06-03-2023

1. PROPOSITIONAL LOGIC:

* Mathematical Logic:-

→ Science of Reasoning. is called Mathematical Logic.

→ It is two types.

① Propositional logic.

② Predicate Logic.

* Proposition (or) statement:-

→ It is a 'declarative sentence' which is either true (or) False but not both.

→ If proposition is 'True' then its truth value is 'True' ['T' or '1'].

→ If proposition is 'False' then its truth value is 'False' ['F' or '0'].

→ Sometimes it will be 'Two valued Logic'

Ex:-

① Today is sunday [06-03-23] - False

② 3+4=7. - True.

③ It will raining on this day of next year. - T or F.

④ Goldbach's conjecture:- [1794]

→ Every even positive integer greater than 2 is sum of two prime numbers.

→ It is a proposition [T or F].

* Sentences which are not propositions:-

① 'All questions'

② 'commands'

③ 'Requests'/'suggestions'.

④ 'Exclamatory sentences'.

⑤ Opinions:

Ex:- I like tea - Not a proposition.

⑥ Self controdictory sentences:

Ex:- $x+2=3$.

if $x=1$: True , else: False

* Which af the following are propositions

① 2 is an even integer.

A) Proposition [Truth value is T]

② Why should we study Discrete Mathematics?

A) Not a proposition [Question].

③ There is an integer x, such that $x^2=3$.

A) Proposition [Truth value is F].

④ Please, Be quite.

A) Not a proposition.

⑤ Dogs can Fly.

A) Proposition [Truth value is F].

⑥ There will be snow in December.

A) Proposition [T or F].

⑦ What a beautiful it is!

A) Not a proposition [Exclamatory sentence]

⑧ Get up and do your exercise.

A) Not a proposition. It is a command.

* Needs of Proposition / Requirements :-

→ Proposition needs two languages.

① "object Language" [Logical]

② "Meta Language" [English]

→ Propositional Logic is also called "Two valued - Language" and "symbolic Language".

* Propositional variables :-

→ The variables which represents the propositions are called Propositional variables.

Ex :- p, q, r, s

P, Q, R, S

P_1, P_2, P_3, P_4.

Ex :- P : Today is Friday.

* Types of Propositions :-

① Atomic / Primitive / Primary / simple :-

→ 'Atomic' can't be ~~in~~divisible. [Can't be breakable]

→ This type represents the 'Fundamentals' of propositions

② compound / composite / Molecular :-

→ This type can be breakable.

→ The combination of atomic propositions is called 'compound Propositions'.

Ex :- Rahim is good boy. He has a watch.

① Rahim is good boy.

② He has a watch.

* Note :-

→ The variables which represents the Atomic propositions are called "Atomic variables."

→ The variables which represents the compound propositions are called "compound variables."

* Logical operators :-

→ These are different types like.

① 'Negation' ($\sim$) — Unary operator

② 'conjunction' ($\wedge$)

③ 'Disjunction' ($\vee$)

④ 'Implication / 'conditional' ($\rightarrow$)

⑤ 'Bi Implication' ($\leftrightarrow$)

} Binary operator.

⑥ 'Exclusive OR' ($\oplus$) — Extra

① Negation ($\sim$) :-

→ 'P' is a propositional variable then Negation denotes that "It is not the case that P" (or) 'not P'

Notation :- $\sim P$ (or) $\neg P$ (or) $\bar{P}$ (or) NOT P.

Ex :-

① P : Today is sunday. (F)

$\sim P$: It is not the case that Today is sunday.

(or)

$\sim P$: not Today is sunday.

(or)

$\sim P$: Today is not sunday. (T)

② P : $2+3 = 5$ (T)

$\sim P$: $2+3 \neq 5$ (F)

* Truth Table for Negation:-

P	~P
T	F
F	T

Ex:-

③ p: All students are intelligent.

q: Atleast one student is not intelligent.

(or)

q: some students are not intelligent.

r: No students is integlligent.

p	q	r
T	F	F
F	T	F

∴ q is negation for p. [But not r].

* Literals:-

→ All atomic variables and all negation atomic variables are called 'Literals'.

(or)

→ All atomic variables and their Negations are called 'Literals'

Ex:- p, q, ~p, ~q.

All are Literals

② conjunction (∧) :-

→ Let 'p','q' are atomic variables then it is defined as "p and q", sometimes, p but q.

Notation :- '$p \wedge q$'

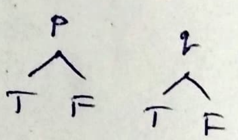
→ Here 'p','q' are atomic variables and '$p \wedge q$' is compound variable, it represents composition Proposition.

→ 'and' belongs to 'object Language' (∧), 'but' belongs to 'Meta Language'.

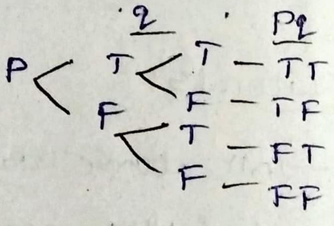
→ $$p \wedge q = \begin{cases} T, & \text{if } p \text{ and } q \text{ both are True.} \\ F, & \text{otherwise it will False.} \end{cases}$$

Truth Table for conjunction :-

p	q	$p \wedge q$	$q \wedge p$
T	T	T	T
T	F	F	F
F	T	F	F
F	F	F	F



$2^n = 2^2 = 4$.



