School of Engineering and Computer Science

Mathematics Applied to Digital Engineering-2

(Boolean Algebra)

Kamel ATTAR

kamel.attar@epita.fr

Week **#1 #2** ✦ 8-15/February/2024 ✦

Definition (In English)

A **boolean statement** is a declarative sentence (that is, a sentence that declares a fact) that is either **true** or **false**, but not both.

Definition (In English)

A **boolean statement** is a declarative sentence (that is, a sentence that declares a fact) that is either **true** or **false**, but not both.

Example

1. "It is raining today" is a good example.
   - If it is raining, the statement is true.
   - If it is not raining, the statement is false.
2. Paris is the capital of France. (is a true statement)
3. The number **4** is positive and the number **3** is negative. (is a false boolean statement)

Definition (In English)

A **boolean statement** is a declarative sentence (that is, a sentence that declares a fact) that is either **true** or **false**, but not both.

Example

1. "It is raining today" is a good example.
   - If it is raining, the statement is true.
   - If it is not raining, the statement is false.
2. Paris is the capital of France. (is a true statement)
3. The number **4** is positive and the number **3** is negative. (is a false boolean statement)

- Computers really like boolean statements. If we recall, computers evaluate binary. Binary has only two values, **0** and **1**.
- Booleans also only have two values, **false** and **true**.
- This means we can really think of two states, **On** and **Off**, **True** and **False**, **1** and **0**. This means **0**, **false** and **off** all correspond to the same state while **1**, **true** and **on** all correspond to the same state.

Example

The following sentences are not propositions:

1. Your place or mine?
2. What's your name?
3. Knock before entering!
4. $x - y = y - x$.

The last Sentence is not a boolean statement because it is neither **true** nor **false**. Note that the last sentences can be turned into a boolean statement if we specify the symbols.

$$x - y = y - x \text{ for all } x, y \in \mathbb{R},$$

then this is a false proposition. If the intention is

$$x - y = y - x \text{ for all } x, y \in \{0\},$$

then this is a true proposition.

Definition (In Math)

Boolean algebra can be defined as a type of algebra that performs logical operations on binary variables. These variables give the truth values that can be represented either by **0** or **1**.

$\neg$	for "not"	or **Negation**
$\wedge$	for "and"	or **Conjunction**
$\vee$	for "or"	or **Disjunction**

Definition (In Math)

Boolean algebra can be defined as a type of algebra that performs logical operations on binary variables. These variables give the truth values that can be represented either by **0** or **1**.

$\neg$	for "not"	or **Negation**
$\wedge$	for "and"	or **Conjunction**
$\vee$	for "or"	or **Disjunction**

- We use letters to denote **statement variables**, $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}, \cdots$.
- The truth value of a statement is true, denoted by $\boldsymbol{T}$ or **1**, if it is a true statement, and the truth value of a statement is false, denoted by $\boldsymbol{F}$ or **0**, if it is a false statement.

Definition (Truth Tables)

A truth table represent the relationship between the truth values of statements and compound statements formed from those statements.

Definition (Truth Tables)

A truth table represent the relationship between the truth values of statements and compound statements formed from those statements.

- A truth table has one column for each input variable, and one final column showing all of the possible results of the logical operation that the table represents.
- Each row of the truth table contains one possible configuration of the input variables, and the result of the operation for those values.
- A truth table has the role of showing the correspondence between the output and all the combinations of values that the input(s) can take.

Truth Table of $\neg A$		Truth Table of $A \wedge B$			Truth Table of $A \vee B$		
A	$\neg A$	A	B	$A \wedge B$	A	B	$A \vee B$
T	F	T	T	T	T	T	T
F	T	T	F	F	T	F	T
		F	T	F	F	T	T
		F	F	F	F	F	F

Example

Truth Table of $(A \vee B) \Rightarrow (A \wedge B)$				
A	B	$A \vee B$	$A \wedge B$	$(A \vee B) \Rightarrow (A \wedge B)$

Truth Table of $(A \wedge B) \vee \neg(A \Rightarrow B)$					
A	B	$A \wedge B$	$A \Rightarrow B$	$\neg(A \Rightarrow B)$	$(A \wedge B) \vee \neg(A \Rightarrow B)$

Example

Truth Table of $(A \vee B) \Rightarrow (A \wedge B)$				
A	B	$A \vee B$	$A \wedge B$	$(A \vee B) \Rightarrow (A \wedge B)$
T	T	T	T	T
T	F	T	F	F
F	T	T	F	F
F	F	F	F	F

Truth Table of $(A \wedge B) \vee \neg(A \Rightarrow B)$					
A	B	$A \wedge B$	$A \Rightarrow B$	$\neg(A \Rightarrow B)$	$(A \wedge B) \vee \neg(A \Rightarrow B)$

Example

Truth Table of $(A \vee B) \Rightarrow (A \wedge B)$				
A	B	$A \vee B$	$A \wedge B$	$(A \vee B) \Rightarrow (A \wedge B)$
T	T	T	T	T
T	F	T	F	F
F	T	T	F	F
F	F	F	F	F

Truth Table of $(A \wedge B) \vee \neg(A \Rightarrow B)$					
A	B	$A \wedge B$	$A \Rightarrow B$	$\neg(A \Rightarrow B)$	$(A \wedge B) \vee \neg(A \Rightarrow B)$
T	T	T	T	F	T
T	F	F	F	T	T
F	T	F	T	F	F
F	F	F	T	F	F

★ Boolean Algebra ★

Basic logic functions

- The easiest way to understand logic functions is to represent them with electrical diagrams that include one or more "**contacts**" and an "**LED**".

★ Boolean Algebra ★

Basic logic functions

- The easiest way to understand logic functions is to represent them with electrical diagrams that include one or more "**contacts**" and an "**LED**".
- A **contact** has only two states: *open* or *closed*.
  - An open contact breaks the current flow
  - whereas a closed contact allows current to flow through it to the next element.

  The simplest contact is an *On/OFF switch*, which requires external force (the human hand) to activate it.

★ Boolean Algebra ★

Basic logic functions

- The easiest way to understand logic functions is to represent them with electrical diagrams that include one or more "**contacts**" and an "**LED**".
- A **contact** has only two states: *open* or *closed*.
  - An open contact breaks the current flow
  - whereas a closed contact allows current to flow through it to the next element.

  The simplest contact is an *On/OFF switch*, which requires external force (the human hand) to activate it.
- This **LED** lights up "provided" that the electrical connections allow current to pass through it.

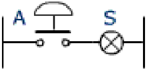
★ Boolean Algebra ★

Basic logic functions

- The easiest way to understand logic functions is to represent them with electrical diagrams that include one or more "**contacts**" and an "**LED**".
- A **contact** has only two states: *open* or *closed*.
  - An open contact breaks the current flow
  - whereas a closed contact allows current to flow through it to the next element.

  The simplest contact is an *On/OFF switch*, which requires external force (the human hand) to activate it.
- This **LED** lights up "provided" that the electrical connections allow current to pass through it.

The following diagram illustrates the simplest condition:
The **LED** lights up if push button $\boldsymbol{A}$ is pressed. in other words ($\boldsymbol{S} = \mathbf{1}$) if ($\boldsymbol{A} = \mathbf{1}$)



- The operation of this circuit is expressed by the logic equation $\boxed{\boldsymbol{S} = \boldsymbol{A}}$
- There are only two possible cases. They are represented in this truth table

$\boldsymbol{A}$	$\boldsymbol{S} = \boldsymbol{A}$
1	**1**
0	**0**

Definition (Conjunction)

The conjunction of A and B, denoted by $A \wedge B$ or $A \cdot B$, is the compound statement "A and B". The conjunction $A \wedge B$ is true when both A and B are true and is false otherwise.

Definition (Conjunction)

The conjunction of $\boldsymbol{A}$ and $\boldsymbol{B}$, denoted by $\boldsymbol{A} \wedge \boldsymbol{B}$ or $\boldsymbol{A} \cdot \boldsymbol{B}$, is the compound statement "$\boldsymbol{A}$ and $\boldsymbol{B}$". The conjunction $\boldsymbol{A} \wedge \boldsymbol{B}$ is true when both $\boldsymbol{A}$ and $\boldsymbol{B}$ are true and is false otherwise.

Example

"The number **4** is positive and the number **3** is negative". This can be viewed as the compound proposition $\boldsymbol{A} \wedge \boldsymbol{B}$ [read "$\boldsymbol{A}$ and $\boldsymbol{A}$"], where $\boldsymbol{A} =$ "**4** is positive" and $\boldsymbol{B} =$ "**3** is negative".

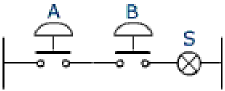
Definition (Conjunction)

The conjunction of A and B, denoted by $A \wedge B$ or $A \cdot B$, is the compound statement "A and B". The conjunction $A \wedge B$ is true when both A and B are true and is false otherwise.

