Unit-3

Logic Concepts

Logic was considered to be branch of philosphy, since the middle of the 19th century, formal logic has been studied in the context of foundations of mathematics, where it was often referred to as Symbolic Logic. Logic helps in investigating and classifying the structure of statements and arguments through the study of formal systems of inference.

Symbolic Logic is often divided into two branches, namely

1. Propositional Logic
2. Predicate Logic

Proposition:- It is a declarative statement that is either true or false (but not both) in a given context.

The field of logic is concerned with Validity, Consistency and Inconsisten

Logical Systems should possess properties such as Consistency, Soundness, Completeness.

Consistency:- Consistency implies that none of the theorems of the system should contradict each other.

Soundness :- Soundness means that the inference rules shall never allow a false inference from two premises. If a system is sound and its axioms are true then its theorems are also guaranteed to be true.

Completeness:- Completeness means that there are no true sentences in the system that cannot be proved in the system.

Propositional Calculus:- Propositional Calculus (PC) refers to a languag of propositions in which a set of rules are used to combine simple propositions to form compound propositions with the help of certain logical operators. These logical operators are often called Connectives.

Well Formed Formula:- A well formed formula is defined as a symbol or a string of symbols generated by the formal grammar of a formal language. The following are the important properties of a well formed formula in propositional calculus:

- The smallest unit is considered to be a well formed formula
- If α is a well formed formula then $\sim\alpha$ is also well formed formula.
- If α and β are well formed formulae, then $\alpha \wedge \beta$, $\alpha \vee \beta$, $\alpha \rightarrow \beta$, $\alpha \leftrightarrow \beta$ are also well formed formulae.

Truth Table:- In PC, a truth table is used to provide operational definitions of important logical operators; it elaborates all possible truth values of a formula. The logical constants in Propositional calculus are true and false and are represented as T and F

Truth Table for Logical Operators

A	B	~A	A ∨ B	A ∧ B	A → B	A ↔ B
T	T	F	T	T	T	T
T	F	F	T	F	F	F
F	T	T	T	F	T	F
F	F	T	F	F	T	T

The truth values of well formulae are calculated by using the truth table.

Equivalence:- Two formulae α and β are said to be logically equivalent ($\alpha \cong \beta$) if and only if the truth values of both are the same for all possible assignments of logical constants to the symbols appearing in the formulae.

Equivalence Laws:- Equivalence laws are used to reduce or simplify a given well formed formula or to derive a new formula from the existing formula.

Name of Relation	Equivalence Relation
Commutative Law	$A \vee B \cong B \vee A$ $A \wedge B \cong B \wedge A$
Associative Law	$A \vee (B \vee C) \cong (A \vee B) \vee C$ $A \wedge (B \wedge C) \cong (A \wedge B) \wedge C$

Name of the Relation	Equivalence Relation
Double Negation	$\sim(\sim A) \cong A$
Distributive Laws	$A \vee (B \wedge C) \cong (A \vee B) \wedge (A \vee C)$ $A \wedge (B \vee C) \cong (A \wedge B) \vee (A \wedge C)$
DeMorgan's Laws	$\sim(A \vee B) \cong \sim A \wedge \sim B$ $\sim(A \wedge B) \cong \sim A \vee \sim B$
Absorption Laws	$A \vee (A \wedge B) \cong A$ $A \vee (A \vee B) \cong A$ $A \vee (\sim A \wedge B) \cong A \vee B$ $A \wedge (\sim A \vee B) \cong A \wedge B$
Idempotent	$A \vee A \cong A$ $A \wedge A \cong A$
Excluded Middle Law	$A \vee \sim A \cong T$
Contradiction Law	$A \wedge \sim A \cong F$
Commonly used equivalence relations	$A \vee F \cong A$ $A \vee T \cong T$ $A \wedge T \cong A$ $A \wedge F \cong F$ $A \rightarrow B \cong \sim A \vee B$ $A \leftrightarrow B \cong (A \rightarrow B) \wedge (B \rightarrow A)$

Let us consider Absorption Law $A \vee (A \wedge B) \cong A$ using truth table

A	B	$A \wedge B$	$A \vee (A \wedge B)$
T	T	T	T
T	F	F	T
F	T	F	F
F	F	F	F

The values of $A \vee (A \wedge B)$ and A are same; therefore these expressions are equivalent.

