UNIT III

Logic Concepts: Introduction, propositional calculus, proportional logic, natural deduction system, axiomatic system, semantic tableau system in proportional logic, resolution refutation in proportional logic, predicate logic

1.1.Propositional Logic Concepts:

- Logic is a study of principles used to
  - distinguish correct from incorrect reasoning.
- Formally it deals with
  - the notion of truth in an abstract sense and is concerned with the principles of valid inferencing.
- A proposition in logic is a declarative statements which are either true or false (but not both) in a given context. For example,
  - “Jack is a male”,
  - "Jack loves Mary" etc.
- Given some propositions to be true in a given context,
  - logic helps in inferencing new proposition, which is also true in the same context.
- Suppose we are given a set of propositions such as
  - “It is hot today" and
  - “If it is hot it will rain", then
  - we can infer that

“It will rain today".

1.2.Well-formed formula

- Propositional Calculus (PC) is a language of propositions basically refers
  - to set of rules used to combine the propositions to form compound propositions using logical operators often called connectives such as Λ, V, ~, →, ↔
- Well-formed formula is defined as:

- An atom is a well-formed formula.
- If α is a well-formed formula, then ~α is a well-formed formula.
- If α and β are well formed formulae, then (α Λ β), (α V β), (α → β), (α ↔ β) are also well-formed formulae.
- A propositional expression is a well-formed formula if and only if it can be obtained by using above conditions.

1.3.Truth Table

- Truth table gives us operational definitions of important logical operators.
  - By using truth table, the truth values of well-formed formulae are calculated.
- Truth table elaborates all possible truth values of a formula.

The meanings of the logical operators are given by the following truth table.

P	Q	~P	P Λ Q	P V Q	P → Q	P ↔ Q
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	F	T	T

1.4.Equivalence Laws:

Commutation

1. P Λ Q ≅ Q Λ P
2. P V Q ≅ Q V P

Association

1. P Λ (Q Λ R) ≅ (P Λ Q) Λ R
2. P V (Q V R) ≅ (P V Q) V R

Double Negation

~ (~ P) ≅ P

Distributive Laws

1. $P \land (Q \lor R) \cong (P \land Q) \lor (P \land R)$
2. $P \lor (Q \land R) \cong (P \lor Q) \land (P \lor R)$

De Morgan's Laws

1. $\sim (P \land Q) \cong \sim P \lor \sim Q$
2. $\sim (P \lor Q) \cong \sim P \land \sim Q$

Law of Excluded Middle

$P \lor \sim P \cong$ T (true)

Law of Contradiction

$P \land \sim P \cong$ F (false)

2. Propositional Logic – PL

- PL deals with
  - the validity, satisfiability and unsatisfiability of a formula
  - derivation of a new formula using equivalence laws.
- Each row of a truth table for a given formula is called its **interpretation** under which a formula can be true or false.
- A formula α is called **tautology** if and only
  - if α is true for all interpretations.
- A formula α is also called **valid** if and only if
  - it is a **tautology.**
- Let α be a formula and if there exist at least one interpretation for which α is true,
  - then α is said to be **consistent** (satisfiable) i.e., if $\exists$ a model for α, then α is said to be consistent .
- A formula α is said to be inconsistent (unsatisfiable), if and only if
  - α is always false under all interpretations.

- We can translate
  - simple declarative and

conditional (if .. then) natural language sentences into its corresponding propositional formulae.

Example

- Show that " It is humid today and if it is humid then it will rain so it will rain today" is a valid argument.
- **Solution:** Let us symbolize English sentences by propositional atoms as follows:

  A : It is humid

  B : It will rain

- Formula corresponding to a text:

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

- Using truth table approach, one can see that α is true under all four interpretations and hence is valid argument.

Truth Table for $((A \rightarrow B) \land A) \rightarrow B$				
A	B	$A \rightarrow B = X$	$X \land A = Y$	**Y→ B**
T	T	T	T	**T**
T	F	F	F	**T**
F	T	T	F	**T**
F	F	T	F	**T**

- Truth table method for problem solving is
  - simple and straightforward and
  - very good at presenting a survey of all the truth possibilities in a given situation.
- It is an easy method to evaluate

- a consistency, inconsistency or validity of a formula, but the size of truth table grows exponentially.
- Truth table method is good for small values of n.

