a set of elements whose values are either 0 or 1, together with a set of operators (and, or, negation), and a set of *axioms* or rules.

The Boolean operation + is called and OR operation. The '+' symbol is used because the logical OR operator behaves rather like the arithmetic addition operator. Similarly, the Boolean logical AND operator is written as a dot because it behaves like the arithmetic multiplication operation.

In Boolean algebra $(A + B) \cdot (C + D) = A \cdot C + A \cdot D + B \cdot C + B \cdot D$, which is exactly the same as in conventional algebra.

By the way, in virtually all books and articles, an expression like $A \cdot C + A \cdot D + B \cdot (C + B \cdot D)$ would be written as $AC + AD + B(C + BD)$ without the dot operator because (like the multiplication sign) it is assumed to exist between two adjacent terms. However, I have included it in all expressions her to remind students.

An important aspect of Boolean algebra is that it has strong similarities with the conventional algebra that you learn at school; in particular, operations are both *associative* and *commutative*; for example, $A \cdot B \cdot C \cdot D = B \cdot D \cdot C \cdot A$ (commutative law) and $A + B + C + D = D + A + B + C$ (*associative law*). In other words, the order in which you perform the addition (logical OR) of variables is unimportant. The same is true for multiplication/logical AND.

Moreover, in Boolean algebra the dot (AND) operator takes precedence over the plus (OR) operator; for example, $A + B \cdot C$ is not the same as $(A + B) \cdot C$ just as $5 + 2 \times 3$ is not the same as $(5 + 2) \times 3$.

Boolean algebra has two constants 1 and 0 (often called *true* and *false*). A constant does not change its value.

Each element in Boolean algebra has a complement; for example, the complement of X is often written as

\bar{X} .

Note that the complement of X is obtained by negating X.

However, since it is difficult to draw over-bars with many word processors and virtually impossible in HTML, it is common for a complement to be indicated by a symbol after the variable. The symbols !, ', and \neg are commonly used, that is, the complement of X is frequently indicated by $X!$, X' , $X\neg$ in books and articles. Here we will use X' to indicate the complement of X. We can say that:

The complement of 0 is 1; that is, $0' = 1$

The complement of 1 is 0; that is, $1' = 0$

The complement of a complement is the original value; that is, $(X')' = X$. Similarly, $1'' = 1$ and $0'' = 0$.

NOTE:

The terms complement, negate and invert are used interchangeably. We can talk about complementing X, inverting X, or negating X. Part of the reason for this state of affairs is that Boolean algebra was derived by mathematicians but implemented by engineers.

NOTE:

Boolean algebra defines only three operators: AND, OR, NOT. Conversely, there is no subtraction operation and there is no division. In conventional algebra, the expression $A + B = A + C$ would be solved by saying

$$A + B = A + C$$

$$B = A + C - A. \text{ Therefore, } B = C.$$

In Boolean algebra, if we have $A + B = A + C$, we cannot say that $B = C$ because that would require subtraction which is not defined. Similarly, if we say that $A \cdot B = A + D$, we cannot say $B \cdot (A - 1) = D$ and $B = (A - 1)/D$.

Postulates of Boolean Algebra

The fundamental postulates (axioms or rules) of Boolean algebra are:

- | | | |
|-----|---|------------------|
| 1a. | $X + 0 = X$ | Identity |
| 1b. | $X \cdot 1 = X$ | |
| 2a. | $X + X' = 1$ | Complement |
| 2b. | $X \cdot X' = 0$ | |
| 3a. | $X + Y = Y + X$ | Commutative law |
| 3b. | $X \cdot Y = Y \cdot X$ | |
| 4a. | $X + (Y + Z) = (X + Y) + Z$ | Associative law |
| 4b. | $X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z$ | |
| 5a. | $X + (Y \cdot Z) = (X + Y) \cdot (X + Z)$ | Distributive law |
| 5b. | $X \cdot (Y + Z) = (X \cdot Y) + (X \cdot Z)$ | |

Of these laws, only 5a is counter-intuitive to those who have studied high-school algebra.

As well as these postulates, there are several theorems of Boolean algebra that can be derived directly from the postulates. These rules are given below and there are proofs in many textbooks and web articles. Students taking the BCS Certificate course do not need to know these formal proofs. The only requirement is to be able to use the theorems to simplify Boolean expressions. Note that in any of the theorems below, X can be replaced by its complement X' and the theorem is still valid.