<u>Propositional Logic</u>:- Propositional logic (prop logic) deals with the validity, satisfiability (Consistency) and unsatisfiability (Inconsistency) of a formula and deals with derivation of a new formula using equivalence laws.

Each row of a truth table for a given formula α is called Interpretation. A formula α is said to be a tautology if and only if the value of α is true for all its interpretations.

- A formula α is said to be valid if and only if it is a tautology.

- A formula α is said to be satisfiable if there exists atleast one interpretation for which α is true

• A formula α is said to be unsatisfiable if the value of α is false under all interpretations.

Example:- Show that the following is a valid argument

If it is humid then it will rain and since it is humid today it will rain

A : It is humid

B: It will rain

$\alpha : ((A \rightarrow B) \wedge A) \rightarrow B$

A	B	$A \rightarrow B$	$(A \rightarrow B) \wedge A$	$((A \rightarrow B) \wedge A) \rightarrow B$
T	T	T	T	T
T	F	F	F	T
F	T	T	F	T
F	F	T	F	T

The given statement is valid.

Truth Table Approach is a simple and straight-forward method and is extremely useful at presenting an overview of truth values in a given situation.

For example, we have to show $\alpha : (A \wedge B \wedge C \wedge D) \rightarrow (B \vee E)$ is valid using truth table approach, we need to construct table with 32 interpretations.

We know that this statement is true for only 2 entries, it is clear that α is true for 30 entries where $(A \wedge B \wedge C \wedge D)$ is false. Hence we are left to verify whether α is true or not for 2 entries only by checking an expression on the right side of $\rightarrow$.

Use of truth table approach in such situations proves to be a wastage of time. Therefore, we require some other methods which can help in proving the validity of the formula directly. Methods are:-

1. Natural Deduction System
2. Axiomatic System
3. Semantic Tableau Method
4. Resolution Refutation method.

<u>Natural Deduction System</u>:- Natural Deduction System (NDS) is thus called because of the fact that it mimics the pattern of the natural reasoning. This system is based on set of deductive inference rules. Assuming that $A_1 \ldots A_K$ where $1 \leq K \leq n$ are set of atoms and α_j where $1 \leq j \leq m$ and β are well formed formulae.

Rule Name	Symbol	Rule	Description
Introducing $\wedge$	$I : \wedge$	If $A_1, A_2 \ldots A_n$ then $A_1 \wedge A_2 \wedge \ldots A_n$	If $A_1 \ldots A_n$ are true, then their conjunction of $A_1 \wedge A_2 \ldots \wedge A_n$ is also true.
Eliminating $\wedge$	$E : \wedge$	If $A_1 \wedge A_2 \ldots \wedge A_n$ then $A_i (1 \leq i \leq n)$	If $A_1 \wedge A_2 \ldots \wedge A_n$ is true, then any A_i is also true
Introducing $\vee$	$I : \vee$	If any $A_i (1 \leq i \leq n)$ then $A_1 \vee A_2 \ldots \vee A_n$	If any $A_i (1 \leq i \leq n)$ is true, the $A_1 \vee A_2 \vee \ldots \vee A_n$ is also true
Eliminating $\vee$	$E : \vee$	If $A_1 \vee A_2 \ldots \vee A_n, A_1 \rightarrow A, \ldots A_n \rightarrow A$, then A	If $A_1 \vee A_2 \ldots \vee A_n, A_1 \rightarrow A, A_2 \rightarrow A \ldots A_n \rightarrow A$ is true, then A is true
Introducing $\rightarrow$	$I : \rightarrow$	If from $\alpha_1, \alpha_2 \ldots \alpha_n$ ~~infer~~ β is proved, then $\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n \rightarrow \beta$ is proved	If given that $\alpha_1, \alpha_2 \ldots \alpha_n$ are tr and from these we deduce β th $\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n \rightarrow \beta$ is also true
Eliminating $\rightarrow$	$E : \rightarrow$	If $A_1 \rightarrow A, A_1$ then A	If $A_1 \rightarrow A$ and A are true then is true. This is called Modus Ponen rule.
Introducing $\leftrightarrow$	$I : \leftrightarrow$	If $A_1 \rightarrow A_2, A_2 \rightarrow A_1$ then $A_1 \leftrightarrow A_2$	If $A_1 \rightarrow A_2$ and $A_2 \rightarrow A_1$ is also tr then $A_1 \leftrightarrow A_2$ is also true
Eliminating $\leftrightarrow$	$E : \leftrightarrow$	If $A_1 \leftrightarrow A_2$ then $A_1 \rightarrow A_2, A_2 \rightarrow A_1$	If $A_1 \leftrightarrow A_2$ is true then $A_1 \rightarrow A_2, A_2 \rightarrow A_1$ is also true