- For example, if a formula contains n atoms, then the truth table will contain 2^n entries.
  - A formula $\alpha : (P \wedge Q \wedge R) \rightarrow (Q \vee S)$ is **valid** can be proved using truth table.
  - A table of 16 rows is constructed and the truth values of α are computed.
  - Since the truth value of α is true under all 16 interpretations, it is valid.
- It is noticed that if $P \wedge Q \wedge R$ is false, then α is true because of the definition of $\rightarrow$.
- Since $P \wedge Q \wedge R$ is false for 14 entries out of 16, we are left only with two entries to be tested for which α is true.
  - So in order to prove the validity of a formula, all the entries in the truth table may not be relevant.
- Other methods which are concerned with proofs and deductions of logical formula are as follows:
  - Natural Deductive System
  - Axiomatic System
  - Semantic Tableaux Method
  - Resolution Refutation Method

3. Natural deduction method – ND

- ND is based on the set of few deductive inference rules.
- The name natural deductive system is given because it mimics the pattern of natural reasoning.
- It has about 10 deductive inference rules.

Conventions:

- E for Elimination.
- P, P_k , $(1 \leq k \leq n)$ are atoms.

- α_k, $(1 \le k \le n)$ and β are formulae.

Natural Deduction Rules:

Rule 1: I-Λ (Introducing Λ)

I-Λ : If $P_1, P_2, \dots, P_n$ then $P_1 \Lambda P_2 \Lambda \dots \Lambda P_n$

Interpretation: If we have hypothesized or proved $P_1, P_2, \dots$ and P_n, then their conjunction $P_1 \Lambda P_2 \Lambda \dots \Lambda P_n$ is also proved or derived.

Rule 2: E-Λ (Eliminating Λ)

E-Λ : If $P_1 \Lambda P_2 \Lambda \dots \Lambda P_n$ then P_i $(1 \le i \le n)$

Interpretation: If we have proved $P_1 \Lambda P_2 \Lambda \dots \Lambda P_n$, then any P_i is also proved or derived. This rule shows that Λ can be eliminated to yield one of its conjuncts.

Rule 3: I-V (Introducing V)

I-V : If P_i $(1 \le i \le n)$ then $P_1 V P_2 V \dots V P_n$

Interpretation: If any Pi $(1 \le i \le n)$ is proved, then $P_1 V \dots V P_n$ is also proved.

Rule 4: E-V (Eliminating V)

E-V : If $P_1 V \dots V P_n$, $P_1 \rightarrow P, \dots, P_n \rightarrow P$ then P

Interpretation: If $P_1 V \dots V P_n$, $P_1 \rightarrow P, \dots$, and $P_n \rightarrow P$ are proved, then P is proved.

Rule 5: I- → (Introducing →)

I- → : If **from $\alpha_1, \dots, \alpha_n$ infer β is proved** then $\alpha_1 \Lambda \dots \Lambda \alpha_n \rightarrow \beta$ is **proved**

Interpretation: If given $\alpha_1, \alpha_2, \dots$ and α_n to be proved and from these we deduce β then $\alpha_1 \Lambda \alpha_2 \Lambda \dots \Lambda \alpha_n \rightarrow \beta$ is also proved.