- | | |
|------------|--------------------------------|
| Theorem 1 | $X \cdot 0 = 0$ |
| Theorem 2 | $X + 1 = 1$ |
| Theorem 3 | $(X + Y)' = X' \cdot Y'$ |
| Theorem 4 | $(X \cdot Y)' = X' + Y'$ |
| Theorem 5 | $X + X \cdot Y = X$ |
| Theorem 6 | $X \cdot (X + Y) = X$ |
| Theorem 7 | $X + X' \cdot Y = X + Y$ |
| Theorem 8 | $X \cdot (X' + Y) = X \cdot Y$ |
| Theorem 9 | $X \cdot Y + X \cdot Y' = X$ |
| Theorem 10 | $(X + Y) \cdot (X + Y') = X$ |

You use these theorems by looking for patterns in a Boolean expression where variables in the expression you are simplifying can be replaced by X and Y, and one of the above theorems used. Consider the following example.

$$F = A \cdot B \cdot C + A' \cdot B \cdot C + A \cdot B' \cdot C + A \cdot B \cdot C'$$

To deal with this equation, we look for ways of rearranging and factoring the expression. Here, we immediately notice that the first two (leftmost) terms have a common factor $B \cdot C$. Let's take this out and see what happens.

$$F = B \cdot C \cdot (A + A') + A \cdot B' \cdot C + A \cdot B \cdot C'$$

Note the term in parentheses. This is $A + A'$; that is, a value and its complement. From postulate 2a we state that $A + A' = 1$. If we make this substitution we get:

$$F = B \cdot C \cdot (A + A') + A \cdot B' \cdot C + A \cdot B \cdot C' = B \cdot C \cdot (1) + A \cdot B' \cdot C + A \cdot B \cdot C' = B \cdot C + A \cdot B' \cdot C + A \cdot B \cdot C'$$

If we examine this expression, we can find three ways of re-factoring; C in the first two terms, A in the last two terms, and B in the first and third terms. Let's take the C out of the first two terms.

$$F = B \cdot C + A \cdot B' \cdot C + A \cdot B \cdot C' = C \cdot (B + A \cdot B') + A \cdot B \cdot C'$$

Consider the $B + A \cdot B'$ term. Can we simplify it? Yes – we can use theorem 7 to write $B + A \cdot B' = B + A$. Now we can continue with:

$$F = C \cdot (B + A \cdot B') + A \cdot B \cdot C' = C \cdot (B + A) + A \cdot B \cdot C' = B \cdot C + A \cdot C + A \cdot B \cdot C'$$

By the way, you can write the elements of products and sums in any order. However, I try and keep them in alphabetic order to make it easier for me to see factors; for example, I would write $C \cdot B$ as $B \cdot C$. Both these terms are equivalent but sticking to alphabetical order reduces the change of an error when you are dealing with expressions with many terms.

Again, we can find three possible re-factorizations. We could take the C out of the first two terms to get $B \cdot C + A \cdot C = C \cdot (B + A)$. However, we cannot simplify $B + A$, so that is of no help.

We can factorise the first and third or the second and third terms. Let's try taking out the B in the first and last terms:

$$F = B \cdot C + A \cdot C + A \cdot B \cdot C' = B \cdot (C + A \cdot C') + A \cdot C$$

Note that we can now use theorem 7 again to get

$$F = B \cdot (C + A) + A \cdot C = B \cdot C + A \cdot B + A \cdot C$$

At this stage, we cannot find any further simplification and the expression is now in its simplest form.

NOTE 1

There are many ways of writing down Boolean equations. However, all Boolean expressions can be expressed in one of two common formats. These are sum of *products form* and *product of sums form*. These two forms are often abbreviated to SoP and PoS, respectively.

A sum of products expression is in the form of the sum (logical OR) of a series of product (AND) terms. The following is a SoP expression:

$$F = B \cdot C' \cdot D + A \cdot B' + A' \cdot C \cdot D$$

A sum of products expression can include a term that is a single element; for example;

$$F = B \cdot D + A + A' \cdot C \cdot D$$

A product of sums expression is in the form of the product (logical AND) of a series of sum (OR) terms. The following is a PoS expression:

$$F = (A + B + C') \cdot (B + D') \cdot (A + C + D')$$

As in the case of a sum of products, a product of sums can have a single variable as a component; for example:

$$F = (A + B + C') \cdot (B + D') \cdot D'$$

The following expression is in neither a SoP form nor a PoS form. In general, such a mixed format is not normally used in textbooks and articles.

$$F = (A + C') \cdot (B + D') \cdot (A + C + B \cdot D') + B \cdot C' \cdot D + A' \cdot C \cdot D$$

NOTE 2

Computer scientists tend to develop their own ways of simplifying Boolean expressions. I usually multiply out bracketed terms before applying theorems. For example, I rarely use theorem 10, $(X + Y) \cdot (X + Y') = X$, because if I encountered $(A + B) \cdot (A + B')$ I would write $(A + B) \cdot (A + B') = A \cdot A + A \cdot B' + A \cdot B + B \cdot B'$. Since $A \cdot A = A$, we can write $A \cdot A + A \cdot B' + A \cdot B + B \cdot B' = A + A \cdot B' + A \cdot B + B \cdot B'$. Because, $A + A \cdot B = A$, we can write

$$A + A \cdot B' + A \cdot B + B \cdot B' = A + B \cdot B'$$

Finally, $B \cdot B' = 0$, which leaves us with $A + B \cdot B' = A$.

