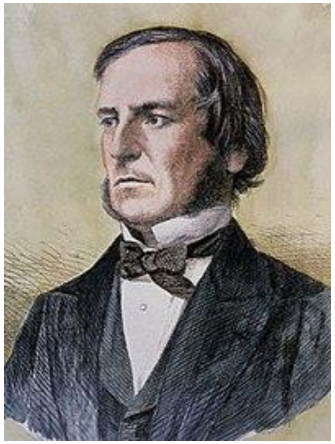
BOOLEAN ALGEBRA

- **Laws of Boolean Algebra**
- **Boolean Algebra Rules**
- **De Morgan Theorems**
- **Truth Table**
- **Venn Diagrams**
- **Simplification of logic expressions**

BOOLEAN ALGEBRA

Boolean algebra was introduced by George Boole in 1854. Boolean algebra is a branch of algebra based on *true* and *false* values. It is first used by Claude Shannon for designing and analyzing logical circuits.



George Bool

We can consider Boolean algebra as a system based on three basic logical operations: AND, OR and NOT.

AND operation corresponds to multiplication in binary, OR operation corresponds to addition, and NOT operation corresponds to complement operation.

Laws of Boolean Algebra

Commutative, associative, distributive and double negation are the laws of Boolean algebra.

Commutative Law

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

Associative Law

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

Distributive Law

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

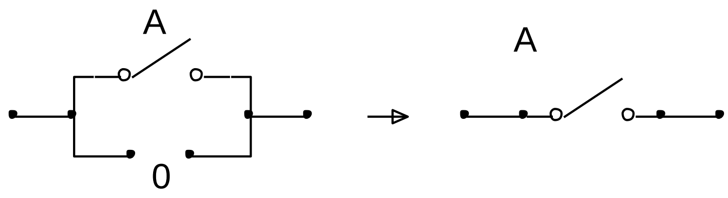
Double Negation Law

- $(A')' = A$

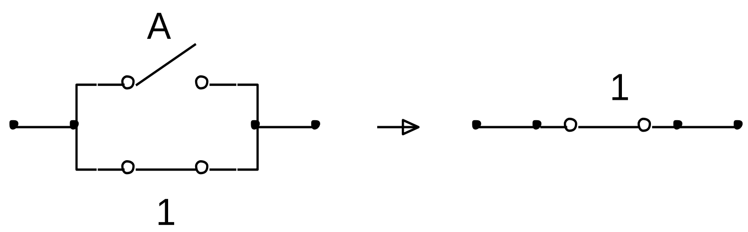
Boolean Algebra Rules

These rules are useful when simplifying logical expressions.

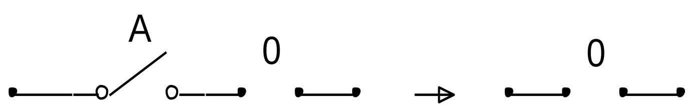
- ***A* + *0* = *A***
  $1 + 0 = 1$
  $0 + 0 = 0$



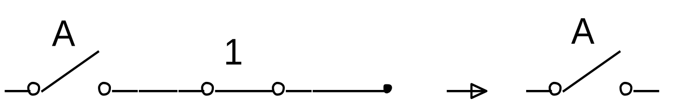
- ***A* + *1* = *1***
  $1 + 1 = 1$
  $0 + 1 = 1$