Ex:-

① p : Ramu is healthy. (T)

q : He has blue eyes (T)

$p \wedge q$: Ramu is healthy and He has blue eyes. (T)

② p : 2+3=5 (T)

q : London is capital of France. (F)

$p \wedge q$: 2+3=5 and London is capital of France (F)

Note :- We can combine Non-Related propositions also.

③ Disjunction ($\vee$) :-

→ Meta language word for $\vee$ is 'or' / 'either or' / 'both'. [$\vee$ – wedge].

Notation :- '$p \vee q$' (or) 'p or q'.

$$\rightarrow p \vee q = \begin{cases} T, & \text{if } p \text{ is true (or) } q \text{ is true (or) both are true} \\ F, & \text{otherwise.} \end{cases}$$

(or)

$$p \vee q = \begin{cases} F, & \text{Both are F} \\ T, & \text{otherwise.} \end{cases}$$

Truth Table for $p \vee q$:-

p	q	$p \vee q$	$q \vee p$
T	T	T	T
T	F	T	T
F	T	T	T
F	F	F	F

Note :-

① Inclusive or :- 'either or (or) both'.

② Exclusive or :- 'either (or)' (or) 'but not both'.

→ Therefore 'Disjunction' is equivalent to 'Inclusive or'

Ex :-

① p: Mark is Rich. (T)

q: Mark is Unhappy (F)

$p \vee q$: Mark is Rich or Mark is unhappy. (T)

②
p: 2+3=5 (T)

q: London is the capital of France. (T)

2+3=5 or London is capital of France or both $= p \vee q$ (T)

④ Conditional statement (or) Implication:-

→ The conditional statement is denoted by '→', '⊃'
↑ Horseshoe

→ Meta Language word for '→' ≡ 'If __ then'
→ Implies

Notation:- '$p \to q$' stands for 'If p then q'

Definition:-

→ p and q are two atomic proposition gives a compound proposition.

$$\boxed{p \longrightarrow q}$$

↓ Hypothesis (or) Antecedent (or) premise

↓ conclusion (or) consequence

$$p \to q = \begin{cases} F, & \text{if } p \text{ is } T \text{ and } q \text{ is } F \\ T, & \text{otherwise.} \end{cases}$$

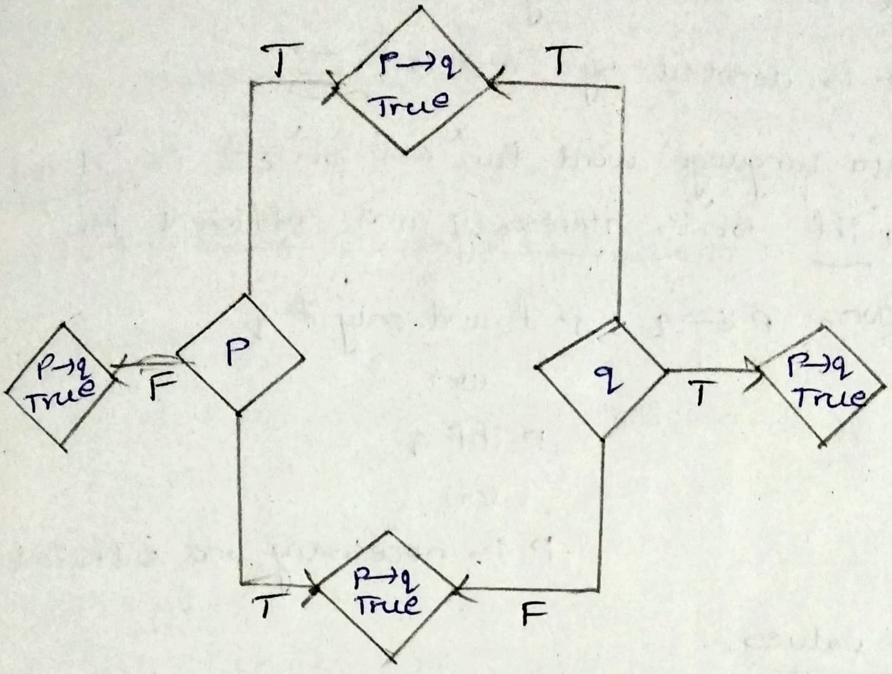
Truth Table:-

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

Conclusions:-

1. If Hypothesis is 'False' then conditional implication is 'True'.
2. If conclusion is 'True' then corresponding conditional composition is 'True'.
3. If Both are 'True' then conditional compositional proposition is 'True'.
4. If Hypothesis is 'True' and conclusion is 'False' then corresponding conditional proposition is 'False'.

* Pictorial Representation :-



* Ex :-

① p : 2+3 = 5

q : London is the capital of France.

$p \to q$: If 2+3=5 then London is capital of France.

② p : I get the salary (T)

q : I will pay off your debt.

$p \to q$: If I get the salary then I will pay off your debt.

* Equivalents for '$p \to q$' in Meta language :

① If p then q

② If p, q

③ q if p

④ q when p.

⑤ q whenever p

⑥ q follows from p.

⑦ q is necessary for p

⑧ p is sufficient for q.

⑨ p only if q.

⑤ Biconditional (or) Bi Implication:-

→ It is also Binary.

→ It is denoted by '$\leftrightarrow$' (or) '$\rightleftarrows$'

→ Meta Language word for '$\leftrightarrow$' (or) '$\rightleftarrows$' is 'If and only if' (or) 'iff' (or) is 'Necessary and sufficient for.'