Rule 6: E- → (Eliminating →) - *Modus Ponen*

E- → : If $P_1 \rightarrow P$, P_1 then P

Rule 7: I- ↔ (Introducing ↔)

I- ↔ : If $P_1 \rightarrow P_2$, $P_2 \rightarrow P_1$ then $P_1 \leftrightarrow P_2$

Rule 8: E- ↔ (Elimination ↔)

E- $\leftrightarrow$: If $P_1 \leftrightarrow P_2$ then $P_1 \rightarrow P_2$, $P_2 \rightarrow P_1$

Rule 9: I- ~ (Introducing ~)

I- ~ : If **from P infer $P_1 \Lambda \sim P_1$** is proved then **~P** is proved

Rule 10: E- ~ (Eliminating ~)

E- ~ : If **from ~ P infer $P_1 \Lambda \sim P_1$** is proved then **P** is proved

- If a formula β is derived / proved from a set of premises / hypotheses { $\alpha_1,\ldots, \alpha_n$ },
  - then one can write it as **from $\alpha_1, \ldots, \alpha_n$ infer β.**
- In natural deductive system,
  - a theorem to be proved should have a form from $\alpha 1, \ldots, \alpha n$ infer β.
- Theorem **infer β** means that
  - there are no premises and β is true under all interpretations i.e., β is a tautology or valid.
- If we assume that $\alpha \rightarrow \beta$ is a premise, then we conclude that β is proved if α is given i.e.,
  - if ‘from α infer β’ is a theorem then $\alpha \rightarrow \beta$ is concluded.
  - The converse of this is also true.

Deduction Theorem: To prove a formula $\alpha_1 \Lambda \alpha_2 \Lambda \ldots \Lambda \alpha_n \rightarrow \beta$, it is sufficient to prove a theorem **from $\alpha_1, \alpha_2, \ldots, \alpha_n$ infer β**.

Example1: Prove that PΛ(QVR) follows from PΛQ

Solution: This problem is restated in natural deductive system as "**from P ΛQ infer P Λ (Q V R)"**. The formal proof is given as follows:

{Theorem} from P ΛQ infer P Λ (Q V R)

{ premise}	P Λ Q	(1)
{ E-Λ , (1)}	P	(2)
{ E-Λ , (1)}	Q	(3)

{ I-V , (3) }	Q V R	(4)
{ I-Λ, (2, 4)}	**P Λ (Q V R)**	**Conclusion**

Example2: Prove the following theorem:

infer ((Q → P) Λ (Q → R)) → (Q → (P Λ R))

Solution:

- In order to prove **infer** ((Q → P) Λ(Q → R)) → (Q → (P Λ R)), prove a theorem **from {**Q → P, Q → R**} infer** Q → (P Λ R).
- Further, to prove **Q → (P Λ R)**, prove a sub theorem **from** Q **infer** PΛ R

{Theorem} from Q → P, Q → R infer Q → (P Λ R)

{ premise 1}		Q → P	(1)
{ premise 2}		Q → R	(2)
{ sub theorem}	from Q infer	P Λ R	(3)
{ premise }		Q	(3.1)
{ E- → , (1, 3.1) }		P	(3.2)
{E- →, (2, 3.1) }		R	(3.3)
{ I-Λ, (3.2,3.3) }		P Λ R	(3.4)
{ I- →, (3)}		**Q → (P Λ R)**	**Conclusion**

4. Axiomatic System for Propositional Logic:

- It is based on the set of only three axioms and one rule of deduction.
  - It is minimal in structure but as powerful as the truth table and natural deduction approaches.
  - The proofs of the theorems are often difficult and require a guess in selection of appropriate axiom(s) and rules.
  - These methods basically require forward chaining strategy where we start with the given hypotheses and prove the goal.

Axiom1 (A1): $\alpha \rightarrow (\beta \rightarrow \alpha)$

Axiom2 (A2): $(\alpha \rightarrow (\beta \rightarrow \gamma)) \rightarrow ((\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \gamma))$

Axiom3 (A3): $(\sim \alpha \rightarrow \sim \beta) \rightarrow (\beta \rightarrow \alpha)$

Modus Ponen (MP) defined as follows:

Hypotheses: $\alpha \rightarrow \beta$ and α ***Consequent:*** β

Examples: Establish the following:

1. {Q} |-(P→Q) i.e., P→Q is a deductive consequence of {Q}.

{Hypothesis} Q (1)

{Axiom A1} $Q \rightarrow (P \rightarrow Q)$ (2)

{MP, (1,2)} P → Q proved

2. { P → Q, Q → R } |- (P → R) i.e., P → R is a deductive consequence of { P → Q, Q → R }.