Example

"The number **4** is positive and the number **3** is negative". This can be viewed as the compound proposition $A \wedge B$ [read "A and A"], where A = "**4** is positive" and B = "**3** is negative".

The **LED** lights up if contacts A and B are activated simultaneously



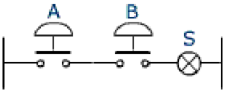
Definition (Conjunction)

The conjunction of A and B, denoted by $A \wedge B$ or $A \cdot B$, is the compound statement "A and B". The conjunction $A \wedge B$ is true when both A and B are true and is false otherwise.

Example

"The number **4** is positive and the number **3** is negative". This can be viewed as the compound proposition $A \wedge B$ [read "A and A"], where A = "4 is positive" and B = "3 is negative".

The **LED** lights up if contacts A and B are activated simultaneously



- Here the four possible combinations of truth values for A and B are listed to the left of the line, and the corresponding truth values of $A \wedge B$ are shown to the right.
- The **AND** operator is represented in the logic equation by a dot. This sign is perfect since the AND function gives the same result as a multiplication.

Truth Table of $A \wedge B$		
A	B	$S = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

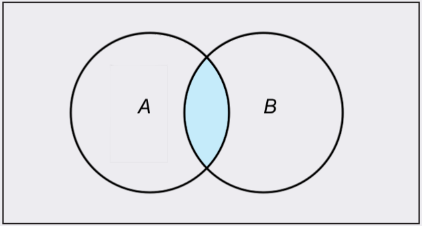
In set theory, the equivalent to **AND** is the intersection of two sets.

Definition (Intersection of Sets)

Let A and B be sets. The intersection of the sets A and B, denoted by $A \cap B$, is the set that contains those elements in both A and B. The venn diagram below illustrates it better.

$$\boxed{A \cap B = \{x : x \in A \quad \text{and} \quad x \in B\}}$$

$$(\forall x)(x \in A \cap B \iff x \in A \quad \text{and} \quad x \in B)$$



$A \cap B$ is shaded

✳ A and B are said to be **disjoint** if $A \cap B = \varnothing$.

The part highlighted in blue is the intersection of sets A and B. It is literally where the two venn diagrams intersect and nothing else. The intersection $\cap$ and AND $\wedge$ symbols are very similar.

Theorem

Let A, B, C and D be sets. Then

1. $A \cap B \subseteq A$ and $A \cap B \subseteq B$
2. $x \notin A \cap B \iff x \notin A \quad or \quad x \notin B$
3. $A \cap B = B \cap A$
4. $A \cap (B \cap C) = (A \cap B) \cap C$
5. $A \cap A = A$
6. If $A \subseteq B$ and $C \subseteq D$, then $A \cap C \subseteq B \cap D$
7. $A \cap \varnothing = \varnothing \cap A = \varnothing$
8. $A \subseteq B \iff A \cap B = A$

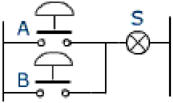
Definition (Disjunction)

The dis-junction of $\boldsymbol{A}$ and $\boldsymbol{B}$, denoted by $\boldsymbol{A} \vee \boldsymbol{B}$ or $\boldsymbol{A} \dotplus \boldsymbol{B}$, is the proposition "$\boldsymbol{A}$ or $\boldsymbol{B}$" The dis-junction $\boldsymbol{A} \vee \boldsymbol{B}$ is false when both $\boldsymbol{A}$ and $\boldsymbol{B}$ are false and is true otherwise.

Definition (Disjunction)

The dis-junction of $\boldsymbol{A}$ and $\boldsymbol{B}$, denoted by $\boldsymbol{A} \vee \boldsymbol{B}$ or $\boldsymbol{A} \dotplus \boldsymbol{B}$, is the proposition "$\boldsymbol{A}$ or $\boldsymbol{B}$" The dis-junction $\boldsymbol{A} \vee \boldsymbol{B}$ is false when both $\boldsymbol{A}$ and $\boldsymbol{B}$ are false and is true otherwise.

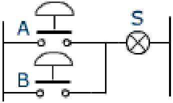
The **LED** lights up if contacts $\boldsymbol{A}$ or $\boldsymbol{B}$ are activated.



Definition (Disjunction)

The dis-junction of $\boldsymbol{A}$ and $\boldsymbol{B}$, denoted by $\boldsymbol{A} \vee \boldsymbol{B}$ or $\boldsymbol{A} \dot{+} \boldsymbol{B}$, is the proposition "$\boldsymbol{A}$ or $\boldsymbol{B}$" The dis-junction $\boldsymbol{A} \vee \boldsymbol{B}$ is false when both $\boldsymbol{A}$ and $\boldsymbol{B}$ are false and is true otherwise.

The **LED** lights up if contacts $\boldsymbol{A}$ or $\boldsymbol{B}$ are activated.



- The **OR** operator is represented in the logic equation by $\dot{+}$. If we look at the last three rows of the truth table, the result of the **OR** operation is similar to the result of an addition.
- The result of $\mathbf{1} \vee \mathbf{1}$, however, differs from $\mathbf{1} + \mathbf{1}$. We use $\dot{+}$ sign to indicate that the operation is not analogous to an addition.

Truth Table of $\boldsymbol{A} \vee \boldsymbol{B}$		
$\boldsymbol{A}$	$\boldsymbol{B}$	$\boldsymbol{S} = \boldsymbol{A} \dot{+} \boldsymbol{B}$
0	**0**	**0**
0	**1**	**1**
1	**0**	**1**
1	**1**	**1**

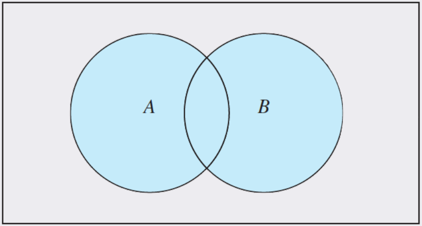
In regular algebra, **OR** is comparable to the union operator $\cup$.

Definition (Union of Sets)

Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be sets. The union of the sets $\boldsymbol{A}$ and $\boldsymbol{B}$, denoted by $\boldsymbol{A} \cup \boldsymbol{B}$ (read: "$\boldsymbol{A}$ union $\boldsymbol{B}$"), is the set that contains those elements that are either in $\boldsymbol{A}$ or in $\boldsymbol{B}$, or in both.

$$\boxed{\boldsymbol{A} \cup \boldsymbol{B} = \{\boldsymbol{x} : \boldsymbol{x} \in \boldsymbol{A} \quad \text{or} \quad \boldsymbol{x} \in \boldsymbol{B}\}}$$

$$(\forall \boldsymbol{x})(\boldsymbol{x} \in \boldsymbol{A} \cup \boldsymbol{B} \Longleftrightarrow \boldsymbol{x} \in \boldsymbol{A} \quad \text{or} \quad \boldsymbol{x} \in \boldsymbol{B})$$



$A \cup B$ is shaded.

The highlighted in blue is what would be in the union of $\boldsymbol{A}$ and $\boldsymbol{B}$. Once again, the union is everything in $\boldsymbol{A}$ added to everything in $\boldsymbol{B}$. The symbol for union ($\cup$) also looks very similar to that of OR ($\vee$).

Theorem

Let A, B, C and D be sets. Then we have

1. $A \subseteq A \cup B$ and $B \subseteq A \cup B$
2. $x \notin A \cup B \iff x \notin A \quad \text{and} \quad x \notin B$
3. $A \cup B = B \cup A$.
4. $A \cup (B \cup C) = (A \cup B) \cup C$.
5. $A \cup A = A$.
6. If $A \subseteq B$ and $C \subseteq D$, then $A \cup C \subseteq B \cup D$.
7. $A \cup \varnothing = \varnothing \cup A = A$.
8. $A \subseteq B \iff A \cup B = B$.

Definition (Negation)

The negation of $\boldsymbol{A}$, denoted by $\neg\boldsymbol{A}$ (also denoted by non$\boldsymbol{A}$ or $\overline{\boldsymbol{A}}$), is the statement **It is not the case that $\boldsymbol{A}$.**

Definition (Negation)

The negation of $\boldsymbol{A}$, denoted by $\neg\boldsymbol{A}$ (also denoted by non$\boldsymbol{A}$ or $\overline{\boldsymbol{A}}$), is the statement **It is not the case that *A*.**

Example

- The negation of the proposition **The Eiffel tower is in Budapest** is **The Eiffel tower is not in Budapest**.
- If $\boldsymbol{A : 4 < 5}$ then $\neg\boldsymbol{A : 4 \geq 5}$.

Definition (Negation)

The negation of A, denoted by $\neg A$ (also denoted by non$A$ or $\overline{A}$), is the statement **It is not the case that A.**

Example

- The negation of the proposition **The Eiffel tower is in Budapest** is **The Eiffel tower is not in Budapest**.
- If $A : 4 < 5$ then $\neg A : 4 \geq 5$.

- The contacts we have used so far are "normally open" contacts. When the push button is released (when $A = 0$) the current does not flow.