Notation:- $p \leftrightarrow q$: p if and only if q

(or)

p iff q

(or)

p is necessary and sufficient for q

Truth values:-

$$p \leftrightarrow q = \begin{cases} T, & \text{if Both have same Truthvalues} \\ F, & \text{otherwise.} \end{cases}$$

Truth Table:-

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Ex:-

① p: House sales fall

q: Interest rates raises.

$p \leftrightarrow q$: House sales fail if and only if Interest rates raises.

*** Note:-**

Another Notation for Bi Implication is

$p \rightleftarrows q$

$p \rightarrow q$ and $q \rightarrow p$

"If p then q and conversely."

* Converse, Inverse and contra positive statements:-

① converse:-

→ The converse of 'p → q' is 'q → p'.

② Inverse:-

→ The Inverse of 'p → q' is '~p → ~q'.

③ contrapositive:-

→ The contrapositive statement of 'p → q' is '~q → ~p'.

Truth Tables:-

p	q	conditional $p \to q$	converse $q \to p$	$\sim p$	$\sim q$	Inverse $\sim p \to \sim q$	contra positive $\sim q \to \sim p$
T	T	T	T	F	F	T	T
T	F	F	T	F	T	T	F
F	T	T	F	T	F	F	T
F	F	T	T	T	T	T	T

→ 'conditional' and 'contrapositive' are logically equal.

→ 'converse' and 'Inverse' are also equivalent.

→ Negation of p implication v q $(\sim p \vee q)$, $(p \to q)$ are Identical.

* Ex-1:-

Which are propositions:-

(i) Delhi is the capital of India.

A) Proposition [T].

(ii) 2+3=4

A) Proposition [F].

Module - 1.2 :- WELL FORMED FORMULAS

10-03-2023

* well-formed Formulas :- (wff) [reads as wuffs].

statement formula :-

→ It is an expression which contains operators, variables and Parenthesis.

→ All statement formulas are not wff's.

well-formed Formula :-

→ A well-formed formula is a statement formula with 'syntactically correct'.

→ It is defined as follows:-

① A 'proposition variable' alone is a wff.

Ex:- $p, q, r, \_\_\_$

② If 'p' is wff then '$\sim p$' is also wff.

Ex:- $\sim p, \sim q, \_\_\_$

③ If 'p' and 'q' are wffs then '$(p \wedge q)$', '$(p \vee q)$', '$(p \to q)$' and '$(p \leftrightarrow q)$' are also wffs.

Note :- Parenthesis is Mandatory here.

④ A string is wff if it can be obtained by using above ①, ② & ③ rules.

Examples :-

① p is wff [by Rule ①]

② $\sim p \vee q$ is not wff [syntax error].

③ $\sim (p \vee q)$ is a wff

④ $(\sim p \wedge q)$ is a wff

⑤ $p \to (q \to r)$ is not a wff

⑥ $(p \to (q \to r))$ is a wff.

⑦ $(p \to (q \to r)$ is not a wff.

⑧ $p \to q \to r$ is not a wff

⑨ $((p \wedge q) \vee (p \wedge r))$ is a wff

⑩ $((p \to q) \to r)$ is not a wff

* **Relaxation to write paranthesis in wff:-**

① For the sake of convenience, we can omit outer brackets of wff.

Ex:- ① $P \rightarrow (q \rightarrow r)$ is wff.

② $\sim p \vee q$ is a wff.

③ $(p \wedge q) \vee (p \wedge r)$ is a wff.

② sometimes a wff is in the form of same operation then we give relaxation.

Ex:- $p \vee q \vee r$ is a wff

* **Order Precedence of a wff:-**

① The order precedence of a wff is as follows:-

① Evaluate 'parenthesis' First.

Note:- Among all parenthesis, inner most is First.

Ex:- $P \rightarrow (q \rightarrow (r \rightarrow (s \rightarrow t)))$

② If wff contains two parenthesis connected by a operator then we go for 'Left to right'.

Ex:- $(p \wedge q) \vee (p \wedge r)$

③

Operator	order Precedence
$\sim$	1
$\wedge$	2
$\vee$	3
$\rightarrow$	4
$\leftrightarrow$	5

* **Types of wff:-**

→ A wff is of Three types:-

① Tautology

② contradiction

③ contingency.

① Tautology :- (T_0).

→ A wff is said to be "Tautology" for each possible assignments of truth values of propositional variables that wff has Truth value 'T' that is the truth ~~variable~~ table column tautology contains only one variable T.

→ Tautology is also called 'universely True'

② contradiction :- (F_0):

→ A wff is said to be contradiction if for each possible assignments of truth values of propositional variables that wff has Truth value 'F'.

→ Another name is 'Absurdity'.

③ contingency :-

→ Another name is 'satisfiable'.

→ A wff is which is neither Tautology nor contradiction is called 'contingency'.

* Examples :-

① Write truth tables for $P \vee \sim P$, $P \wedge \sim P$.

A) $P \vee \sim P$

P	$\sim P$	$P \vee \sim P$
T	F	T
F	T	T

∴ $P \vee \sim P$ is a Tautology

$P \wedge \sim P$.

P	$\sim P$	$P \wedge \sim P$
T	F	F
F	T	F

∴ $P \wedge \sim P$ is a contradiction

11-03-2023

Module - 1.3 :- LOGICAL EQUIVALENCE AND LAWS OF LOGIC.

* Laws of Logical Equivalence :- [Properties].