- ***A* . *0* = *0***
  $1 . 0 = 0$
  $0 . 0 = 0$



- ***A* . *1* = *A***
  $1 . 1 = 1$
  $0 . 1 = 0$

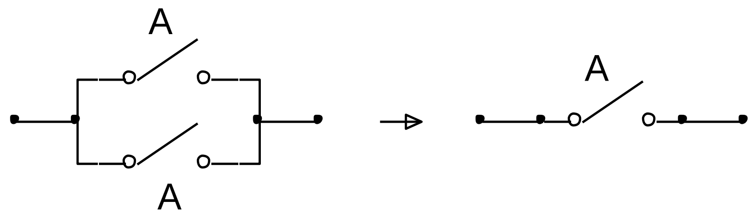


Boolean Algebra Rules

- ***A + A = A***

  1 + 1 = 1

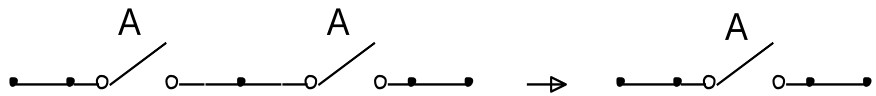
  0 + 0 = 0



- ***A . A = A***

  1 . 1 = 1

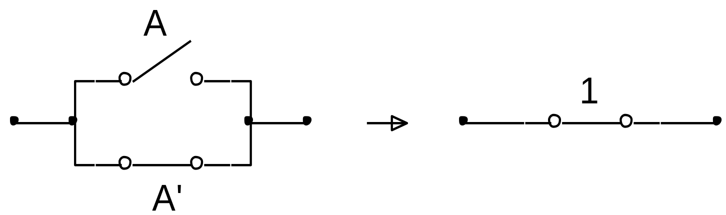
  0 . 0 = 0



- ***A + A' = 1***

  1 + 0 = 1

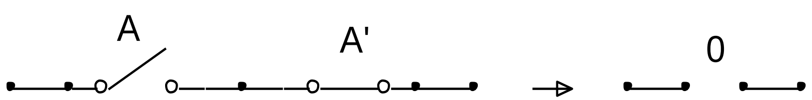
  0 + 1 =1



- ***A . A'= 0***

  1 . 0 = 0

  0 . 1 = 0



- ***(A')' = A***

Boolean Algebra Rules

The following rules are frequently used to simplify logical expressions.

- $\boldsymbol{A + A.B = A}$

  $A.(1+B) = A$

- $\boldsymbol{A+ B.C = (A+B).(A+C)}$

  $(A+B).(A+C)=A.A+A.C+A.B+B.C = \underline{A+A.C}+A.B+B.C$

  $= \underline{A+A.B}+B.C=A+B.C$

- $\boldsymbol{A + A'.B = A+ B}$

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

De Morgan Theorems

1st Theorem

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

2nd Theorem

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

Augustus De Morgan

De Morgan theorems can also be applied to logical expressions with more than two variables.

- $(A.B.C)' = A' + B' + C'$
- $(A+B+C)' = A'.\ B'.\ C'$

Let's prove that $(A+B+C)' = A'.B'.C'$

$[A + (B+C)]' = A'.(B+C)' = A'.B'.C'$

Truth Table

Truth table shows all the outputs that corresponds to all possible values that variables can have. If there are *n* variables in the expression, then the number of combinations is 2^n.

There's a practical method to generate all combinations. The column of LSB is filled with a 0 and a 1 repeatedly. Then the next column is filled with two 0's and two 1's repeatedly. The next column is filled with four 0's and four 1's repeatedly. And the rest of the columns are filled the same way. The number of 0's and 1's increase by the power of 2. (2^0, 2^1, 2^2, 2^3,…)

A	*B*	*Output*
0	0	
0	1	
1	0	
1	1	

A	*B*	*C*	*Output*
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Truth Table

Example: Let's generate the truth table of $F(A,B,C) = A+B'C$

We can find out all the outputs by writing all the possible values of variables in the equation for all combinations.

The second way to find out all outputs is by interpreting the expression.

We can say that, the output (*F*) will be 1 if A=1 or B'C=1. For *B'C* to be 1, *B* must be 0 and *C* must be 1.

A	***B***	***C***	***F***	
0	0	0	0	
0	0	1	**1**	→ *B*=0 and *C*=1
0	1	0	0	
0	1	1	0	
1	0	0	**1**	*A*=1
1	0	1	**1**	
1	1	0	**1**	
1	1	1	**1**	

Truth Table

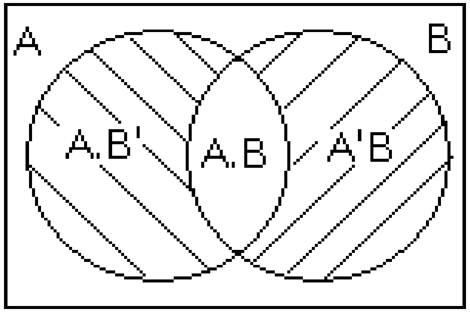
Example: Let's prove De Morgan's theorem (A+*B)'* = *A'* **.** *B'* using truth tables.

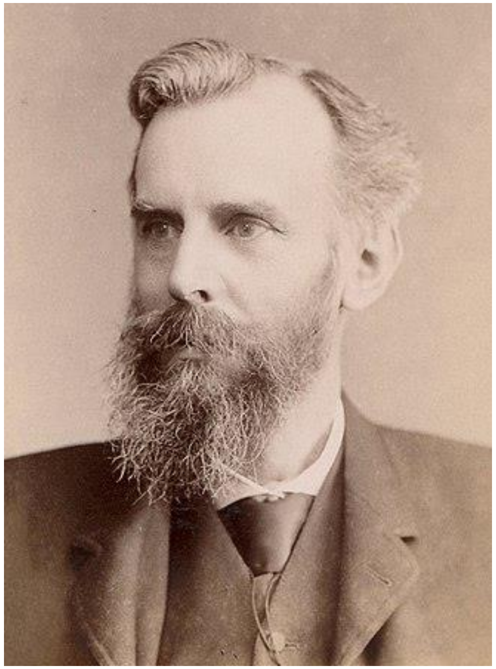
If the output columns are identical in truth tables of these two expressions (the one on the left and the one on the right), we can say that these expressions are identical.