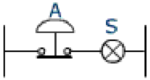
Definition (Negation)

The negation of $\boldsymbol{A}$, denoted by $\neg \boldsymbol{A}$ (also denoted by non$\boldsymbol{A}$ or $\overline{\boldsymbol{A}}$), is the statement **It is not the case that *A*.**

Example

- The negation of the proposition **The Eiffel tower is in Budapest** is **The Eiffel tower is not in Budapest**.
- If $\boldsymbol{A : 4 < 5}$ then $\neg \boldsymbol{A : 4 \geq 5}$.

- The contacts we have used so far are "normally open" contacts. When the push button is released (when $\boldsymbol{A}$ **= 0**) the current does not flow.
- We now use a "normally closed" contact to illustrate the NO function. The current passes but it cuts when the contact is activated (when $\boldsymbol{A}$ **= 1**).



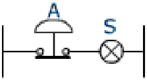
Definition (Negation)

The negation of $\boldsymbol{A}$, denoted by $\neg \boldsymbol{A}$ (also denoted by non$\boldsymbol{A}$ or $\overline{\boldsymbol{A}}$), is the statement **It is not the case that $\boldsymbol{A}$.**

Example

- The negation of the proposition **The Eiffel tower is in Budapest** is **The Eiffel tower is not in Budapest**.
- If $\boldsymbol{A : 4 < 5}$ then $\neg \boldsymbol{A : 4 \geq 5}$.

- The contacts we have used so far are "normally open" contacts. When the push button is released (when $\boldsymbol{A = 0}$) the current does not flow.
- We now use a "normally closed" contact to illustrate the NO function. The current passes but it cuts when the contact is activated (when $\boldsymbol{A = 1}$).



- The truth value of $\neg \boldsymbol{A}$, is the opposite of the truth value of $\boldsymbol{A}$.
- The column to the left of the vertical line lists the possible truth values of $\boldsymbol{A}$. To the right of the line are the corresponding truth values for $\overline{\boldsymbol{A}}$.

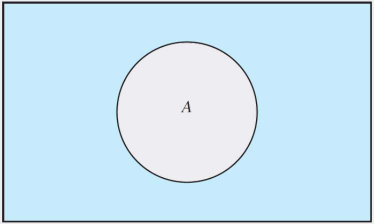
Truth Table of $\neg \boldsymbol{A}$

$\boldsymbol{A}$	$\boldsymbol{S = \overline{A}}$
0	**1**
1	**0**

Definition (The complement of a set)

Let $\boldsymbol{U}$ be the universal set. The complement of the set $\boldsymbol{A}$ is denoted by $\overline{\boldsymbol{A}}$, is the complement of $\boldsymbol{A}$ with respect to the universe.

$$\boxed{\overline{A} = \{x : x \in U \quad \text{and} \quad x \notin A\}}$$



$\bar{A}$ is shaded.

Example

Let $A = \{a, e, i, o, u\}$ (where the universal set is the set of letters of the English alphabet). Then $\overline{A} = \{b, c, d, f, g, h, j, k, l, m, n, p, q, r, s, t, v, w, x, y, z\}$.

We have now introduced three important logical connectives: conjunctions, dis-junctions and negations.

We have now introduced three important logical connectives:
conjunctions, dis-junctions and negations.

- We can use these connectives to build up complicated combinations of logic functions involving .

Example

$(\boldsymbol{A} \cdot \boldsymbol{B}) \dot{+} \neg \boldsymbol{C}$; $\boldsymbol{A} \cdot (\boldsymbol{B} \dot{+} \boldsymbol{C})$

We have now introduced three important logical connectives: conjunctions, dis-junctions and negations.

- We can use these connectives to build up complicated combinations of logic functions involving .

Example

$(A \cdot B) \dotplus \neg C$; $A \cdot (B \dotplus C)$

- We can use truth tables to determine the truth values of these combinations.

We have now introduced three important logical connectives: conjunctions, dis-junctions and negations.

- We can use these connectives to build up complicated combinations of logic functions involving .

Example

$(A \cdot B) \dotplus \neg C$; $A \cdot (B \dotplus C)$

- We can use truth tables to determine the truth values of these combinations.
- Each electrical diagram corresponds to an equation.

We have now introduced three important logical connectives: conjunctions, dis-junctions and negations.

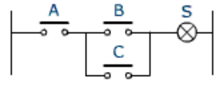
- We can use these connectives to build up complicated combinations of logic functions involving .

Example

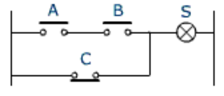
$(A \cdot B) \dot{+} \neg C$; $A \cdot (B \dot{+} C)$

- We can use truth tables to determine the truth values of these combinations.
- Each electrical diagram corresponds to an equation.
- The correspondence between a diagram and a logical function is systematic:
  - Parallel contacts correspond to the ***OR*** function
  - Serial contacts correspond to the ***AND*** function
  - A normally closed contact represents the ***NO*** function

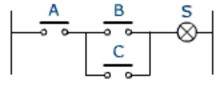
Example



Example

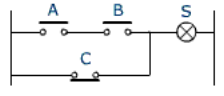


Example

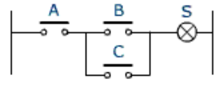


$$S = A \cdot (B + C)$$

Example



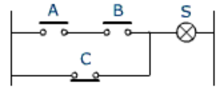
Example



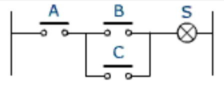
$$S = A \cdot (B \dotplus C)$$

Truth Table of $A \cdot (B \dotplus C)$				
A	B	C	$B \dotplus C$	S
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

Example



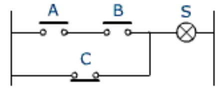
Example



$$S = A \cdot (B \dotplus C)$$

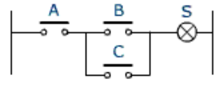
Truth Table of $A \cdot (B \dotplus C)$				
A	B	C	$B \dotplus C$	S
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

Example



$$S = (A \cdot B) \dotplus \neg C$$

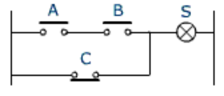
Example



$$S = A \cdot (B \dotplus C)$$

Truth Table of $A \cdot (B \dotplus C)$				
A	B	C	$B \dotplus C$	S
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

Example



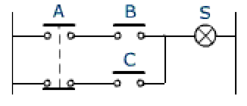
$$S = (A \cdot B) \dotplus \neg C$$

Truth Table of $(A \cdot B) \dotplus \neg C$					
A	B	C	$\neg C$	$A \cdot B$	S
0	0	0	1	0	1
0	0	1	0	0	0
0	1	0	1	0	1
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	0	0	0
1	1	0	1	1	1
1	1	1	0	1	1

Example

We consider the following electric circuit

a. Find the logic equation that corresponds to this circuit.
b. Compute it's truth table.



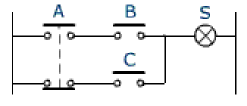
Solution

Example

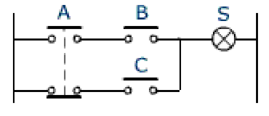
We consider the following electric circuit

a. Find the logic equation that corresponds to this circuit.

b. Compute it's truth table.



Solution



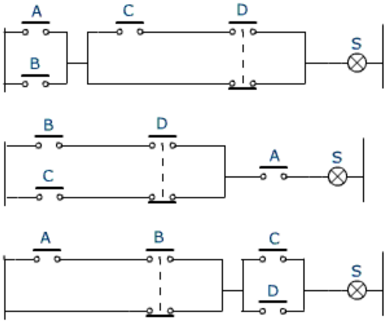
$$\boldsymbol{S = (A \cdot B) \dotplus (\overline{A} \cdot C)}$$

Truth Table of $(A \cdot B) \dotplus (\overline{A} \cdot C)$

N	A	B	C	$\overline{A}$	$\overline{A} \cdot C$	$A \cdot B$	S
0	0	0	0	1	0	0	0
1	0	0	1	1	1	0	1
2	0	1	0	1	0	0	0
3	0	1	1	1	1	0	1
4	1	0	0	0	0	0	0
5	1	0	1	0	0	0	0
6	1	1	0	0	0	1	1
7	1	1	1	0	0	1	1

✎ **Exercise 1.** Compute the truth table of $(A \dot{+} B) \cdot \overline{(A \cdot B)}$

✎ **Exercise 2.** Find the logic equation that corresponds to the following circuits.