1. $p \rightarrow q \equiv \sim q \rightarrow \sim p$
2. $p \rightarrow q \equiv \sim p \vee q$
3. $\sim p \rightarrow q \equiv p \vee q$
4. $\sim(p \rightarrow q) \equiv p \wedge \sim q$
5. $p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$
6. $p \leftrightarrow q \equiv (p \wedge q) \vee (\sim p \wedge \sim q)$.

* Laws of Logic / Logical Rules :-

1. Idempotent Laws :-

1. $p \vee p \equiv p$.
2. $p \wedge p \equiv p$.

2. Commutative Laws :-

$p \wedge q \equiv q \wedge p$

$p \vee q \equiv q \vee p$.

3. Absorption Laws :-

$p \wedge (p \vee q) \equiv p$.

$p \vee (p \wedge q) \equiv p$.

4. Associative Laws :-

$p \wedge (q \wedge r) \equiv (p \wedge q) \wedge r \equiv p \wedge q \wedge r$.

$p \vee (q \vee r) \equiv (p \vee q) \vee r \equiv p \vee q \vee r$.

⑤ Distributive Law :-

$$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r).$$

$$p \vee (q \wedge r) = (p \vee q) \wedge (p \vee r).$$

⑥ Double Negation property :-

$$\sim(\sim p) \equiv p.$$

$$\sim\sim p \equiv p.$$

⑦ DeMorgan laws :-

$$\sim(p \vee q) \equiv \sim p \wedge \sim q.$$

$$\sim(p \wedge q) = \sim p \vee \sim q.$$

⑧ Inverse Law :-

$p \wedge \sim p \equiv F_0$. [Contradiction]

$p \vee \sim p \equiv T_0$. [Tautology].

⑨ Identity Law :-

$$p \wedge T_0 \equiv p.$$

$$p \vee F_0 \equiv p.$$

⑩ Domination Laws :-

$$p \vee T_0 \equiv T_0.$$

$$p \wedge F_0 \equiv F_0.$$

Note :-

$$p \rightarrow (q \rightarrow r) \equiv (p \wedge q) \rightarrow r.$$

* Replacement Rule :-

① Let 'a' be formula $(p \to q) \to r$. $[p \to q \equiv \sim p \vee q]$

Let 'a_1' be obtained by replacing $p \to q$ by $\sim p \vee q$.

$a_1 : (\sim p \vee q) \to r$.

$\therefore a \equiv a_1$

$\Rightarrow (p \to q) \to r \equiv (\sim p \vee q) \to r$.

② Let 'b' be formula $p \to (p \vee q)$. $[\because p \equiv \sim(\sim p)]$

$b_1 : p \to (\sim(\sim p) \vee q)$.

$\therefore$ By Replacement rule,

$b \equiv b_1$

$\Rightarrow p \to (p \vee q) \equiv p \to (\sim(\sim p) \vee q)$

* Methods to prove Logical Equivalent of two Formulas :-

Method-1 :-

→ To prove 'a' is logically equal to 'b', it is enough to prove that all the truth values of a and b are same for each possible assignment of Truthvalues of propositional variables involving in the formulas 'a' and 'b'.

Method-2 :-

→ $a \equiv b$, if and only if $a \leftrightarrow b$ is 'tautology'.

Method-3 :-

→ To prove 'a' is logically equal to 'b', by using 'the Laws of Logic'.

* Logical Equivalence :-

→ Let 'a' and 'b' be two wffs and let $P_1, P_2 --- P_n$ be all propositional variables occuring in a & b. Then a and b are equivalent [logically equivalent] if truth values of 'a' & 'b' are equal of '2^n' possible combinations of truth values assigned to $P_1, P_2 --- P_n$

* Theorems :-

① 'a' and 'b' are equivalent if and only if 'a ⇔ b' is a 'tautology'

* Note :-

→ We represent equivalence of a and b by 'a ⇔ b' or 'a ≡ b'. [read as a is equivalent to b]

→ The symbol '⇔' or '≡' is related to Meta language and it is a relation.

→ a ≡ b and b ≡ a are same [a, b have same Truth values].

→ If a ≡ b and b ≡ c then a ≡ c. [Transitive relation]

* Dual of a Formula :-

→ The dual of 'a', denoted by 'a^*', is formula obtained from a by replacing '∧' and '∨' by '∨' and '∧' and also 'T_0' and 'F_0' by 'F_0' and 'T_0'.

Ex :- ① $P \vee \sim P$ Dual :- $P \wedge \sim P$

② $P \vee T_0$ Dual :- $P \wedge F_0$.

③ $(P \vee \sim q)$ Dual :- $(P \wedge \sim q)$

④ $(P \vee \sim q) \wedge (r \vee F_0)$ Dual :- $(P \wedge \sim q) \vee (r \wedge T_0)$.

Note :- $\boxed{(a^*)^* = a}$

* Principle of Duality :-

→ If $a \equiv b$ then $a^* \equiv b^*$.

17-03-2023 Module - 1.4 :- Normal Forms

* **Decision Problem** :-

→ It is a problem that determines the given formula is 'Tautology' (or) 'contradiction' (or) 'satisfiable.' in a finite no of steps.

* **Literals** :- P, ~P, q, ~q.

* **Term (or) Elementary product** :-

→ A Literal (or) conjunction [product] of two or more Literals is called a 'Term' or 'Elementary product'

Ex:-

① Let P, q are two propositional variables

Literals are P, ~P, q, ~q.