Rule Name	Symbol	Rule	Description
Introducing $\sim$	$I : \sim$	If from A infer $A_1 \wedge \sim A_1$ is proved then $\sim A$ is proved	If from A (which is true), a contradiction is proved then truth of $\sim A$ is also proved.
Eliminating $\sim$	$E : \sim$	If from $\sim A$ infer $A_1 \wedge \sim A_1$ is proved then A is proved	If from $\sim A$, a contradiction is proved then truth of A is also proved.

Example:- Prove that $A \wedge (B \vee C)$ is deduced from $A \wedge B$

Given $A \wedge B$

$E : \wedge$ A

$E : \wedge$ B

$I : \vee$ $B \vee C$

$I : \wedge$ $A \wedge (B \vee C)$

Prove that infer $((A \to B) \wedge (B \to C)) \to (A \to C)$

Given $A \to B$

$B \to C$

from A infer C

$E : \to$ A

$E : \to$ B

$E : \to$ C

$I ; \rightarrow \qquad A \rightarrow C$

Axiomatic System :- The Axiomatic System is based on a set of three axioms and one rule of deduction. In axiomatic system, the proofs of the theorems are often difficult and require a guess in selection of appropriate axioms.

In this system, only two logical operator NOT ($\sim$) and IMPLIES ($\rightarrow$) allowed to form a formula. It should be noted that other logical operators such as $\wedge, \vee, \longleftrightarrow$ can be easily expressed in terms of '$\sim$' and '$\rightarrow$' using equivalence laws.

Axiom 1 : $\alpha \rightarrow (\beta \rightarrow \alpha)$

Axiom 2 : $[\alpha \rightarrow (\beta \rightarrow \gamma)] \rightarrow [(\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \gamma)]$

Axiom 3 : $(\sim\alpha \rightarrow \sim\beta) \rightarrow (\beta \rightarrow \alpha)$

Modus Ponen Rule : Hypothesis : $\alpha \rightarrow \beta$ and α; Consequent : β

Interpretation of Modus Ponen Rule :- Given that $\alpha \rightarrow \beta$ and α are hypothesis, β is infered as a consequent.

Let $\Sigma = \{\alpha_1, \alpha_2 \ldots, \alpha_n\}$ be a set of hypothesis. The formula α is defined to be deductive consequence of Σ if either α is an axiom or a hypothesis or is derived from α_j where $1 \leq j \leq n$, using modus ponen rule

It is represented as $\{\alpha_1, \alpha_2 \ldots \alpha_n\} \vdash \alpha$ or more formally as $\Sigma \vdash \alpha$. "

α is deduced from axioms only and no hypothesis are used. In such situation, α is said to be theorem.

Example:- Establish that $A \to C$ is a deductive consequence of $\{A \to B, B \to C\}$

Theorem : $\{A \to B,\ B \to C\} \vdash A \to C$

Hypothesis 1 : $A \to B$ — 1

Hypothesis 2 : $B \to C$ — 2

Instance of Axiom 1 : $(B \to C) \to (A \to (B \to C))$ — 3

Modus Ponen (2, 3) : $A \to (B \to C)$ — 4

Instance of Axiom 2 : $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$ — 5

Modus Ponen (4, 5) : $(A \to B) \to (A \to C)$ — 6

Modus Ponen (1, 6) : $(A \to C)$ — 7

Deduction Theorem :- Given that Σ is a set of hypothesis and α, β are well formed formula. If β is proved from $\{\Sigma \cup \alpha\}$ according to the deduction theorem $(\alpha \to \beta)$ is proved from Σ

$$\{\Sigma \cup \alpha\} \vdash \beta \text{ implies } \Sigma \vdash (\alpha \to \beta)$$

Converse of Deduction Theorem :-

$$\Sigma \vdash (\alpha \to \beta), \text{ then } \{\Sigma \cup \alpha\} \vdash \beta$$

Semantic Tableau System:- In both natural deduction and axiomatic systems, forward chaining approach is used for the construction of proofs and derivations. Semantic Tableau and Resolution Refutation proofs follows backward chaining approach.