✎ **Exercise 3.**

a. Establish a truth table for the Boolean functions $X = (\overline{A} \dot{+} B) \cdot (\overline{C} \dot{+} B)$ and $Y = A \cdot (\overline{B} \dot{+} C)$

b. Design an electric circuit to model X and Y in **a.**

N	AND form	OR form	Identity Name
❶	$A \cdot 1 = A$	$A \dotplus 0 = A$	*Identity laws*
❷	$A \cdot 0 = 0$	$A \dotplus 1 = 1$	*Domination laws*
❸	$A \cdot A = A$	$A \dotplus A = A$	*Idempotent laws*
❹	$\overline{(\overline{A})} = A$		*Double negation laws*
❺	$A \cdot B = B \cdot A$	$A \dotplus B = B \dotplus A$	*Commutative laws*
❻	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$	$(A \dotplus B) \dotplus C = A \dotplus (B \dotplus C)$	*Associative laws*
❼	$A \dotplus (B \cdot C) = (A \dotplus B) \cdot (A \dotplus C)$	$A \cdot (B \dotplus C) = (A \cdot B) \dotplus (A \cdot C)$	*Distributive laws*
❽	$\overline{A \cdot B} = \overline{A} \dotplus \overline{B}$	$\overline{A \dotplus B} = \overline{A} \cdot \overline{B}$	*De Morgan's laws*
❾	$A \cdot \overline{A} = 0$	$A \dotplus \overline{A} = 1$	*Inverse laws*
❿	$A \dotplus (A \cdot B) = A$	$A \cdot (A \dotplus B) = A$	*Absorption laws*

① The basic properties of Boolean algebra are commutative property, associative Property and distributive property.

② The associative property of Boolean algebra states that the **OR** ing and **AND** ing of several variables results in the same regardless of the grouping of the variables.

③ The commutative property states that the order in which the variables are **OR** ed and **AND** ed makes no difference.

N	AND form	OR form	Identity Name
❶	$A \cdot 1 = A$	$A \dotplus 0 = A$	*Identity laws*
❷	$A \cdot 0 = 0$	$A \dotplus 1 = 1$	*Domination laws*
❸	$A \cdot A = A$	$A \dotplus A = A$	*Idempotent laws*
❹	$\overline{(\overline{A})} = A$		*Double negation laws*
❺	$A \cdot B = B \cdot A$	$A \dotplus B = B \dotplus A$	*Commutative laws*
❻	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$	$(A \dotplus B) \dotplus C = A \dotplus (B \dotplus C)$	*Associative laws*
❼	$A \dotplus (B \cdot C) = (A \dotplus B) \cdot (A \dotplus C)$	$A \cdot (B \dotplus C) = (A \cdot B) \dotplus (A \cdot C)$	*Distributive laws*
❽	$\overline{A \cdot B} = \overline{A} \dotplus \overline{B}$	$\overline{A \dotplus B} = \overline{A} \cdot \overline{B}$	*De Morgan's laws*
❾	$A \cdot \overline{A} = 0$	$A \dotplus \overline{A} = 1$	*Inverse laws*
❿	$A \dotplus (A \cdot B) = A$	$A \cdot (A \dotplus B) = A$	*Absorption laws*

④ The distributive property states that **AND** ing several variables and **OR** ing the result With a single variable is equivalent to **OR** ing the single variable with each of the the several Variables and then **AND** ing the sums.

⑤ De Morgan suggested two theorems that form important part of Boolean algebra. They are,

- The complement of a product is equal to the sum of the complements.
- The complement of a sum term is equal to the product of the complements.

N	AND form	OR form	Identity Name
❶	$A \cdot 1 = A$	$A \dotplus 0 = A$	*Identity laws*
❷	$A \cdot 0 = 0$	$A \dotplus 1 = 1$	*Domination laws*
❸	$A \cdot A = A$	$A \dotplus A = A$	*Idempotent laws*
❹	$\overline{(\overline{A})} = A$		*Double negation laws*
❺	$A \cdot B = B \cdot A$	$A \dotplus B = B \dotplus A$	*Commutative laws*
❻	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$	$(A \dotplus B) \dotplus C = A \dotplus (B \dotplus C)$	*Associative laws*
❼	$A \dotplus (B \cdot C) = (A \dotplus B) \cdot (A \dotplus C)$	$A \cdot (B \dotplus C) = (A \cdot B) \dotplus (A \cdot C)$	*Distributive laws*
❽	$\overline{A \cdot B} = \overline{A} \dotplus \overline{B}$	$\overline{A \dotplus B} = \overline{A} \cdot \overline{B}$	*De Morgan's laws*
❾	$A \cdot \overline{A} = 0$	$A \dotplus \overline{A} = 1$	*Inverse laws*
❿	$A \dotplus (A \cdot B) = A$	$A \cdot (A \dotplus B) = A$	*Absorption laws*

Truth Table of $A \cdot 1$		
A	1	$A \cdot 1$
1	1	1
0	1	0

$\Longrightarrow A \cdot 1 = A$

Truth Table of $A \dotplus 1$		
A	1	$A \dotplus 1$
1	1	1
0	1	1

$\Longrightarrow A \dotplus 1 = 1$

N	AND form	OR form	Identity Name
❶	$A \cdot 1 = A$	$A \dot{+} 0 = A$	*Identity laws*
❷	$A \cdot 0 = 0$	$A \dot{+} 1 = 1$	*Domination laws*
❸	$A \cdot A = A$	$A \dot{+} A = A$	*Idempotent laws*
❹	$\overline{(\overline{A})} = A$		*Double negation laws*
❺	$A \cdot B = B \cdot A$	$A \dot{+} B = B \dot{+} A$	*Commutative laws*
❻	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$	$(A \dot{+} B) \dot{+} C = A \dot{+} (B \dot{+} C)$	*Associative laws*
❼	$A \dot{+} (B \cdot C) = (A \dot{+} B) \cdot (A \dot{+} C)$	$A \cdot (B \dot{+} C) = (A \cdot B) \dot{+} (A \cdot C)$	*Distributive laws*
❽	$\overline{A \cdot B} = \overline{A} \dot{+} \overline{B}$	$\overline{A \dot{+} B} = \overline{A} \cdot \overline{B}$	*De Morgan's laws*
❾	$A \cdot \overline{A} = 0$	$A \dot{+} \overline{A} = 1$	*Inverse laws*
❿	$A \dot{+} (A \cdot B) = A$	$A \cdot (A \dot{+} B) = A$	*Absorption laws*

Truth Table of $A \cdot 0$		
A	0	$A \cdot 0$
1	0	0
0	0	0

$\Longrightarrow A \cdot 0 = 0$

Truth Table of $A \dot{+} 0$		
A	0	$A \dot{+} 0$
1	0	1
0	0	0

$\Longrightarrow A \dot{+} 0 = A$

N	AND form	OR form	Identity Name
1	$A \cdot 1 = A$	$A \dotplus 0 = A$	*Identity laws*
2	$A \cdot 0 = 0$	$A \dotplus 1 = 1$	*Domination laws*
3	$A \cdot A = A$	$A \dotplus A = A$	*Idempotent laws*
4	$\overline{(\overline{A})} = A$		*Double negation laws*
5	$A \cdot B = B \cdot A$	$A \dotplus B = B \dotplus A$	*Commutative laws*
6	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$	$(A \dotplus B) \dotplus C = A \dotplus (B \dotplus C)$	*Associative laws*
7	$A \dotplus (B \cdot C) = (A \dotplus B) \cdot (A \dotplus C)$	$A \cdot (B \dotplus C) = (A \cdot B) \dotplus (A \cdot C)$	*Distributive laws*
8	$\overline{A \cdot B} = \overline{A} \dotplus \overline{B}$	$\overline{A \dotplus B} = \overline{A} \cdot \overline{B}$	*De Morgan's laws*
9	$A \cdot \overline{A} = 0$	$A \dotplus \overline{A} = 1$	*Inverse laws*
10	$A \dotplus (A \cdot B) = A$	$A \cdot (A \dotplus B) = A$	*Absorption laws*

Truth Table of $A \cdot A$

A	$A \cdot A$
1	1
0	0

$\Longrightarrow A \cdot A = A$

Truth Table of $A \dotplus A$

A	$A \dotplus A$
1	1
0	0

$\Longrightarrow A \dotplus A = A$

N	AND form	OR form	Identity Name
❶	$A \cdot 1 = A$	$A \dot{+} 0 = A$	*Identity laws*
❷	$A \cdot 0 = 0$	$A \dot{+} 1 = 1$	*Domination laws*
❸	$A \cdot A = A$	$A \dot{+} A = A$	*Idempotent laws*
❹	$\overline{(\overline{A})} = A$		*Double negation laws*
❺	$A \cdot B = B \cdot A$	$A \dot{+} B = B \dot{+} A$	*Commutative laws*
❻	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$	$(A \dot{+} B) \dot{+} C = A \dot{+} (B \dot{+} C)$	*Associative laws*
❼	$A \dot{+} (B \cdot C) = (A \dot{+} B) \cdot (A \dot{+} C)$	$A \cdot (B \dot{+} C) = (A \cdot B) \dot{+} (A \cdot C)$	*Distributive laws*
❽	$\overline{A \cdot B} = \overline{A} \dot{+} \overline{B}$	$\overline{A \dot{+} B} = \overline{A} \cdot \overline{B}$	*De Morgan's laws*
❾	$A \cdot \overline{A} = 0$	$A \dot{+} \overline{A} = 1$	*Inverse laws*
❿	$A \dot{+} (A \cdot B) = A$	$A \cdot (A \dot{+} B) = A$	*Absorption laws*