Term	size [No. of Literals]
P	1
~p	1
q	1
~q	1
P∧P	2
P∧~p	2
p∧q	2
p∧~q	2
P∧~p∧q	3
p∧q∧~q	3
P∧q∧~P∧~q	4

→ We can write Terms for three Literals or three propositional variables also.

* clause or Elementary sum:-

→ A Literal (or) Disjunction (sum) of 2 or more Literals is called a clause.

Ex:-

clause	size.
p	1
$\sim p$	1
q	1
$\sim q$	1
$p \vee q$	2
$p \vee \sim q$	2
$p \vee \sim p \vee q$	3

Note:-

→ When a term contains variable and Negation of same variable then it is a contradiction.

Ex:- $(p \wedge \sim p) \wedge q$

$F_0 \wedge q = F_0$

→ When a clause contains propositional variable & Negation of that then it is a Tautology.

Ex:- $(p \vee \sim p) \vee q$ $\quad T_0$

* Normal Forms :-

→ These are two types:-

① Disjunctive Normal form [DNF]

② conjunctive Normal form [CNF].

① Disjunctive Normal Form [DNF]:-

→ Let 'a' be a formula.

A formula which is equivalent to 'a' and which consist of a sum of elementary product is called a 'DNF' of 'a'.

∴ DNF = A term (or) sum ($\vee$) of 2 or more products.

structure :-

$(l_1 \wedge l_2 \wedge l_3) \vee (l_4 \wedge l_5 \wedge l_6) \vee (l_7 \wedge l_8 \wedge l_9)$.

procedure to obtain DNF of given formula 'a'!

step-1 :- Replace $\rightarrow$, $\leftrightarrow$ with/ by using logical connectives $\sim, \wedge, \vee$.

step-2 :- Apply Demorgans Laws & Double Negation Laws if necessary.

step-3 :- Apply Distributive Laws repeatedly until we get sum of products (Terms).

$$\boxed{p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)}$$

② conjunctive Normal Form [CNF] :-

structure :-

$(l_1 \vee l_2 \vee l_3) \wedge (l_4 \vee l_5 \vee l_6) \wedge (l_7 \vee l_8 \vee l_9)$.

clauses

Definition :-

→ Product of ~~sum~~ clauses (or) product of sums is called 'CNF'.

→ The CNF of given formula is defined as.

CNF ≡ a clause (or) product(n) of 2 or more sums [clauses]

Procedure :-

step-1 & 2 :- same

step-3 :- Apply Distributive Laws repeatedly until we get product of sums [clauses]

$$\boxed{p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)}$$

* Note :-

→ Let 'a' is a formula, if its DNF is contradiction [Every term in it is contradiction] then a is contradiction.

→ $a \equiv DNF$ F_0

$a \equiv CNF$ T_0.

* Examples of DNF & CNF :-

① $p \to q \equiv \sim p \vee q$ → DNF

→ CNF

→ some formulas have both CNF & DNF.

② $p \leftrightarrow q \equiv (p \wedge q) \vee (\sim p \wedge \sim q)$ → DNF

$\underbrace{(p \wedge q)}_{\text{Term}} \vee \underbrace{(\sim p \wedge \sim q)}_{\text{Term}}$

(sum of two terms) → DNF

③ $p \leftrightarrow q \equiv (\sim p \vee q) \wedge (p \vee \sim q)$ → CNF

$\underbrace{(\sim p \vee q)}_{\text{clause}} \wedge \underbrace{(p \vee \sim q)}_{\text{clause}}$

(sum of 2 clauses) → CNF.

Module - 1.5

* Minterms :-

→ Minterm is a conjunction (product) of literals of different variables. in which each literal occurs once. with the condition that the size of that conjunction is same as no. of atomic variables.

* General structure :-

→ Let $P_1, P_2, \ldots P_n$ are 'n' atomic variables then minterm is of the following form.

$$\boxed{K_1 \wedge K_2 \wedge K_3 \wedge \ldots \ldots \wedge K_n.}$$

∴ size = 'n'. [No of variables]

Here 'K_i' is the either 'P_i' or '$\sim P_i$'. i.e Literal of corresponding variable P_i.

For $K_1 \rightarrow P_1, \sim P_1$

$K_2 \rightarrow P_2, \sim P_2$

$K_3 \rightarrow P_3, \sim P_3$

$\vdots$

$K_n \rightarrow P_n, \sim P_n$

→ We will get '2^n' minterms of n variables.

Case-i :-

→ If $n=1$ [only one variable say P].

$2^1 = 2$. [Two minterms are possible].

K_1

∴ $P, \sim P$ are minterms.

case-ii :-

→ If $n=2$ [two atomic variables]

Let P, q are two propositional variables.

$K_1 \wedge K_2$

∴ $2^2 = 4$

∴ $P, \sim P, q, \sim q$ are minterms.

variable	Literal
① P	$\left.\begin{matrix} P \\ \sim P \end{matrix}\right\} K_1$
② q	$\left.\begin{matrix} q \\ \sim q \end{matrix}\right\} K_2$

K_1	K_2	Minterm	size
p — p	q	$p \wedge q$	2
	$\sim q$	$p \wedge \sim q$	2
p — $\sim p$	q	$\sim p \wedge q$	2
	$\sim q$	$\sim p \wedge \sim q$	2

→ Every Minterm is a Term (or) Elementary product.

→ Every product [Term] is need not be a Minterm

case-iii :

→ If $n = 3$.

Let p, q, r are three propositional variables.

structure is $k_1 \wedge k_2 \wedge k_3$.

No. of Minterms $= 2^3 = \underline{8}$.