In Semantic Tableau method, a set of rules are applied systematical on a formula or a set of formulae in order to establish consistency or inconsistency. Semantic Tableau is a binary tree which is constructed by using semantic tableau rules with a formula as a root. These rules and building proofs using this method are discussed in detail

Semantic Tableau Rules:-

$\alpha \wedge \beta$ is true if both α and β are true —

$$\begin{array}{c} \alpha \wedge \beta \\ | \\ \alpha \\ | \\ \beta \end{array}$$

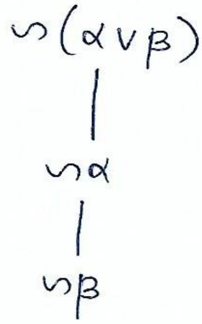
$\sim(\alpha \wedge \beta)$ is true if either $\sim\alpha$ or $\sim\beta$ is true —

$$\begin{array}{ccc} & \sim(\alpha \wedge \beta) & \\ \swarrow & & \searrow \\ \sim\alpha & & \sim\beta \end{array}$$

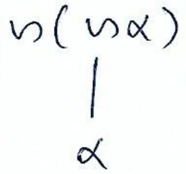
$\alpha \vee \beta$ is true if either α or β is true —

$$\begin{array}{ccc} & \alpha \vee \beta & \\ \swarrow & & \searrow \\ \alpha & & \beta \end{array}$$

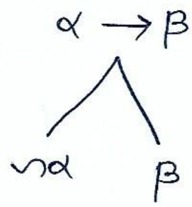
$\neg(\alpha \vee \beta)$ is true if both $\neg\alpha$ and $\neg\beta$ are true —



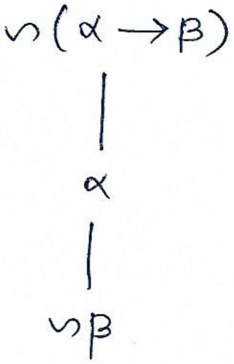
$\neg(\neg\alpha)$ is true then α is true —



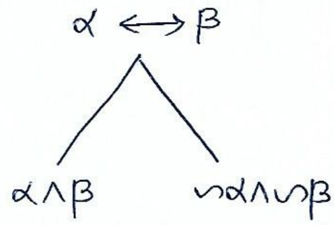
$\alpha \rightarrow \beta$ is true then $\neg\alpha \vee \beta$ is true —



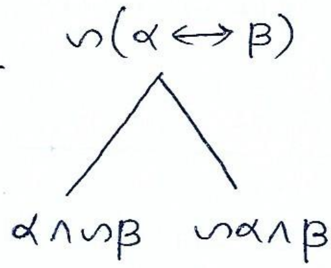
$\neg(\alpha \rightarrow \beta)$ is true then $\alpha \wedge \neg\beta$ is true —



$\alpha \leftrightarrow \beta$ true then $(\alpha \wedge \beta) \vee (\neg\alpha \wedge \neg\beta)$ is true —