Truth Table of $A \cdot \overline{A}$

A	$\overline{A}$	$A \cdot \overline{A}$
1	0	0
0	1	0

$\Longrightarrow A \cdot \overline{A} = 0$

Truth Table of $A \dot{+} \overline{A}$

A	$\overline{A}$	$A \dot{+} \overline{A}$
1	0	1
0	1	1

$\Longrightarrow A \dot{+} \overline{A} = 1$

$$A \dotplus 1 = 1 \cdot (A \dotplus 1) = (A \dotplus \overline{A}) \cdot (A \dotplus 1) = A \dotplus (\overline{A} \cdot 1) = A \dotplus \overline{A} = 1$$

$$A \cdot 0 = 0 \dotplus (A \cdot 0) = (A \cdot \overline{A}) \dotplus (A \cdot 0) = A \cdot (\overline{A} \dotplus 0) = A \cdot \overline{A} = 0$$

$$A \dotplus A = (A \dotplus A) \cdot 1 = (A \dotplus A) \cdot (A \dotplus \overline{A}) = A \dotplus (A \cdot \overline{A}) = A \dotplus 0 = A$$

$$A \cdot A = (A \cdot A) \dotplus 0 = (A \cdot A) \dotplus (A \cdot \overline{A}) = A \cdot (A \dotplus \overline{A}) = A \cdot 1 = A$$

$$A \dotplus (A \cdot B) = (A \cdot 1) \dotplus (A \cdot B) = A \cdot (1 \dotplus B) = A \cdot 1 = A$$

$$A \cdot (A \dotplus B) = (A \cdot A) \dotplus (A \cdot B) = A \cdot (1 \dotplus B) = A \cdot 1 = A$$

✎ **Exercise 4.** Prove the following logic equations using the truth table.

a. $\overline{A \dotplus B} = \overline{A} \cdot \overline{B}$ b. $\overline{A \cdot B} = \overline{A} \dotplus \overline{B}$ c. $\overline{A} \dotplus A \cdot B = \overline{A} \dotplus B$

✎ **Exercise 5.** Simplify the following expressions and check your answer by drawing up truth tables.

a. $ABC \dotplus \overline{A}BC$ b. $A \dotplus \overline{A}BC \dotplus \overline{A}B\overline{C}$ c. $(\overline{A} \dotplus B)(A \dotplus B)$ d. $PQ \dotplus (\overline{P} \dotplus \overline{Q})(R \dotplus S)$

✎ **Exercise 6.** Find the complement of the functions $f_1(x, y, z) = \overline{x}y\overline{z} \dotplus \overline{x}\,\overline{y}z$ and $f_2(x, y, z) = (x \dotplus y \dotplus \overline{z})(x \dotplus \overline{y} \dotplus \overline{z})$. By applying De-Morgan's theorem.

✎ **Exercise 7.** Find a Boolean product of the Boolean variables x, y, and z, or their complements, that has the value **1** if and only if

a. $x = y = 0, z = 1$ b. $x = 0, y = 1, z = 0$ c. $x = 0, y = z = 1$ d. $x = y = z = 0$

✎ **Exercise 8.** Find a Boolean sum of the Boolean variables x, y, and z, or their complements, that has the value **0** if and only if

a. $x = y = 0, z = 1$ b. $x = 0, y = 1, z = 0$ c. $x = 0, y = z = 1$ d. $x = y = z = 0$

1. An Overview
   - In English
   - In Math
2. Basic logic functions
   - AND operator
   - OR operator
   - NOT operator
3. Combinations of logic functions
   - Boolean functions
   - Boolean Identities
   - Proofs of Boolean identities
4. Representing boolean functions
   - Disjunctive normal form
   - Conjunctive normal form
   - Deriving (SOP) and (POS) from a truth table
   - Conversions between Canonical forms

- Logical functions are generally expressed in terms of different combinations of logical variables with their true forms as well as the complement forms.

- Logical functions are generally expressed in terms of different combinations of logical variables with their true forms as well as the complement forms.

- Binary logic values obtained by the logical functions and logic variables are in binary form.

- Logical functions are generally expressed in terms of different combinations of logical variables with their true forms as well as the complement forms.

- Binary logic values obtained by the logical functions and logic variables are in binary form.

- An arbitrary logic function can be expressed in the following forms.
  - Sum of the Products (SOP)
  - Product of the Sums (POS)

- Every Boolean function can be expressed as a Boolean sum of minterms.

Definition (Minterm)

A minterm of the Boolean variables $x_1, x_2, \cdots, x_n$ is a Boolean product $y_1 y_2 \cdots y_n$, where $y_i = x_i$ or $y_i = \overline{x_i}$. Hence, a minterm is a product of n literals[a], with one literal for each variable.

[a] A literal is a Boolean variable or its complement

- Every Boolean function can be expressed as a Boolean sum of minterms.

A minterm is a Boolean product of Boolean variables or their complements.

$$ABC,\ \overline{A}BC,\ \overline{A}B\overline{C},\ A\overline{B}C \cdots$$

- Every Boolean function can be expressed as a Boolean sum of minterms.
- For three variables function, eight minterms are possible as listed in the following table

A	B	C	Minterm
0	0	0	$\boldsymbol{\overline{A}\,\overline{B}\,\overline{C}}$
0	0	1	$\boldsymbol{\overline{A}\,\overline{B}C}$
0	1	0	$\boldsymbol{\overline{A}\,B\,\overline{C}}$
0	1	1	$\boldsymbol{\overline{A}\,B\,C}$
1	0	0	$\boldsymbol{A\,\overline{B}\,\overline{C}}$
1	0	1	$\boldsymbol{A\,\overline{B}\,C}$
1	1	0	$\boldsymbol{A\,B\,\overline{C}}$
1	1	1	$\boldsymbol{A\,B\,C}$

This means, for the above three variables example, if $\boldsymbol{A = 0}$, $\boldsymbol{B = 1}$, $\boldsymbol{C = 1}$ i.e., for input combination of **011**, there is only one combination $\boldsymbol{\overline{A}BC}$ that has the value **1**, the rest of the seven combinations have the value **0**.

- Every Boolean function can be expressed as a Boolean sum of minterms.
- For three variables function, eight minterms are possible as listed in the following table

A	B	C	Minterm
0	0	0	$\overline{A}\,\overline{B}\,\overline{C}$
0	0	1	$\overline{A}\,\overline{B}C$
0	1	0	$\overline{A}\,B\,\overline{C}$
0	1	1	$\overline{A}\,B\,C$
1	0	0	$A\,\overline{B}\,\overline{C}$
1	0	1	$A\,\overline{B}\,C$
1	1	0	$A\,B\,\overline{C}$
1	1	1	$A\,B\,C$

- This shows that every Boolean function can be represented using the Boolean operators $\cdot$, $\dot{+}$, and $^{-}$

- Every Boolean function can be expressed as a Boolean sum of minterms.
- For three variables function, eight minterms are possible as listed in the following table

A	B	C	Minterm
0	0	0	$\overline{A}\,\overline{B}\,\overline{C}$
0	0	1	$\overline{A}\,\overline{B}C$
0	1	0	$\overline{A}\,B\,\overline{C}$
0	1	1	$\overline{A}\,B\,C$
1	0	0	$A\,\overline{B}\,\overline{C}$
1	0	1	$A\,\overline{B}\,C$
1	1	0	$A\,B\,\overline{C}$
1	1	1	$A\,B\,C$

- This shows that every Boolean function can be represented using the Boolean operators $\cdot$, $\dot{+}$, and $-$

Definition (Disjunctive Normal Form)

The disjunctive normal form (DNF) of a degree-n Boolean function f is the unique sum of minterms of the variables $x_1, \cdots, x_n$ that represents f.

- Every Boolean function can be expressed as a Boolean sum of minterms.
- For three variables function, eight minterms are possible as listed in the following table

A	B	C	Minterm
0	0	0	$\overline{A}\,\overline{B}\,\overline{C}$
0	0	1	$\overline{A}\,\overline{B}C$
0	1	0	$\overline{A}\,B\,\overline{C}$
0	1	1	$\overline{A}\,B\,C$
1	0	0	$A\,\overline{B}\,\overline{C}$
1	0	1	$A\,\overline{B}\,C$
1	1	0	$A\,B\,\overline{C}$
1	1	1	$A\,B\,C$

- This shows that every Boolean function can be represented using the Boolean operators $\cdot$, $\dot{+}$, and $\bar{}$

Definition (Disjunctive Normal Form)

The disjunctive normal form (DNF) of a degree-n Boolean function f is the unique sum of minterms of the variables $x_1, \cdots, x_n$ that represents f.

- Because every Boolean function can be represented using these operators we say that the set $\{\cdot, \dot{+}, \bar{}\}$ is **functionally complete**.