K_1	K_2	K_3	Minterms	size
p — p	q	r	$p \wedge q \wedge r$	3
		$\sim r$	$p \wedge q \wedge \sim r$	
	$\sim q$	r	$p \wedge \sim q \wedge r$	
		$\sim r$	$p \wedge \sim q \wedge \sim r$	
p — $\sim p$	q	r	$\sim p \wedge q \wedge r$	
		$\sim r$	$\sim p \wedge q \wedge \sim r$	
	$\sim q$	r	$\sim p \wedge \sim q \wedge r$	
		$\sim r$	$\sim p \wedge \sim q \wedge \sim r$.	

* Maxterms :

→ Max terms are Disjunctions of Literals of different variables in which each literal occurs once and size is equal to no of variables

* General structure :

$K_1 \vee K_2 \vee K_3 \vee \text{———} \vee K_n$

case - i :

K_1 :- $p, \sim p$ $[2^1]$.

Case-ii:

K_1	K_2	Maxterm	size
p	q	$p \vee q$	2.
	$\sim q$	$p \vee \sim q$	
$\sim p$	q	$\sim p \vee q$	
	$\sim q$	$\sim p \vee \sim q$	

$\rightarrow 2^2 = 4.$

case-iii:

K_1	K_2	K_3	Maxterms	size.
p	q	r	$p \vee q \vee r$	3.
		$\sim r$	$p \vee q \vee \sim r$	
	$\sim q$	r	$p \vee \sim q \vee r$	
		$\sim r$	$p \vee \sim q \vee \sim r$	
$\sim p$	q	r	$\sim p \vee q \vee r$	
		$\sim r$	$\sim p \vee q \vee \sim r$	
	$\sim q$	r	$\sim p \vee \sim q \vee r$	
		$\sim r$	$\sim p \vee \sim q \vee \sim r$	

$\rightarrow 2^3 = 8.$

* **Notations of Minterm & Maxterms:-**

$\rightarrow$ Minterm representation.

$m_0, m_1, m_2, m_3, \text{-----} m_{2^n - 1}$

$\rightarrow$ Maxterm representation.

$M_0, M_1, M_2, M_3, \text{-----} M_{2^n - 1}$

* case-i :- For Minterms :-

* Binary digit of b_i :-

Binary digit of b_i		Literal of corresponding variable P_i in Minterms.
0	$\longrightarrow$	$\sim P_i$ [-ve literal]
1	$\longrightarrow$	P_i (+ve literal).

* case-ii :- For Maxterms :-

Binary digit of b_i		Literal of corresponding variable P_i in Maxterms.
0	$\longrightarrow$	P_i [+ve literal]
1	$\longrightarrow$	$\sim P_i$ [-ve liberal]

* **Examples:-**

→ $P_1, P_2, \text{———} P_6$.

"$2^6 = 64$"

Minterms. are $m_0, m_1, m_2, \text{---} m_{63}$.

Maxterms are $M_0, M_1, M_2, \text{———} M_{63}$.

Ex:- "m_{28}"

Binary string for 28 is "011100".

$$
\begin{array}{r|l}
2 & 28 \\
2 & 14 - 0 \\
2 & 7 - 0 \\
2 & 3 - 1 \\
2 & 1 - 1 \\
 & 0 - 1
\end{array}
$$

First way:- $\overrightarrow{011100}$

Minterms:- $\sim P_1 \wedge P_2 \wedge P_3 \wedge P_4 \wedge \sim P_5 \wedge \sim P_6$.

Maxterms:- $P_1 \vee \sim P_2 \vee \sim P_3 \vee \sim P_4 \vee P_5 \vee P_6$.

} for 28

Second way:- [Binary string to minterm / Maxterm].

Ex:- $P_1 \wedge P_2 \wedge \sim P_3 \wedge P_4 \wedge \sim P_5 \wedge P_6$.

1 1 0 1 0 1 → Minterms.

$$
\begin{array}{cccccc}
1 & 1 & 0 & 1 & 0 & 1 \\
2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0
\end{array}
$$

$\Rightarrow 1\times2^5 + 1\times2^4 + 1\times2^2 + 1\times2^0$

$\Rightarrow 32 + 16 + 4 + 1$

$\Rightarrow 53$.

∴ It is "m_{53}".

* **case(i):-** $n=2$.

Let p, q.

→ No. of Minterms = No. of Maxterms $= 2^2 = 4$.

Minterms	Maxterms
$m_3 = p \wedge q$	$M_0 = p \vee q$
$m_2 = p \wedge \sim q$	$M_1 = p \vee \sim q$
$m_1 = \sim p \wedge q$	$M_2 = \sim p \vee q$
$m_0 = \sim p \wedge \sim q$	$M_3 = \sim p \vee \sim q$

case-ii :- for n=3.

→ Let p, q, r are given atomic variables then

No. of minterms = No. of maxterms = $2^3 = 8$.

Minterms	Maxterms
$m_7 = p \wedge q \wedge r$	$p \vee q \vee r = M_0$
$m_6 = p \wedge q \wedge \sim r$	$p \vee q \vee \sim r = M_1$
$m_5 = p \wedge \sim q \wedge r$	$p \vee \sim q \vee r = M_2$
$m_4 = p \wedge \sim q \wedge \sim r$	$p \vee \sim q \vee \sim r = M_3$
$m_3 = \sim p \wedge q \wedge r$	$\sim p \vee q \vee r = M_4$
$m_2 = \sim p \wedge q \wedge \sim r$	$\sim p \vee q \vee \sim r = M_5$
$m_1 = \sim p \wedge \sim q \wedge r$	$\sim p \vee \sim q \vee r = M_6$
$m_0 = \sim p \wedge \sim q \wedge \sim r$	$\sim p \vee \sim q \vee \sim r = M_7$.