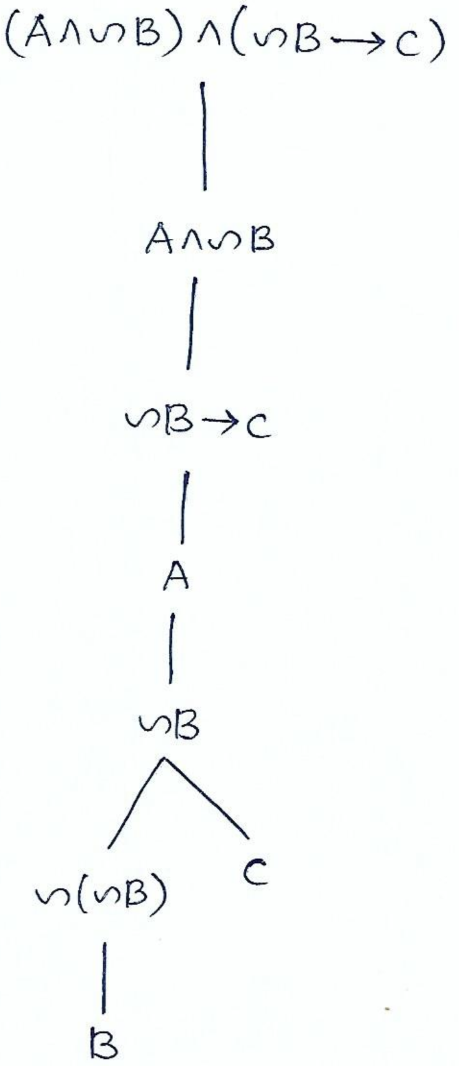


$\neg(\alpha \leftrightarrow \beta)$ is true then $(\alpha \wedge \neg\beta) \vee (\neg\alpha \wedge \beta)$ is true —



Example: Construct a semantic tableau for a formula $(A \land \neg B) \land (\neg B \to C$

Given $(A \land \neg B) \land (\neg B \to C)$



The path from root to B becomes closed because of the presence of complementary atoms $(B, \neg B)$, while the other path remain open.

- A path is said to be contradictory when the complementary atoms appear in the same path.

<u>Contradictory Tableau</u>:- If all paths of a tableau for a given formula α are found to be closed.

Satisfiable :- A formula α is said to be satisfiable, if it has atleast one open path.

Tableau Provable :- If all the paths of tableau for a given formula α are found to be open, then it is Tableau Provable.

* A set of formulae $S = \{\alpha_1, \alpha_2 .. \alpha_n\}$ is said to be satisfiable if the formula in the set has atleast one α open path.

* A set of formulae $S = \{\alpha_1, \alpha_2 .. \alpha_n\}$ is said to be unsatisfiable if a tableau is a contradictory tableau.

Resolution Refutation in Propositional Logic :- Resolution Refutation can be used in propositional logic to prove a formula or derive a goal from a given set of clauses by contradiction. The term clause is used to denote a special formula containing the boolean operators $\wedge$ and $\vee$. It uses a single inference rule, which is known as resolution based on modus ponen inference rule. Negation of the goal to be proved is added to the given set of clauses and using the resolution principle, it is shown that there is a refutation in the new set.

Conversion of a Formula into a set of clauses:- In propositional logic, there are two normal forms. They are:-

1. Disjunctive Normal Form
2. Conjunctive Normal Form.

Disjunctive Normal Form:- It is represented as disjunction of conjunction that is in form $(L_{11} \wedge L_{12} \cdots \wedge L_{1m}) \vee \cdots (L_{p1} \wedge L_{p2} \cdots \wedge L_{pk})$

Conjunctive Normal Form:- It is represented as conjunction of disjunction that is in form $(L_{11} \vee L_{12} \cdots \vee L_{1m}) \wedge (L_{21} \vee L_{2m}) \wedge (L_{p1} \wedge L_{p2} \cdots \wedge L_{pk})$

Conversion of a Formula to its CNF:- Any formula in propositional logic can be easily transformed into its equivalent CNF using the equivalence laws by following steps:

- Eliminate double negation signs by using $\sim(\sim A) \cong A$
- Use Demorgan's Laws to push $\sim$(negation) immediately before the atomic formula. $\sim(A \wedge B) \cong \sim A \vee \sim B$

  $\sim(A \vee B) \cong \sim A \wedge \sim B$

- Use distributive law to get CNF $A \vee (B \wedge C) \cong (A \vee B) \wedge (A \vee C)$

- Eliminate $\rightarrow$ and $\leftrightarrow$ by using the following equivalence laws:

$$A \rightarrow B \cong \sim A \vee B$$

$$A \leftrightarrow B \cong (A \rightarrow B) \wedge (B \rightarrow A)$$

<u>Example:</u>- Convert the formula $(\sim A \rightarrow B) \wedge (C \wedge \sim A)$ into its equivalent CNF