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

Venn Diagrams

We can use Venn diagrams to visualize logical expressions. Each set represents a variable. Inner part of a set represents the variable itself, and the outer part of a set represents the complement of the variable.



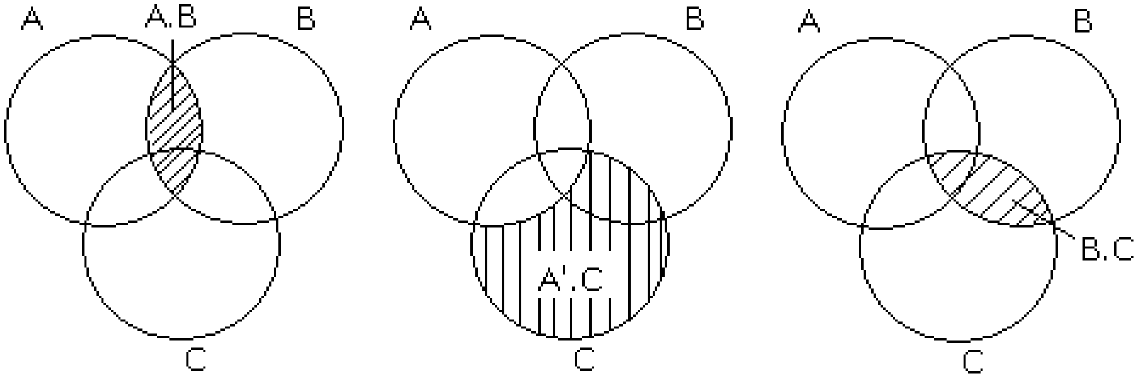


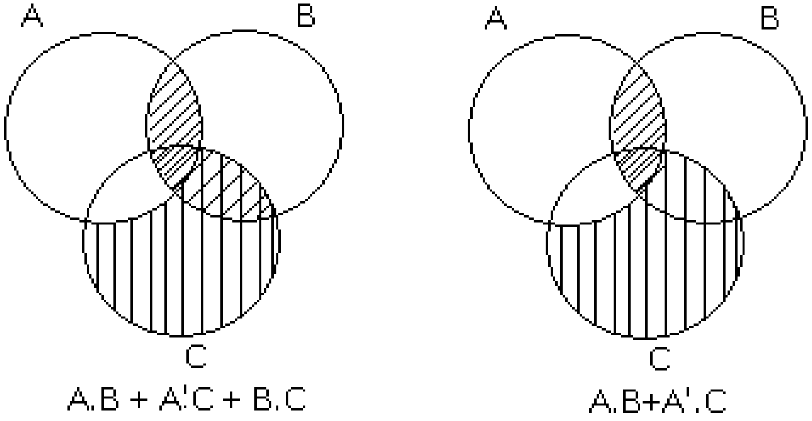
John Venn

We can prove equations with Venn diagrams.

Venn Diagrams

Example: Let's prove that $F(A,B,C) = A.B + A'.C + B.C = A.B + A'.C$ with Venn diagrams.





Simplification of Logical Expressions

Complicated logical expressions can be simplified using Bool algebra rules. Simplifying logical expression helps us build simpler and cost effective logical circuits. We can simplify expressions by collecting like terms or extending the expression with a term which does not affect the output (e.g. A+A').

Example: Let's simplify $F(A,B,C)= ABC'+\underline{A'B'C}+A'BC+\underline{A'B'C'}$

First, we collect like terms. We pull out the common factor (A'B') from 2nd and 4th terms.

$F(A,B,C) = ABC'+A'B'(\underline{C+C'})+A'BC$ — *A+A'=1 rule applied*

$= ABC'+ \underline{A}'B'+\underline{A}'BC$

$= ABC'+A'(\underline{B'+BC})$ — *A+A'B = A+B rule applied*

$= ABC'+A'(B'+C)$

$= ABC'+A'B'+A'C$

Simplification of Logical Expressions

Example: Let's simplify $F(A,B,C) = A.B + A'.C + B.C$

We can extend the term B.C with (A+A').

$$F(A,B,C) = A.B + A'.C + B.C.(A+A') = \underline{A.B} + A'.C + \underline{A.B.C} + A'.B.C$$

$$= A.B.(1+C) + A'.C.(1+B) = A.B + A'.C$$

Example: Let's simplify $F(A,B,C) = AB' + A(B+C)' + B(B+C)'$

We apply De Morgan theorems to the 2nd and 3rd terms.

$F(A,B,C) = AB'+A(B'C')+\underline{B(B}'C')$ — $B.B'= 0.$

$= \underline{AB}'+\underline{AB}'C'$ — Pull out AB'

$= AB'(1+C')$ — $1+C'=1$

$= AB'$