✯ Canonical sum of product expression: ✯

1. When a Boolean function is expressed as the logical sum of all the minterms from the rows of a truth table, for which the value of the function is 1, it is referred to as the canonical sum of product expression.

✫ Canonical sum of product expression: ✫

1. When a Boolean function is expressed as the logical sum of all the minterms from the rows of a truth table, for which the value of the function is 1, it is referred to as the canonical sum of product expression.
2. The same can be expressed in a compact form by listing the corresponding decimal equivalent codes of the minterms containing a function value of **1**.

✯ Canonical sum of product expression: ✯

1. When a Boolean function is expressed as the logical sum of all the minterms from the rows of a truth table, for which the value of the function is 1, it is referred to as the canonical sum of product expression.
2. The same can be expressed in a compact form by listing the corresponding decimal equivalent codes of the minterms containing a function value of **1**.

Example

For example, if the canonical sum of product form of a three-variable logic function F has the minterms $\overline{A}BC$, $A\overline{B}C$, and $AB\overline{C}$, this can be expressed as the sum of the decimal codes corresponding to these minterms as below.

$$
\begin{aligned}
f(A,B,C) &= \sum(3,5,6) \\
&= m_3 + m_5 + m_6 \\
&= \overline{A}BC + A\overline{B}C + AB\overline{C}
\end{aligned}
$$

where $\sum(3,5,6)$ represents the summation of minterms corresponding to decimal codes **3**, **5**, and **6**.

✫ Conversion to sum of minterms: ✫

The canonical sum of products form of a logic function can be obtained by using the following procedure:

1. Expand the expression in to sum of **AND** terms (SOP) using distributive law.

✯ Conversion to sum of minterms: ✯

The canonical sum of products form of a logic function can be obtained by using the following procedure:

1. Expand the expression in to sum of **AND** terms (SOP) using distributive law.

2. Check each term (**AND** term) in the given logic function. Each term is inspected to contains all the variables. Retain if it is the case (minterm), continue to examine the next term in the same manner.

✯ Conversion to sum of minterms: ✯

The canonical sum of products form of a logic function can be obtained by using the following procedure:

1. Expand the expression in to sum of **AND** terms (SOP) using distributive law.

2. Check each term (**AND** term) in the given logic function. Each term is inspected to contains all the variables. Retain if it is the case (minterm), continue to examine the next term in the same manner.

3. Examine for the variables that are missing in each product which is not a minterm. . If it missing one or more variables, it is ANDed with an expression ($\boldsymbol{X}\dotplus\overline{\boldsymbol{X}}$), where $\boldsymbol{X}$ is one of the missing variable.

✭ Conversion to sum of minterms: ✭

The canonical sum of products form of a logic function can be obtained by using the following procedure:

1. Expand the expression in to sum of **AND** terms (SOP) using distributive law.

2. Check each term (**AND** term) in the given logic function. Each term is inspected to contains all the variables. Retain if it is the case (minterm), continue to examine the next term in the same manner.

3. Examine for the variables that are missing in each product which is not a minterm. . If it missing one or more variables, it is ANDed with an expression ($X \dotplus \overline{X}$), where X is one of the missing variable.

4. Then we use the distributive property $XY(Z \dotplus \overline{Z}) = XYZ \dotplus XY\overline{Z}$ to obtain the sum of products of literals.

✯ Conversion to sum of minterms: ✯

The canonical sum of products form of a logic function can be obtained by using the following procedure:

1. Expand the expression in to sum of **AND** terms (SOP) using distributive law.

2. Check each term (**AND** term) in the given logic function. Each term is inspected to contains all the variables. Retain if it is the case (minterm), continue to examine the next term in the same manner.

3. Examine for the variables that are missing in each product which is not a minterm. . If it missing one or more variables, it is ANDed with an expression ($\boldsymbol{X \dotplus \overline{X}}$), where $\boldsymbol{X}$ is one of the missing variable.

4. Then we use the distributive property $\boldsymbol{XY(Z \dotplus \overline{Z}) = XYZ \dotplus XY\overline{Z}}$ to obtain the sum of products of literals.

5. Finally, we remove all term (minterm) that appear more than once.

We will use examples to illustrate one important way to find a Boolean expression that represents a Boolean function.

Example

Find the canonical sum of product expansion of the Boolean variables $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$, or their complements of the Boolean function $\boldsymbol{f(x, y, z) = (x + z)y}$ by using the boolean identities.

Solution

$$f(x, y, z) = (x + z)y$$

We will use examples to illustrate one important way to find a Boolean expression that represents a Boolean function.

Example

Find the canonical sum of product expansion of the Boolean variables $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$, or their complements of the Boolean function $\boldsymbol{f(x, y, z) = (x + z)y}$ by using the boolean identities.

Solution

$$\boldsymbol{f(x, y, z) = (x + z)y = xy + zy} \qquad \text{(Distributive Law)}$$

We will use examples to illustrate one important way to find a Boolean expression that represents a Boolean function.

Example

Find the canonical sum of product expansion of the Boolean variables $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$, or their complements of the Boolean function $\boldsymbol{f(x, y, z) = (x + z)y}$ by using the boolean identities.

Solution

$$\boldsymbol{f(x, y, z) = (x + z)y = xy + zy} \qquad \text{(Distributive Law)}$$

$$\boldsymbol{= xy.1 + zy.1} \qquad \text{(Identity Law)}$$

We will use examples to illustrate one important way to find a Boolean expression that represents a Boolean function.

Example

Find the canonical sum of product expansion of the Boolean variables x, y, and z, or their complements of the Boolean function $f(x, y, z) = (x + z)y$ by using the boolean identities.

Solution

$$\begin{aligned} f(x, y, z) = (x + z)y &= xy + zy && \text{(Distributive Law)} \\ &= xy.1 + zy.1 && \text{(Identity Law)} \\ &= xy(z + \overline{z}) + zy(x + \overline{x}) && \text{(Unit Property)} \end{aligned}$$

We will use examples to illustrate one important way to find a Boolean expression that represents a Boolean function.

Example

Find the canonical sum of product expansion of the Boolean variables $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$, or their complements of the Boolean function $\boldsymbol{f(x, y, z) = (x + z)y}$ by using the boolean identities.

Solution

$$
\begin{aligned}
f(x, y, z) &= (x + z)y = xy + zy && \text{(Distributive Law)}\\
&= xy.1 + zy.1 && \text{(Identity Law)}\\
&= xy(z + \overline{z}) + zy(x + \overline{x}) && \text{(Unit Property)}\\
&= xyz + xy\overline{z} + zyx + zy\overline{x} && \text{(Distributive Law)}
\end{aligned}
$$

We will use examples to illustrate one important way to find a Boolean expression that represents a Boolean function.

Example

Find the canonical sum of product expansion of the Boolean variables $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$, or their complements of the Boolean function $\boldsymbol{f(x, y, z) = (x + z)y}$ by using the boolean identities.

Solution

$$
\begin{aligned}
f(x,y,z) &= (x+z)y = xy + zy && \text{(Distributive Law)}\\
&= xy.1 + zy.1 && \text{(Identity Law)}\\
&= xy(z+\overline{z}) + zy(x+\overline{x}) && \text{(Unit Property)}\\
&= xyz + xy\overline{z} + zyx + zy\overline{x} && \text{(Distributive Law)}\\
&= xyz + xy\overline{z} + yxz + \overline{x}yz && \text{(Commutative Law)}
\end{aligned}
$$

We will use examples to illustrate one important way to find a Boolean expression that represents a Boolean function.

Example

Find the canonical sum of product expansion of the Boolean variables $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$, or their complements of the Boolean function $\boldsymbol{f(x, y, z) = (x + z)y}$ by using the boolean identities.

Solution

$$
\begin{aligned}
\boldsymbol{f(x,y,z)} &= \boldsymbol{(x+z)y = xy + zy} && \text{(Distributive Law)}\\
&= \boldsymbol{xy.1 + zy.1} && \text{(Identity Law)}\\
&= \boldsymbol{xy(z+\overline{z}) + zy(x+\overline{x})} && \text{(Unit Property)}\\
&= \boldsymbol{xyz + xy\overline{z} + zyx + zy\overline{x}} && \text{(Distributive Law)}\\
&= \boldsymbol{xyz + xy\overline{z} + yxz + \overline{x}yz} && \text{(Commutative Law)}\\
&= \boldsymbol{xyz + xy\overline{z} + \overline{x}yz} && \text{(Idempotent Law)}
\end{aligned}
$$

We will use examples to illustrate one important way to find a Boolean expression that represents a Boolean function.

Example

Find the canonical sum of product expansion of the Boolean variables x, y, and z, or their complements of the Boolean function $f(x, y, z) = (x + z)y$ by using the boolean identities.