Given formula

$$(\sim A \rightarrow B) \wedge (C \wedge \sim A) \cong (\sim(\sim A) \vee B) \wedge (C \wedge \sim A)$$

$$(\because A \rightarrow B \cong (\sim A \vee B))$$

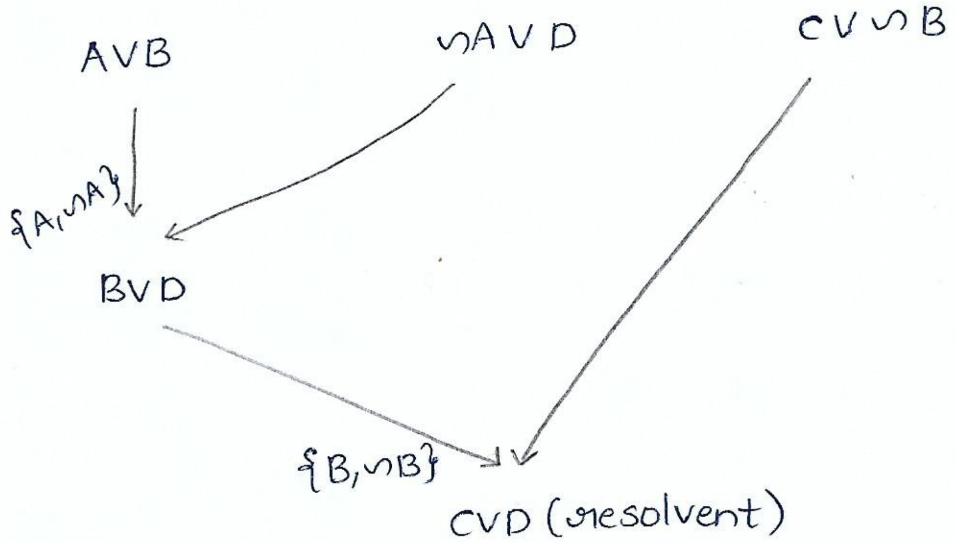
$$\cong (A \vee B) \wedge C \wedge \sim A$$

Set of clauses $= \{(A \vee B), C, \sim A\}$

<u>Resolution of Clauses:</u>- Two clauses can be resolved by eliminating complementary pair of literals. A new clause is constructed by disjunction of both the literals in both the clauses. Two clauses C_1 and C_2 contain a complementary pair of literals $\{L, \sim L\}$, then this clauses may be resolved together by deleting L from C_1 and $\sim L$ from C_2. The new clause is a resolvent of C_1 and C_2. The clauses C_1 and C_2 are parent clauses.

The resolution tree is a inverted binary tree with the last node being a resolvent, which is generated as a part of resolution process.

<u>Example</u>: Find resolvent of clauses in the set $\{A \vee B, \sim A \vee D, C \vee \sim B\}$



- If c is a resolvent of two clauses c_1 and c_2, then c is called Logical consequence of the set of clauses $\{c_1, c_2\}$. This is known as Resolution principle.
- If a contradiction is derived from a set S of clauses using resolution then S is said to be unsatisfiable. Derivation of contradiction for a set 'S' by resolution method is called Resolution Refutation of S

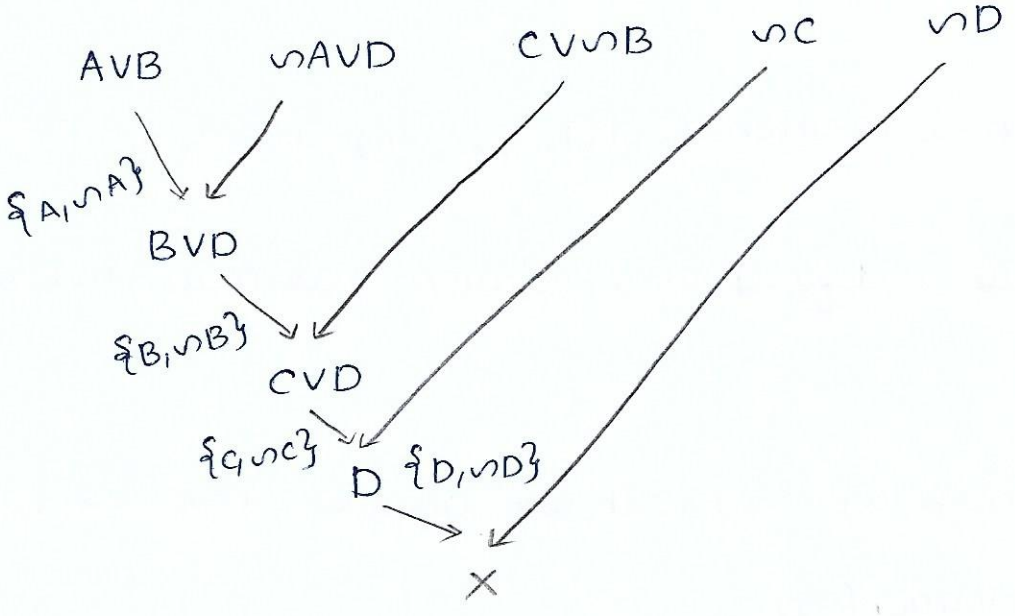
- A clause c is said to be a logical consequence of S if c is derived from S.