{Hypothesis} $P \rightarrow Q$ (1)

{Hypothesis} $Q \rightarrow R$ (2)

{Axiom A1} $(Q \rightarrow R) \rightarrow (P \rightarrow (Q \rightarrow R))$ (3)

{MP, (2, 3)} $P \rightarrow (Q \rightarrow R)$ (4)

{Axiom A2} $(P \rightarrow (Q \rightarrow R)) \rightarrow$

$((P \rightarrow Q) \rightarrow (P \rightarrow R))$ (5)

{MP , (4, 5)} $(P \rightarrow Q) \rightarrow (P \rightarrow R)$ (6)

{MP, (1, 6)} **P → R proved**

4.1.Deduction Theorems in Axiomatic System

Deduction Theorem:

If $\sum$ is a set of hypotheses and α and β are well-formed formulae , then $\{\sum \cup \alpha\}$ |- β implies $\sum$ |- $(\alpha \rightarrow \beta)$.

Converse of deduction theorem:

Given $\sum$ |- $(\alpha \rightarrow \beta)$,

we can prove $\{\sum \cup \alpha\}$ |- β.

Useful Tips

1. Given α, we can easily prove $\beta \rightarrow \alpha$ for any well-formed formulae α and β.

2. Useful tip

If $\alpha \rightarrow \beta$ is to be proved, then include α in the set of hypotheses $\sum$ and derive β from the set $\{\sum \cup \alpha\}$. Then using deduction theorem, we conclude $\alpha \rightarrow \beta$.

Example: Prove $\sim P \rightarrow (P \rightarrow Q)$ using deduction theorem.

Proof: Prove $\{\sim P\}$ |- $(P \rightarrow Q)$ and

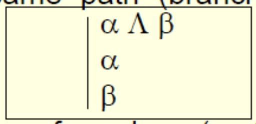
|- $\sim P \rightarrow (P \rightarrow Q)$ follows from deduction theorem.

5. Semantic Tableaux System in PL

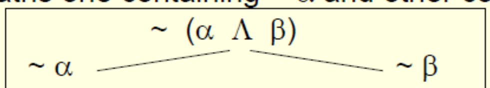
- Earlier approaches require
  - construction of proof of a formula from given set of formulae and are called direct methods.
- In **semantic tableaux**,
  - the set of rules are applied systematically on a formula or set of formulae to establish its consistency or inconsistency.
- *Semantic tableau*
  - binary tree constructed by using semantic rules with a formula as a root
- Assume α and β be any two formulae.

5.1. Semantic Tableaux Rules

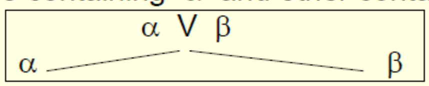
Rule 1: A tableau for a formula ($\alpha \wedge \beta$) is constructed by adding both α and β to the same path (branch). This can be represented as follows:



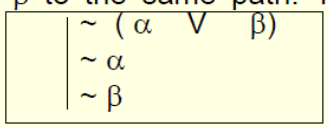
Rule 2: A tableau for a formula $\sim(\alpha \wedge \beta)$ is constructed by adding two alternative paths one containing $\sim \alpha$ and other containing $\sim\beta$.



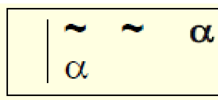
Rule 3: A tableau for a formula ($\alpha \vee \beta$) is constructed by adding two new paths one containing α and other containing β.



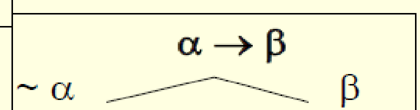
Rule 4: A tableau for a formula $\sim(\alpha \vee \beta)$ is constructed by adding both $\sim \alpha$ and $\sim \beta$ to the same path. This can be expressed as follows:



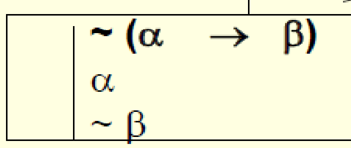
Rule 5:



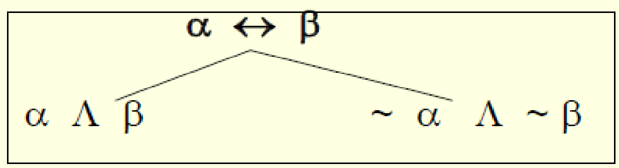
Rule 6:



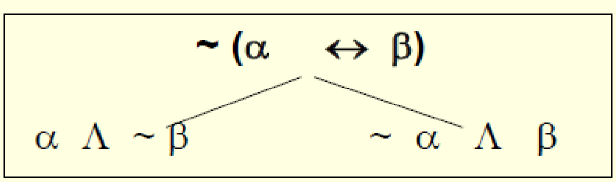
Rule 7:



Rule 8: $\alpha \leftrightarrow \beta \cong (\alpha \wedge \beta) \vee (\sim \alpha \wedge \sim\beta)$



Rule 9: $\sim(\alpha \leftrightarrow \beta) \cong (\alpha \wedge \sim\beta) \vee (\sim \alpha \wedge \beta)$

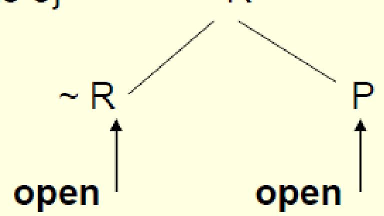


5.2.Consistency and Inconsistency

- If an atom P and ~ P appear on a same path of a semantic tableau,
  - then inconsistency is indicated and such path is said to be c**ontradictory** or **closed** (finished) path.
  - Even if one path remains **non contradictory** or **unclosed** (open), then the formula α at the root of a tableau is **consistent**.
- **Contradictory tableau** (or **finished tableau):**
  - It defined to be a tableau in which all the paths are contradictory or closed (finished).
- If a tableau for a formula α at the root is a contradictory tableau**,**
  - then a formula α is said to be inconsistent.

- Show that α: (Q Λ ~ R) Λ (R → P) is ***consistent*** and find its model.

{Tableau root}	(Q Λ~ R)Λ(R → P)	(1)
{Apply rule 1 to 1}	(Q Λ ~ R)	(2)
	(R→P)	(3)
{Apply rule 1 to 2}	Q	
{Apply rule 6 to 3}	~R	



- { Q = T, R = F } and { P = T , Q = T, R = F } are models of α.

- Show that α : (P Λ Q $\rightarrow$ R) Λ (~P $\rightarrow$ S) Λ Q Λ ~ R Λ ~ S is inconsistent using tableaux method.

(Root) (P ΛQ $\rightarrow$ R) Λ (~P $\rightarrow$ S) Λ Q Λ ~R Λ ~S (1)

{Apply rule 1 to 1} P Λ Q $\rightarrow$ R (2)

~P $\rightarrow$ S (3)

Q

~ R

~ S

{Apply rule 6 to 3} ~ ~P = P S

Closed: {S, ~ S} on the path

{Apply rule 6 to 2)} ~ (P Λ Q) R

Closed { R, ~ R}

~P ~ Q

Closed {P, ~ P} **Closed**{Q, ~ Q}

- α is inconsistent as we get contradictory tableau.

6. Resolution Refutation in PL

- *Resolution refutation:* Another simple method to prove a formula by contradiction.
- Here negation of goal is added to given set of clauses.
  - If there is a refutation in new set using resolution principle then goal is proved
- During resolution we need to identify two clauses,
  - one with positive atom (P) and other with negative atom (~ P) for the application of resolution rule.
- Resolution is based on modus ponen inference rule.

6.1.Disjunctive & Conjunctive Normal Forms

- *Disjunctive Normal Form* (DNF): A formula in the form (L_{11} Λ ….. Λ L_{1n}) V ..… V (L_{m1} Λ ….. Λ L_{mk}), where all L_{ij} are literals.
  - Disjunctive Normal Form is disjunction of conjunctions.
- *Conjunctive Normal Form* (CNF): A formula in the form (L_{11} V ….. V L_{1n}) Λ …… Λ (L_{p1} V ….. V L_{pm}) , where all L_{ij} are literals.