Solution

$$\begin{aligned}
f(x,y,z) &= (x+z)y = xy + zy && \text{(Distributive Law)}\\
&= xy.1 + zy.1 && \text{(Identity Law)}\\
&= xy(z+\overline{z}) + zy(x+\overline{x}) && \text{(Unit Property)}\\
&= xyz + xy\overline{z} + zyx + zy\overline{x} && \text{(Distributive Law)}\\
&= xyz + xy\overline{z} + yxz + \overline{x}yz && \text{(Commutative Law)}\\
&= xyz + xy\overline{z} + \overline{x}yz && \text{(Idempotent Law)}
\end{aligned}$$

✎ **Exercise 9.** Find the canonical sum of product expansion of the Boolean variables x, y, and z, or their complements of the following Boolean functions by using the boolean identities:
a. $f(x, y, z) = (x + y)\overline{z}$ b. $f(x, y, z) = x\overline{y}$ c. $f(x, y, z) = x$

- Every Boolean function can be expressed as a Boolean sum of maxterms.

Definition (Maxterm)

A maxterm of the Boolean variables $x_1, x_2, \cdots, x_n$ is a Boolean sum $y_1 + y_2 + \cdots + y_n$, where $y_i = x_i$ or $y_i = \overline{x_i}$. Hence, a maxterm is a sum of n literals[a], with one literal for each variable.

[a] A literal is a Boolean variable or its complement

- Every Boolean function can be expressed as a Boolean sum of maxterms.

A maxterm is a Boolean sum of Boolean variables or their complements.

$$\boldsymbol{A+B+C,\ \overline{A}+B+C,\ \overline{A}+B+\overline{C},\ A+\overline{B}+C\cdots}$$

- Every Boolean function can be expressed as a Boolean sum of maxterms.
- For three variables function, eight minterms are possible as listed in the following table

A	B	C	Minterm
0	0	0	$A+B+C$
0	0	1	$A+B+\overline{C}$
0	1	0	$A+\overline{B}+C$
0	1	1	$A+\overline{B}+\overline{C}$
1	0	0	$\overline{A}+B+C$
1	0	1	$\overline{A}+B+\overline{C}$
1	1	0	$\overline{A}+\overline{B}+C$
1	1	1	$\overline{A}+\overline{B}+\overline{C}$

This means, for the above three variables example, if $A = 1$, $B = 1$, $C = 0$ i.e., for input combination of 110, there is only one combination $\overline{A}+\overline{B}+C$ that has the value **0**, the rest of the seven combinations have the value **1**

- Every Boolean function can be expressed as a Boolean sum of maxterms.
- For three variables function, eight minterms are possible as listed in the following table

A	B	C	Minterm
0	0	0	$\boldsymbol{A+B+C}$
0	0	1	$\boldsymbol{A+B+\overline{C}}$
0	1	0	$\boldsymbol{A+\overline{B}+C}$
0	1	1	$\boldsymbol{A+\overline{B}+\overline{C}}$
1	0	0	$\boldsymbol{\overline{A}+B+C}$
1	0	1	$\boldsymbol{\overline{A}+B+\overline{C}}$
1	1	0	$\boldsymbol{\overline{A}+\overline{B}+C}$
1	1	1	$\boldsymbol{\overline{A}+\overline{B}+\overline{C}}$

- This shows that every Boolean function can be represented using the Boolean operators $\cdot$, $\dot{+}$, and $\_$

- Every Boolean function can be expressed as a Boolean sum of maxterms.
- For three variables function, eight minterms are possible as listed in the following table

A	B	C	Minterm
0	0	0	$A+B+C$
0	0	1	$A+B+\overline{C}$
0	1	0	$A+\overline{B}+C$
0	1	1	$A+\overline{B}+\overline{C}$
1	0	0	$\overline{A}+B+C$
1	0	1	$\overline{A}+B+\overline{C}$
1	1	0	$\overline{A}+\overline{B}+C$
1	1	1	$\overline{A}+\overline{B}+\overline{C}$

- This shows that every Boolean function can be represented using the Boolean operators $\cdot$, $\dot{+}$, and $-$

Definition (Conjunctive Normal Form)

The conjunctive normal form (CNF) of a degree-n Boolean function f is the unique product of maxterms of the variables $x_1, \cdots, x_n$ that represents f.

- Every Boolean function can be expressed as a Boolean sum of maxterms.
- For three variables function, eight minterms are possible as listed in the following table

A	B	C	Minterm
0	0	0	$A+B+C$
0	0	1	$A+B+\overline{C}$
0	1	0	$A+\overline{B}+C$
0	1	1	$A+\overline{B}+\overline{C}$
1	0	0	$\overline{A}+B+C$
1	0	1	$\overline{A}+B+\overline{C}$
1	1	0	$\overline{A}+\overline{B}+C$
1	1	1	$\overline{A}+\overline{B}+\overline{C}$

- This shows that every Boolean function can be represented using the Boolean operators $\cdot$, $\dot{+}$, and $^{-}$

Definition (Conjunctive Normal Form)

The conjunctive normal form (CNF) of a degree-n Boolean function f is the unique product of maxterms of the variables $x_1, \cdots, x_n$ that represents f.

- Because every Boolean function can be represented using these operators we say that the set $\{\cdot, \dot{+}, ^{-}\}$ is **functionally complete**.

✯ Canonical product of sum expression: ✯

1. When a Boolean function is expressed as the logical product of all the maxterms from the rows of a truth table, for which the value of the function is 0, it is referred to as the *canonical product of sum expression*.

✯ Canonical product of sum expression: ✯

1. When a Boolean function is expressed as the logical product of all the maxterms from the rows of a truth table, for which the value of the function is 0, it is referred to as the *canonical product of sum expression*.
2. The same can be expressed in a compact form by listing the corresponding decimal equivalent codes of the maxterms containing a function value of **0**.

✯ Canonical product of sum expression: ✯

1. When a Boolean function is expressed as the logical product of all the maxterms from the rows of a truth table, for which the value of the function is 0, it is referred to as the *canonical product of sum expression*.
2. The same can be expressed in a compact form by listing the corresponding decimal equivalent codes of the maxterms containing a function value of **0**.

Example

For example, if the canonical sum of product form of a three-variable logic function F has the minterms $\boldsymbol{A+B+C}$, $\boldsymbol{A+\overline{B}+C}$, and $\boldsymbol{\overline{A}+B+\overline{C}}$, this can be expressed as the product of the decimal codes corresponding to these maxterms as below.

$$\begin{aligned} f(A,B,C) &= \prod(0,2,5) \\ &= M_0 M_2 M_5 \\ &= (A+B+C)(A+\overline{B}+C)(\overline{A}+B+\overline{C}) \end{aligned}$$

where $\prod(\mathbf{0,2,5})$ represents the product of maxterms corresponding to decimal codes **0**, **2**, and **5**.

✫ Conversion to product of maxterms: ✫

The canonical product of sums form of a logic function can be obtained by using the following procedure.

1. Expand the expression in to product of **OR** terms (POS) using distributive law.

✯ Conversion to product of maxterms: ✯

The canonical product of sums form of a logic function can be obtained by using the following procedure.

1. Expand the expression in to product of **OR** terms (POS) using distributive law.

2. Check each term (**OR** term) in the given logic function. Each term is inspected to see if it contains all the variables. Retain it if it is the case (maxterm), continue to examine the next term in the same manner.

✫ Conversion to product of maxterms: ✫

The canonical product of sums form of a logic function can be obtained by using the following procedure.

1. Expand the expression in to product of **OR** terms (POS) using distributive law.

2. Check each term (**OR** term) in the given logic function. Each term is inspected to see if it contains all the variables. Retain it if it is the case (maxterm), continue to examine the next term in the same manner.

3. Examine for the variables that are missing in each sum term that is not a maxterm. If it missing one or more variables, it is ORed with an expression ($\boldsymbol{X\overline{X}}$), where $\boldsymbol{X}$ is one of the missing variable.

✫ Conversion to product of maxterms: ✫

The canonical product of sums form of a logic function can be obtained by using the following procedure.

1. Expand the expression in to product of **OR** terms (POS) using distributive law.
2. Check each term (**OR** term) in the given logic function. Each term is inspected to see if it contains all the variables. Retain it if it is the case (maxterm), continue to examine the next term in the same manner.
3. Examine for the variables that are missing in each sum term that is not a maxterm. If it missing one or more variables, it is ORed with an expression ($\boldsymbol{X\overline{X}}$), where $\boldsymbol{X}$ is one of the missing variable.
4. Then we use the distributive property $\boldsymbol{X + YZ = (X + Y)(X + Z)}$ to obtain the product of sum of literals.

✯ Conversion to product of maxterms: ✯

The canonical product of sums form of a logic function can be obtained by using the following procedure.

1. Expand the expression in to product of **OR** terms (POS) using distributive law.

2. Check each term (**OR** term) in the given logic function. Each term is inspected to see if it contains all the variables. Retain it if it is the case (maxterm), continue to examine the next term in the same manner.

3. Examine for the variables that are missing in each sum term that is not a maxterm. If it missing one or more variables, it is ORed with an expression ($X\overline{X}$), where X is one of the missing variable.