NOTE 3

Negation can be applied to an individual term *or to an entire expression*. Consider the following example:

$$F = B \cdot C' \cdot D + A \cdot B' + (A' \cdot C \cdot D)'$$

Let's write this in overbar form for the sake of clarity.

$$F = B \cdot \overline{C} \cdot D + A \cdot \overline{B} + \overline{A' \cdot C \cdot D}$$

When an entire expression is negated, you must use theorems 3 and 4 to evaluate the result. For example, $(X + Y)'$ is **not** equal to $X' + Y'$. This is probably the most common mistake made by students. The second most common mistake is to say that $A \cdot B' + A' \cdot B = 1$. No it is not! The expression $A \cdot B' + A' \cdot B$ cannot be simplified; indeed, it has a special name and is called the *exclusive OR* of A and B. This is often written A EOR B or A XOR B and it even had its own symbol: $A \oplus B$.

Theorems 3 and 4 are known collectively as deMorgan's theorem which states '*to negate an expression, complement its components and change products into sums and vice versa*'. Consider the following example where we negate an expression with three product terms A, B·C', D·E·F.

$$F = (A + B \cdot C' + D \cdot E \cdot F)'$$

In order to apply deMorgan's theorem we must invert the elements and change the operators. The three elements are A, B·C', and D·E·F and the operators are all +. Therefore, we can write:

$$F = (A)' \cdot (B \cdot C')' \cdot (D \cdot E \cdot F)'$$

Note that we now have three negated elements. One is a simple variable. The other two, (B·C') and (D·E·F) are expressions. So, we have to apply deMorgan's theorem a second time to these elements. This gives:

$$F = A' \cdot (B' + C) \cdot (D' + E' + F')$$

Examples of Boolean Simplification

Example 1: $F = A + (B \cdot A)'$

The second component $(B \cdot A)'$ is negated and we must use deMorgan's theorem $(B \cdot A)' = B' + A'$. We can now write

$$F = A + (B \cdot A)' = A + B' + A'$$

Postulate 2a says that $X + X' = 1$, therefore $F = A + A' + B' = 1 + B'$

Theorem 2 says that $1 + X' = 1$, therefore $F = 1 + B' = 1$.

The final result is $F = 1$ which means that the expression is true and independent of variables A and B.

Note also that I have gone through this example step-by-step slowly. With practice, you would immediately recognize that $X + X'$ in a sum of products expression would be 1 and that adding (ORing) any further terms would not change the result.

Example 2: $F = (A + B) \cdot (A + C)$

$$F = A \cdot A + A \cdot C + A \cdot B + B \cdot C$$

Immediately you can see that we have $A \cdot A$ which is A and that any other product term containing A will be redundant. Consequently, we are left with:

$$F = A + B \cdot C$$

Example 3: $F = A' \cdot B' \cdot C + (A \cdot B + A \cdot C)'$

Here we have a nonstandard format. This is in neither SoP or PoS form because we have a term that is negated. We have to use deMorgan's theorem to remove the negation (complementation) of this term.; that is,

$$(A \cdot B + A \cdot C)' = (A \cdot B)' \cdot (A \cdot C)'$$

We've complemented the elements and changed the OR operator to and AND operator. However, we still have negated expressions. We have to continue.

$$(A \cdot B + A \cdot C)' = (A \cdot B)' \cdot (A \cdot C)' = (A' + B') \cdot (A' + C')$$

Now, negation is applied only to single variables and we can continue:

$$F = A' \cdot B' \cdot C + (A \cdot B + A \cdot C)' = A' \cdot B' \cdot C + (A' + B') \cdot (A' + C')$$

We can immediately simplify the product of the two sums, $(A' + B') \cdot (A' + C')$, to get $A' + B' \cdot C'$

$$F = A' \cdot B' \cdot C + (A' + B') \cdot (A' + C') = A' \cdot B' \cdot C + A' + B' \cdot C'$$

We now have a sum of products expression. However, it is not in its most simple form. We can take the A' out of the first two terms to get:

$$F = A' \cdot B' \cdot C + A' + B' \cdot C' =$$

$$\text{Since } B' \cdot C + 1 = 1, F = A' \cdot (B' \cdot C + 1) + B' \cdot C' = A' \cdot (1) + B' \cdot C' = A' + B' \cdot C'$$

The final value is $F = A' + B' \cdot C'$ which is in its most simple form.