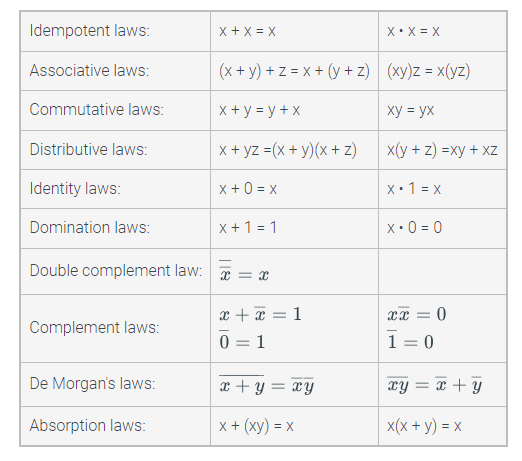
| Boolean Sign 1 = True 0 = False | Logical Sign T = True F = False |
| --- | --- |
| Multiplication \* | AND ( ^ ) |
| Addition + | OR ( V ) |
| Complement (bar) | NOT ( ~ ) |
| Circuit **Closed** = **1** => Electric **can** flow through  Circuit **Open** = **0**  => Electric **can’t** flow through | |

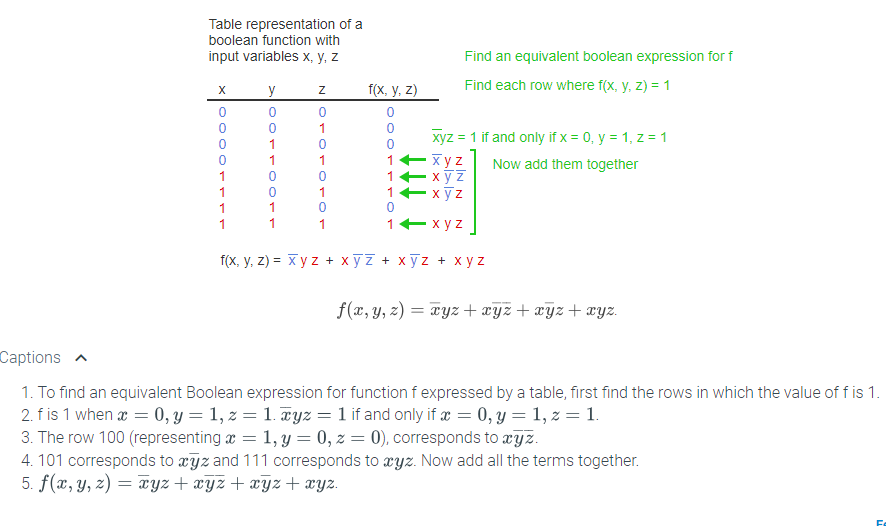


**Minterms** are included for the rows in which the function evaluates to 1. New row is (+) - Or

**Find expressions of Minterm** , if **x** = 1 then **x Bar**  = 0.

**Maxterm** is similar to Minterms but the negation ( bar ) of the variable will be flipped.

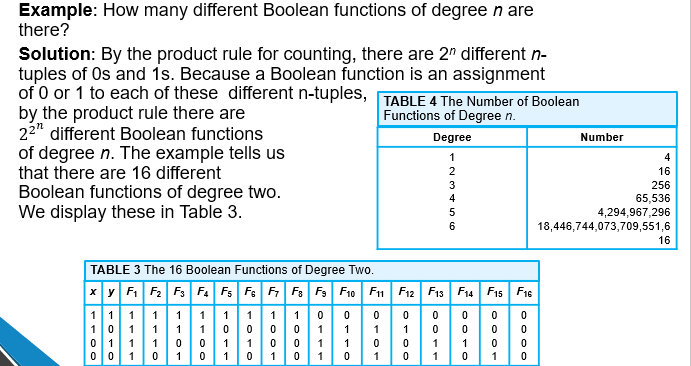
**Find expressions of Minterm** , if **x** = 0 then **x Bar** = 1.



**Conj = (A\*B) Disj = (A+B)**

**DNF Form: Conj-Disj-Conj-...**

**CJF Form: Dish-Conj-Disj-...**



* Degree is how many variables we have. For example **x,y** are degrees of 2.

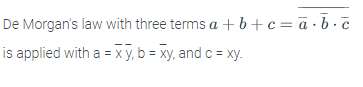
**2^2^n** => 2^2^2 equals 16. That mean we’re having 16 combination of Zeros (0) and Ones (1) from **x,y**

**The NAND Operator mean Not And ( Opposite with And )**

**Symbol: Arrow Up**

**The NOR Operator mean Not OR ( Opposite with Or )**

**Symbol: Arrow Down**

**Elimination of addition (+) in DNF**

**Another Morgan Law**

**The NAND operation (which stands for "not and") is denoted by the symbol ↑. The expression x ↑ y is equivalent to xy. The NOR operation (which stands for "not or") is denoted by the symbol ↓. The expression x ↓ y is equivalent to x + y.**

**Some others:**





**Expression satisfied is when its final produce equal to 1**