- Using the resolution refutation concept, a clause c is defined to be logical sequence of S if and only if the set $S' = S \cup \{\sim c\}$ is unsatisfiable, that is a contradiction is deduced from the set S', assuming that initially set S is satisfiable.

<u>Example</u>: Using resolution refutation principle show that $C \vee D$ is a logical consequence of $S = \{A \vee B, \sim A \vee D, C \vee \sim B\}$

Given $S = \{A \vee B, \sim A \vee D, C \vee \sim B\}$

$S' = \{A \vee B, \sim A \vee D, C \vee \sim B, \sim C, \sim D\}$ $\sim(C \vee D) \cong \sim C \wedge \sim D$



Hence, we get contradiction from s', we conclude that $(C \vee D)$ is a logical consequence of $S = \{A \vee B, \sim A \vee D, C \vee \sim B\}$

Predicate Logic:- Predicate logic is an extension to propositional logic.

Limitations of Propositional Logic:-

1. We cannot draw conclusions about similarities between the statements.

2. No quantifiers in propositional logic

Predicate Logic:- It has three more logical notions as compared to propositional logic.

Term:-

- a constant or a variable or a function (n place).
- A function is defined as a mapping that maps n terms to a single term
- An n-place function is written as $f(t_1, t_2, .. t_n)$ where $t_1, t_2 ..$ are term

Predicate:- A predicate is defined as a relation that maps 'n' terms to a truth value.

Quantifiers:- Quantifiers are used with ~~are~~ variables. There are two types. 1. Universal Quantifier ($\forall$)

2. Existential Quantifier ($\exists$)

First Order Predicate Calculus:- If the quantification in predicate formula is only on simple variables and not on predicates or functions then it is called First Order predicate Calculus.

Ex: $\forall x \forall y (p(x) \rightarrow p(y))$

Second Order Predicate Calculus:- If the quantification is over first order predicate and functions then it is called Second Order Predicate calculus.

Ex: $\forall p (p(x) \leftrightarrow p(y))$

First Order predicate Logic:- When inference rules are added to first order predicate calculus then it becomes First Order Predicate Logic (FOL).

Interpretations of Formulae in FOL:- In first order predicate logic (FOL), An interpretation I for a formula 'α' consists of non-empty domain D and an assignment of values to each constant, function symbol and predicate symbol.

$\forall x \ p(x) =$ true if and only if $p(x) =$ true, $\forall x \in D$, otherwise it is false

$\exists x \ p(x) =$ true if and only if $\exists \ c \in D$ that $p(c) =$ true, otherwise it is false

→ A formula 'α' is said to be satisfiable if and only if there exists an interpretation I such that $I[\alpha] = True$

→ A formula 'α' is said to be unsatisfiable if and only if there exists no interpretation that satisfies 'α'.

→ A formula 'α' is valid if and only if for every interpretation I, $I[\alpha] = True$

→ A formula 'α' is logical consequence of a set of formulae $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ if and only if for every interpretation I, if $I[\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n] = True$, then $I[\alpha] = True$

Prenex Normal Form:- A closed formula 'α' is said to be in PNF if and only if α is represented as $(Q_1 x_1)(Q_2 x_2) \ldots (Q_n x_n) M$ where