4. Then we use the distributive property $X + YZ = (X + Y)(X + Z)$ to obtain the product of sum of literals.

5. Finally, we remove all term (maxterm) that appear more than once.

We will use examples to illustrate one important way to find a Boolean expression that represents a Boolean function.

Example

Obtain the canonical product of the sum form of the following function.
$F(A, B, C) = (A + \overline{B})(B + C)(A + \overline{C})$

Solution

$$F(A, B, C) = (A + \overline{B} + 0)(B + C + 0)(A + \overline{C} + 0) \qquad \text{(Identity Law)}$$

We will use examples to illustrate one important way to find a Boolean expression that represents a Boolean function.

Example

Obtain the canonical product of the sum form of the following function.
$F(A, B, C) = (A + \overline{B})(B + C)(A + \overline{C})$

Solution

$$\begin{aligned} F(A, B, C) &= (A + \overline{B} + 0)(B + C + 0)(A + \overline{C} + 0) && \text{(Identity Law)} \\ &= (A + \overline{B} + C\overline{C})(B + C + A\overline{A})(A + \overline{C} + B\overline{B}) && \text{(Inverse Law)} \end{aligned}$$

We will use examples to illustrate one important way to find a Boolean expression that represents a Boolean function.

Example

Obtain the canonical product of the sum form of the following function.

$F(A, B, C) = (A + \overline{B})(B + C)(A + \overline{C})$

Solution

$$\begin{aligned} F(A, B, C) &= (A + \overline{B} + 0)(B + C + 0)(A + \overline{C} + 0) && \text{(Identity La} \\ &= (A + \overline{B} + C\overline{C})(B + C + A\overline{A})(A + \overline{C} + B\overline{B}) && \text{(Inverse La} \\ &= (A + \overline{B} + C)(A + \overline{B} + \overline{C})(B + C + A)(B + C + \overline{A})(A + \overline{C} + B)(A + \overline{C} + \overline{B}) && \text{(Distributive La} \end{aligned}$$

We will use examples to illustrate one important way to find a Boolean expression that represents a Boolean function.

Example

Obtain the canonical product of the sum form of the following function.
$F(A,B,C) = (A+\overline{B})(B+C)(A+\overline{C})$

Solution

$$\begin{aligned}
F(A,B,C) &= (A+\overline{B}+0)(B+C+0)(A+\overline{C}+0) && \text{(Identity La} \\
&= (A+\overline{B}+C\overline{C})(B+C+A\overline{A})(A+\overline{C}+B\overline{B}) && \text{(Inverse La} \\
&= (A+\overline{B}+C)(A+\overline{B}+\overline{C})(B+C+A)(B+C+\overline{A})(A+\overline{C}+B)(A+\overline{C}+\overline{B}) && \text{(Distributive La} \\
&= (A+\overline{B}+C)(A+\overline{B}+\overline{C})(A+B+C)(\overline{A}+B+C)(A+B+\overline{C}) && \text{(Idempotent La}
\end{aligned}$$

✎ **Exercise 10.** Express the Boolean function $f(x, y, z) = xy + \overline{x}z$ in a product of maxterms.

✎ **Exercise 11.** Find the sum-of-products and product-of-sums expansion of each of the Boolean functions:

a. $f(x, y, z) = x + y + z$

b. $f(x, y, z) = (x + z)y$

c. $f(x, y, z) = x$

b. $f(x, y, z) = x\overline{y}$

✎ **Exercise 12.** Express each of these Boolean functions using the operators $\dot{+}$ and $-$

a. $f(x, y, z) = x + y + z$

b. $f(x, y, z) = x + \overline{y}(\overline{x} + z)$

c. $f(x, y, z) = \overline{x + \overline{y}}$

b. $f(x, y, z) = \overline{x}(x + \overline{y} + \overline{z})$

✫ Deriving a sum of products (SOP) and product of sum (POS) expression from a truth table ✫

- The sum of products (SOP) expression of a Boolean function can be obtained from its truth table by summing or performing OR operation of the product terms corresponding to the combinations containing a function value of **1**.

  In the product terms the input variables appear either in true (uncomplemented) form if it contains the value **1**, or in complemented form if it possesses the value **0**.

✯ Deriving a sum of products (SOP) and product of sum (POS) expression from a truth table ✯

- The sum of products (SOP) expression of a Boolean function can be obtained from its truth table by summing or performing OR operation of the product terms corresponding to the combinations containing a function value of **1**.

> In the product terms the input variables appear either in true (uncomplemented) form if it contains the value **1**, or in complemented form if it possesses the value **0**.

- The product of sums (POS) expression of a Boolean function can also be obtained from its truth table by a similar procedure. Here, an AND operation is performed on the sum terms corresponding to the combinations containing a function value of **0**.

> In the sum terms the input variables appear either in true (uncomplemented) form if it contains the value **0**, or in complemented form if it possesses the value **1**.

Example

Now, consider the following truth table , for a three-input function $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$.

Here

- the output $\boldsymbol{Y = f(A, B, C)}$ value is **1** for the input conditions of **010**, **100**, **101**, and **110**, and their corresponding product terms are $\overline{A}B\overline{C}$, $A\overline{B}\,\overline{C}$, $A\overline{B}C$, and $AB\overline{C}$ respectively.
- the output $\boldsymbol{Y}$ value is **0** for the input conditions of **000**, **001**, **011**, and **111**, and their corresponding product terms are $A + B + C$, $A + B + \overline{C}$, $A + \overline{B} + \overline{C}$, and $\overline{A} + \overline{B} + \overline{C}$ respectively.

Inputs			Output	minterms	Maxterms
A	B	C	Y		
0	0	0	0		$A + B + C$
0	0	1	0		$A + B + \overline{C}$
0	1	0	1	$\overline{A}B\overline{C}$	
0	1	1	0		$A + \overline{B} + \overline{C}$
1	0	0	1	$A\overline{B}\,\overline{C}$	
1	0	1	1	$A\overline{B}C$	
1	1	0	1	$AB\overline{C}$	
1	1	1	0		$\overline{A}+\overline{B} + \overline{C}$

SOP $Y = \overline{A}B\overline{C} + A\overline{B}\,\overline{C} + A\overline{B}C + AB\overline{C}$

POS $Y = (A + B + C)(A + B + \overline{C})(A + \overline{B} + \overline{C})(\overline{A} + \overline{B} + \overline{C})$

✯ Conversions between Canonical forms ✯

- We noted that the complement of a function expressed as the sum of products (SOP) equals to the products of sum (POS).

✯ Conversions between Canonical forms ✯

- We noted that the complement of a function expressed as the sum of products (SOP) equals to the products of sum (POS).
- This is because the original function is expressed by those minterms that make the function equal to **1**, while its complement is **1** for those maxterms whose values are **0**.

✭ Conversions between Canonical forms ✭

- We noted that the complement of a function expressed as the sum of products (SOP) equals to the products of sum (POS).
- This is because the original function is expressed by those minterms that make the function equal to **1**, while its complement is **1** for those maxterms whose values are **0**.

To convert from on canonical form to another:

1. interchanging the symbols $\prod$ and $\sum$

✯ Conversions between Canonical forms ✯

- We noted that the complement of a function expressed as the sum of products (SOP) equals to the products of sum (POS).
- This is because the original function is expressed by those minterms that make the function equal to **1**, while its complement is **1** for those maxterms whose values are **0**.

To convert from on canonical form to another:

1. interchanging the symbols $\prod$ and $\sum$
2. then list those numbers missing in the original form.

✯ Conversions between Canonical forms ✯

- We noted that the complement of a function expressed as the sum of products (SOP) equals to the products of sum (POS).
- This is because the original function is expressed by those minterms that make the function equal to **1**, while its complement is **1** for those maxterms whose values are **0**.

To convert from on canonical form to another:

1. interchanging the symbols $\prod$ and $\sum$
2. then list those numbers missing in the original form.
3. In order to find the missing terms, one must realize the total number of minterms and maxterms is $\mathbf{2^n}$, where $\boldsymbol{n}$ is the number of binary variables in the function

✭ Conversions between Canonical forms ✭

- We noted that the complement of a function expressed as the sum of products (SOP) equals to the products of sum (POS).
- This is because the original function is expressed by those minterms that make the function equal to **1**, while its complement is **1** for those maxterms whose values are **0**.

To convert from on canonical form to another:

1. interchanging the symbols $\prod$ and $\sum$
2. then list those numbers missing in the original form.
3. In order to find the missing terms, one must realize the total number of minterms and maxterms is $\mathbf{2^n}$, where $\boldsymbol{n}$ is the number of binary variables in the function

Example

Convert the Boolean function $\boldsymbol{f(x, y, z) = \sum(0, 2, 4, 5)}$ to the other canonical form.

Solution

The number of variables is three $\boldsymbol{(x, y, z)}$ therefore the total numbers is in range $\mathbf{(0 \cdots 2^3 - 1) = (0 \cdots 7)}$ therefore, $\boldsymbol{f(x, y, z) = \prod(1, 3, 6, 7)}$

Thank you! Questions?