→ Notation for minterms :-

$m_0, m_1, m_2, \text{———}, m_7$

$2\,\underline{|\,0}$
$\;\;\underline{|\,0}-0$

$2\,\underline{|\,1}$
$\;\;\underline{|\,0}-1$

		Min	Max
m_0 →	000 →	$\sim p \wedge \sim q \wedge \sim r$ →	$p \vee q \vee r$
m_1 →	001 →	$\sim p \wedge \sim q \wedge r$ →	$p \vee q \vee \sim r$.

* Note points :-

① $\boxed{\sim m_i = M_i}$ [Negation of minterm = Corresponding maxterm]

Ex:- $\sim m_4 = \sim(p \wedge \sim q \wedge \sim r)$

$= \sim p \vee q \vee r$

$= M_4$.

② $\boxed{\sim M_i = m_i}$ $\forall$ 'i'.

③ Dual of minterm is $\boxed{m_j^* = M_{(2^n-1)-j}}$ $\forall j$

Ex:- $m_4^* = p \vee \sim q \vee \sim r$

$= M_{7-4}$

$= M_3$.

④ Dual of maxterm is $\boxed{M_j^* = m_{(2^n-1)-j}}$ $\forall j$

Ex:- $M_7^* = m_{7-7} = m_0$.

* Truth Table for Minterms :-

- a = sum of all minterms.

i.e., "$a = m_0 \vee m_1 \vee m_2 \vee m_3$."

For $n=2$, $[p, q]$:-

p	q	$\sim p$	$\sim q$	$\sim p \wedge \sim q$ m_0	$\sim p \wedge q$ m_1	$p \wedge \sim q$ m_2	$p \wedge q$ m_3	a	$m_0 \vee m_1$	$m_0 \vee m_1 \vee m_2$
T	T	F	F	F	F	F	$\boxed{T}$	T	F	F
T	F	F	T	F	F	$\boxed{F}$	F	T	F	T
F	T	T	F	F	$\boxed{T}$	F	F	T	T	T
F	F	T	T	$\boxed{T}$	F	F	F	T	T	T

combinations		minterms.
c_1 : $\overset{1}{T}\ \overset{1}{T}$	$\longrightarrow$	$m_3 = p\wedge q$
c_2 : $\overset{1}{T}\ \overset{0}{F}$	$\longrightarrow$	$m_2 = p\wedge\sim q$
c_3 : $\overset{0}{F}\ \overset{1}{T}$	$\longrightarrow$	$m_1 = \sim p\wedge q$
c_4 : $\overset{0}{F}\ \overset{0}{F}$	$\longrightarrow$	$m_0 = \sim p\wedge\sim q$.

* Canarical Normal Forms:-

→ There are two canarical Normal Forms.

1. PDNF [Principle Disjunctive Normal Form].
2. PCNF [Principle conjunctive Normal form].

① **PDNF**:- [Principle Disjunctive Normal Form]

(or)

[sum of products of cananical Normal Form].

→ A Minterm alone (or) Disjunction of minterms alone is a 'PDNF'.

(or)

→ for a given formula an equivalent formula which is of the form a minterm (or) Disjunction of Minterms only is called 'PDNF' of given formula.

General structure:-

→ Let 'a' be a formula then PDNF of 'a' is as follows:-

$a \equiv (\text{minterm}1) \vee (\text{minterm}2) \vee \_\_\_\_ \vee (\text{minterm}n)$.

Note:-

→ A formula is contains Disjunction of Minterms. need not all minterms present.

* compact form of PDNF :-

→ Let 'a' be a formula and PDNF of 'a' is as follows:-

"$a \equiv m_i \vee m_j \vee m_k$".

Here m_i, m_j, m_k are 'minterms'.

$a \equiv m_i \vee m_j \vee m_k$

$\Rightarrow \boxed{a \equiv \sum i, j, k.}$

This is the compact form of PDNF.

Ex:-

① $P \leftrightarrow q \equiv (P \wedge q) \vee (\sim p \wedge \sim q)$

$\equiv m_3 \vee m_0.$

$\equiv \sum 0, 3.$

$\therefore$ It is PDNF contact form.

Note:-

→ Every PDNF is a DNF.

→ Every CPDNF is a CNF.

① Since Every minterm is a term, so every PDNF is also a DNF.

② Since For any formula, the PDNF is always unique.

② PCNF [Principle conjunctive Normal Form]:-

(or)

[Product of sums cannonical Normal Form].

→ For a given formula, an equivalent formula which is of the form a Maxterm (or) product of Maxterms only is called PCNF of given formula.

* General structure of PCNF:-

→ Let 'a' be a given formula then PCNF of 'a' is also follows:-

$a \equiv$ (Maxterm alone) $\wedge$ (Maxterm2) $\wedge$ ____ $\wedge$ (Maxterm n)

*compact form of PCNF:-

→ Let 'a' be a given formula and the PCNF of 'a' is as follows:-

$a = M_k \wedge M_l \wedge M_m \wedge M_n.$

$\Rightarrow \boxed{a \equiv \Pi_{k,l,m,n}}$

This is the compact form of PCNF.