Q_K are quantifiers

x_K are variables

M is formula free from Quantifiers

Conversion into PNF:- Using equivalence laws, we can convert a formula into PNF

Law 1:- $(Qx)\alpha[x]\beta \cong (Qx)(\alpha[x]\beta)$

Law 2:- $\alpha (Qx) \beta[x] \cong (Qx)(\alpha \beta[x])$

Law 3: $\sim(\forall x)\,\alpha[x] \cong (\exists x)(\sim\alpha[x])$

Law 4: $\sim(\exists x)\,\alpha[x] \cong (\forall x)(\sim\alpha[x])$

Law 5:- $(\forall x)\,\alpha[x] \wedge (\forall x)\,\beta[x] \cong (\forall x)(\alpha[x] \wedge \beta[x])$

Law 6:- $(\exists x)\,\alpha[x] \vee (\exists x)\,\beta[x] \cong (\exists x)(\alpha[x] \vee \beta[x])$

Skolemization:- The prenex normal form of a given formula can be further transformed into a special form called Skolemization or standard form. The process of eliminating existential quantifiers from the prefix of PNF and replacing the corresponding variable by a constant or a function is called skolemization.

- A constant or a function is called skolem Constant or function

Steps for Skolemization:-

- Scan prefix from left to right till we obtain first existential quantifier
  - If Q_1 is the first existential quantifier then choose a new constant $c \notin$ {set of constants in M}. Replace all occurrences of x_1 appearing in matrix M by c and delete $(Q_1 x_1)$ from prefix to obtain a new prefix and matrix

- If Q_r is the first existential quantifier and $Q_1 \ldots Q_{r-1}$ are universal quantifiers appearing before Q_r, then choose a new $(r-1)$ place function symbol $f \notin \{$set of functions appearing in $M\}$

Replace all occurences of X_r in M by $f(X_1, \ldots X_{r-1})$ and remove $(Q_r X_r)$ from prefix

• Repeat the process till all existential quantifiers are removed from M.

<u>Clauses in FOL</u>:- Let $S = \{C_1, C_2, \ldots, C_K\}$ be a set of clauses that represents a standard form of a formula 'α'

→ 'S' is said to be unsatisfiable if and only if there exists no interpretation that satisfies all the clauses of S simultaneously.

→ A formula 'α' is unsatisfiable if and only if its corresponding set 'S' is unsatisfiable

→ 'S' is said to be satisfiable if and only if each clause is satisfiable.

<u>Resolution Refutation in Predicate Logic</u>:- Resolution Refutation method in FOL is used to test unsatisfiability of a set (s) of clauses corresponding to the predicate formula.

→ A deduction of a contradiction from a set of clauses is Resolution Refutation of S.

→ This resolution principle checks whether a contradiction is contained in or derived from S

→ Resolution for the clauses containing no variable is simple and becomes complicated when clauses contain variables. In such cases, two complementary literals are resolved after proper substitutions so that both literals have same arguments.

<u>Example</u>:- Find the resolvent of two clauses CL_1 and CL_2, where p, q and r are predicate symbols, x is a variable and f is unary function

$$CL_1 = P(x) \vee Q(x)$$

$$CL_2 = \sim P(f(x)) \vee R(x)$$

<u>Sol</u>: Given $CL_1 = P(x) \vee Q(x)$

$CL_2 = \sim P(f(x)) \vee R(x)$

Substitute $f(a)$ for x in CL_1 and 'a' for x in CL_2,

where 'a' is a new constant from the domain

$$CL_3 = P(f(a)) \vee Q(f(a))$$

$$CL_4 = \sim P(f(a)) \vee R(a)$$

Resolvent of CL_3 and CL_4 is $Q(f(a)) \vee R(a)$ because the clauses

CL_3 and CL_4 has complementary literals

$$CL' = Q(f(a) \vee R(a) \Rightarrow CL' = Q(f(x)) \vee R(x)$$

We substitute $f(x)$ for 'x' in CL_4'

$$CL' = Q(f(x)) \vee R(x)$$

We notice that CL is an instance of CL' and can be obtained

from CL' by substituting a for x.

→ L is some predicate formula and set of clauses $S = \{C_1, C_2 \dots C_n\}$

* L is a logical consequence iff $\{S \cup \sim L\} = \{C1, C_2 .., C_n, \sim L\}$ is

unsatisfiable.

* Soundness and completeness of resolution theorem states that there

is a resolution refutation of 'S' iff 'S' is unsatisfiable.

* L is a logical consequence of S iff there is resolution refutation of $S \cup \{\sim L\}$