  - CNF is conjunction of disjunctions or
  - CNF is conjunction of clauses

- *Clause:* It is a formula of the form ($L_1 V \ldots V L_m$), where each L_k is a positive or negative atom.

6.2.Conversion of a Formula to its CNF

- Each PL formula can be converted into its equivalent CNF.
- Use following equivalence laws:
    - $P \rightarrow Q \cong \sim P \ V \ Q$
    - $P \leftrightarrow Q \cong (P \rightarrow Q) \ \Lambda \ (Q \rightarrow P)$
  - Double Negation
    - $\sim\sim P \cong P$
  - (De Morgan's law)
    - $\sim (P \ \Lambda \ Q) \cong \sim P \ V \ \sim Q$
    - $\sim (P \ V \ Q) \cong \sim P \ \Lambda \ \sim Q$
  - (Distributive law)

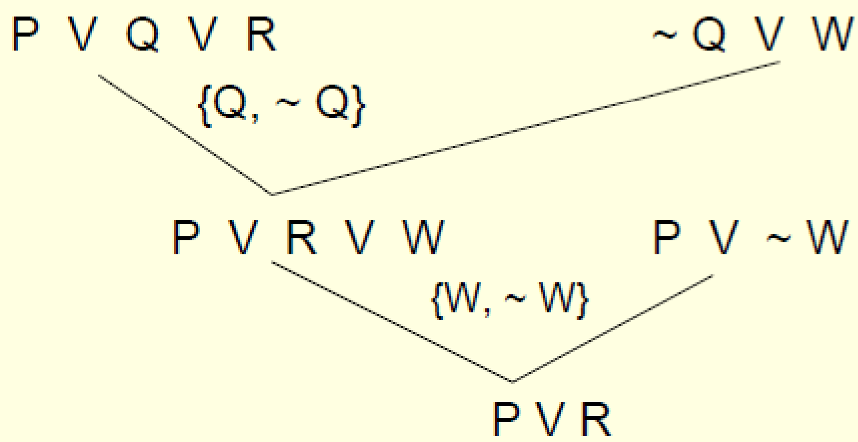
  $P \ V \ (Q \ \Lambda \ R) \cong (P \ V \ Q) \ \Lambda \ (P \ V \ R)$

6.3Resolvent of Clauses

- If two clauses C_1 and C_2 contain a complementary pair of literals {L, ~L},
  - then these clauses may be resolved together by deleting L from C_1 and ~ L from C_2 and constructing a new clause by the disjunction of the remaining literals in C_1 and C_2.
- The new clause thus generated is called **resolvent** of C_1 and C_2 .
  - Here C1 and C2 are called parents of resolved clause.
- Inverted binary tree is generated with the last node (root) of the binary tree to be a resolvent.

This is also called resolution tree.

- Find resolvent of the following clauses:
  - C_1 = P V Q V R; C_2 = ~ Q V W; C_3 = P V ~ W
- Inverted Resolution Tree



- Resolvent(C1,C2, C3) = P V R

6.4Logical Consequence

- **Theorem1**: If C is a resolvent of two clauses C_1 and C_2 , then C is a *logical consequence* of {C_1 , C_2 }.
  - A deduction of an empty clause (or resolvent as contradiction) from a set S of clauses is called a *resolution refutation* of S.
- **Theorem2:** Let S be a set of clauses. A clause C is a *logical consequence* of S iff the set S'= S ∪ {~ C} is *unsatisfiable*.
  - In other words, C is a logical consequence of a given set S iff an empty clause is deduced from the set S'.

- Show that C V D is a logical consequence of
  - S ={AVB, ~ AVD, C V~ B} using resolution refutation principle.
- First we will add negation of logical consequence
  - i.e., ~ (C V D) ≅ ~C Λ ~D to the set S.
  - Get S' = {A V B, ~ A V D, C V~ B, ~C, ~D}.
- Now show that S' is unsatisfiable by deriving contradiction using resolution principle.