Ex:-

① $p \leftrightarrow q \equiv (\sim p \vee q) \wedge (p \vee \sim q)$.

$\equiv M_2 \wedge M_1$

$\equiv \Pi_{1,2}$

$p \vee q$	M_0
$p \vee \sim q$	M_1
$\sim p \vee q$	M_2
$\sim p \vee \sim q$	M_3.

Note points :-

① For a given formula, the 'PCNF' is always unique.

② For two (different) formulas have same PCNF then those two formulas are 'logically equivalent'.

③ $\sim(\Sigma_{i,j,k}) \equiv \sim(m_i \vee m_j \vee m_k)$.

$\equiv \sim m_i \wedge \sim m_j \wedge m_k$ [∵ De morgans].

$\equiv M_i \wedge M_j \wedge M_k$ [∵ $\sim m_i = M_i$].

$\equiv \Pi_{i,j,k}$.

∴ $\boxed{\sim(\Sigma_{i,j,k}) \equiv \Pi_{i,j,k}}$

④ $\boxed{\sim(\Pi_{i,j,k}) = \Sigma_{i,j,k}}$

Methods to find PDNF and PCNF :-

There are three methods to find PDNF and PCNF.

Method 1 :- using Truth Table :-

Case (i) :- Finding PDNF :-

Step 1 :- Construct a truth Table of the given formula (wff) Say "a".

Step 2 :- For every truth value "T" of the given formula "a", select the "minterm", which also has the truth value "T" for the same combination of the truth values of the propositional variables.

Step 3 :- The disjunctive of minterms selected in step (2) is the required "PDNF" of given formula "a"

Case (ii) :- Finding PCNF

Step 1 :- Construct a truth table of the given formula (wff), Say "a"

Step 2 :- For every truth value "F" of the given formula "a", select the "Maxterms", which also has the truth value "F" for the same combination of the truth values of the propositional variables

Step 3 :- The conjunction of Maxterms selected in step (2) is the required "PCNF" of given formula "a".

Method 2:- By Using laws of logic (or) without constructing the Truth table

Case (i):- Finding "PDNF"

Step 1:- Obtain the DNF of a given formula (wff)

Step 2:- ~~Drops~~ Drop elementary products (i.e terms) which are contradictions. (That is, Drop the elementary product contains a variable and negation of same variable)

Step 3:- Apply Idempotent law to elementary products which are contain a variable occurs more than once

~~Note~~ Note:- Apply commutative law (if necessary)

Step 4:- After that we can get minterms in disjunctions with the help of introducing the missing factors.

~~That is~~ If P_i and $\sim P_i$ are missing in an elementary product "α", replace α by $(\alpha \wedge P_i) \vee (\alpha \wedge \sim P_i)$

Let "α" is not minterm but it is a term factors.

Let "α" is not maxterm, but it is an elementary sum.

If P_i and $\sim P_i$ are missing in this elementary sum "α", replace "α" by $(\alpha \vee P_i) \wedge (\alpha \vee \sim P_i)$

That is $\alpha \equiv \alpha \vee F_0$

$\equiv \alpha \vee (P_i \wedge \sim P_i)$ (Express F_0 as in missing variables)

$\alpha \equiv (\alpha \vee P_i) \wedge (\alpha \vee \sim P_i)$

In short cut,

$\alpha \equiv$ product of all Maxterms which are containing "α".

Step 5:- Duplication of Maxterms are avoided by using Idempotent law.

~~Note~~ Ex 5 $\alpha \equiv \alpha \wedge T_0$

$\equiv \alpha \wedge (P_i \vee \sim P_i)$ (here write down missing variables) T_0 is express as

$\equiv (\alpha \wedge P_i) \vee (\alpha \wedge \sim P_i)$

~~≡ i.e write down all minterms,~~ ~~which are~~ ~~contain~~

That is ... (In short cut)

$\alpha \equiv$ write down sum of all minterms which are contain "α".

Step 5:- Duplication of minterms are avoided, by using Idempotent law.

Case (ii) :- Finding PCNF :-

Step 1:- Obtain the CNF of a given formula (wff)

Step 2:- Drop elementary sums (i.e clauses) which are tautologies (That is Drop elementary sums contain a variable and negation of same variable)

Step 3:- Apply Idempotent law to elementary sums which are contain a variable occurs more than once

Note:- apply commutative law (if necessary)

Step 4:- After that we can get maxterms in conjunctions with the help of introducing missing

Let "a" be a formula (wff)

Case (i) :- finding PDNF of "a" when its PCNF is known

Step 1:- If the PCNF of a given formula "a" containing 'n' variables is known, then PCNF of "$\sim a$" is the product (conjunction) of the remaining Maxterms which donot appear the PCNF of "a".

Step 2:- Since $a \equiv \sim(\sim a)$, the PDNF of "a" can be obtained by applying DeMorgan laws to PCNF of "$\sim a$".

i.e PDNF of $a \equiv \sim$(PCNF of $\sim a$)

Case (2) :- Finding PCNF of "a" when PDNF of "a" is Known :-

Step 1:- If the PDNF of a given formula (wff) "a" Containing n variables is Known, then PDNF of "$\sim a$" is the Sum (Disjunction) of the remaining minterms which donot appear in the PDNF of "a"

Step 2:- Since $a \equiv \sim(\sim a)$, the PCNF of "a" can be obtained by applying DeMorgan laws to PDNF of "$\sim a$"

i.e. ~~PDNF~~

PCNF of $a \equiv \sim$ (PDNF of "$\sim a$")