UNIT 4 PREDICATE ANDPROPOSITIONAL LOGIC

4.1 INTRODUCTION

Logic is the study and analysis of the nature of the valid argument, the reasoning tool by which valid inferences can be drawn from a given set of facts and premises. It is the basis on which all the sciences are built, and this mathematical theory of logic is called symbolic logic. The English mathematician George Boole (1815-1864) seriously studied and developed this theory, called symbolic logic.

The reason why the subject-matter of the study is called Symbolic Logic is that symbols are used to denote facts about objects of the domain and relationships between these objects. Then the symbolic representations and not the original facts and relationships are manipulated in order to make conclusions or to solve problems.

The basic building blocks of arguments in symbolic logic are declarative sentences, called propositions or statements. In MCS – 212 i.e., Discrete Mathematics you learned about predicates and propositions and ways of combining them to form more complex propositions. Also, you learned about the propositions that contain the quantifiers 'for All' and 'there exists'. In symbolic logic, the goal is to determine which propositions are true and which are false. Truth table a tool to find out all possible outcome of a proposition's truth value was also discussed in MCS-212.

Logical rules have the power to give accuracy to mathematical statements. These rules come to rescue when there is a need to differentiate between valid and invalid mathematical arguments. Symbolic logic may be thought of as a formal language for representing facts about objects and relationships between objects of a problem domain along with a precise inferencing mechanism for reasoning and deduction.

Using symbolic logic, we can formalize our arguments and logical reasoning in a manner that can easily show if the reasoning is valid, or is a fallacy. How we symbolize the reasoning is what is presented in this unit.

4.2 OBJECTIVES

After going through this unit, you should be able to:

1. Understand the meaning of propositional logic.
2. Differentiate between atomic and compound propositions.
3. Know different types of connectives, their associated semantics and corresponding truth tables.
4. Define propositional rules of inference and replacement.
5. Differentiate between valid and satisfiable arguments.

4.3 INTRODUCTION TO PROPOSITIONAL LOGIC

Apart from the application of logic in mathematics, it also helps in various other tasks related to computer science. It is widely used to design the electronic circuitry, programming of android applications, applying artificial intelligence to different tasks, etc. In simple terms, a proposition is a statement which is either true or false.

Consider the following statements:

1. Earth revolves around the sun.
2. Water freezes at 100° Celsius.
3. An hour has 3600 seconds.
4. 2 is the only even prime number.
5. Mercury is the closest planet to the Sun in the solar system.
6. The USA lies on the continent of North America.
7. $1 + 2 = 4$.
8. 15 is a prime number.
9. Moon rises in the morning and sets in the evening.
10. Delhi is the capital of India.

For all the above statements, one can easily conclude whether the particular statement is true or false so these are propositions. First statement is a universal truth. Second statement is false as the water freezes at 0° Celsius. Third statement is again a universal truth. Fourth statement is true. Fifth statement is also true as it is again a universal truth. On similar lines, the sixth statement is also true. Seventh and eighth statements are false again as they deny the basic mathematical rules. The Ninth statement is a negation of the universal truth so it is a false statement. The Tenth Statement is also true.

Now consider the following statements:

1. What is your name?
2. $a + 5 = b$.
3. Who is the prime minister of India?
4. p is less than 5.
5. Pay full attention while you are in the class.
6. Let's play football in the evening.
7. Don't behave like a child, you are grown up now!
8. How much do you earn?
9. $\angle X$ is an acute angle greater than 27°.

For all the above statements, we can't say anything about their truthfulness so they are not propositions. First and third statements are neither true nor false as they are interrogative in nature. Also, we can't say anything about the truthfulness of the second statement until and unless we have the values of a and b. Similar reasoning applies to the fourth statement as well. We can't say anything about fifth, sixth and seventh statements again as they are informative statements. Eighth statement is again interrogative in nature. Again, we can't say anything about the truthfulness of the ninth statement until and unless we have the value of $\angle X$.

Propositional logic has the following facts:

1. Propositional statements can be either true or false, they can't be both simultaneously.
2. Propositional logic is also referred to as binary logic as it works only on the two values 1 (True) and 0 (False).

3. Symbols or symbolic variables such as x, y, z, P, Q, R, etc. are used for representing the logic and propositions.
4. Any proposition or statement which is always valid (true) is known as a tautology.
5. Any proposition or statement which is always invalid (false) is known as a contradiction.
6. A table listing all the possible truth values of a proposition is known as a truth table.
7. Objects, relations (or functions) and logical connectives are the basic building blocks of propositional logic.
8. Logical connectives are also referred to as logical operators.
9. Statements which are interrogative, informative or opinions such as "Where is Chandni Chowk located?", "Mumbai is a good city to live in", "Result will be declared on 31st March" are not propositions.

4.4 SYNTAX OF PROPOSITIONAL LOGIC

The syntax of propositional logic allows two types of sentences to represent knowledge. The two types are as follows:

4.4.1 Atomic Propositions

These are simplest propositions containing a single proposition symbol and are either true or false. Some of the examples of atomic propositions are as follows:

1. "Venus is the closest planet to the Sun in the solar system" is an atomic preposition since it is a false fact.
2. "7 – 3 = 4" is an atomic preposition as it is a true fact.

4.4.2 Compound Propositions

They are formed by a collection of atomic propositions joined with logical connectives or logical operators. Some of the examples of compound propositions are as follows:

1. The Sun is very bright today and its very hot outside.
2. Diana studies in class 8th and her school is in Karol Bagh.

☞ **Check Your Progress 1**

Which of the following statements are propositions? Write yes or no.

1. How are you?
2. Sachin Tendulkar is one of the best cricketers in India.
3. The honorable Ram Nath Kovind is the 10th and current president of India.
4. Lord Ram of the kingdom of Ayodhya is an example of a people's king.
5. In which year did prophet Muhammad received verbal revelations from the Allah in the cave Mount Hira presently located in the Saudi Arabia?
6. Akbar was the founder of the Mughal dynasty in India.
7. In the year 2019, renowned actor Shri Amitabh Bachchan was awarded with the Padma Vibhushan which is the second highest civilian honour of the republic of India.
8. One should avoid eating fast food in order to maintain good health.
9. What is your age?
10. The first case of COVID-19 was detected in China.
11. Name the author of the book series "Shiva Trilogy".
12. Former prime minister of India, Shri Atal Bihari Vajpayee was a member of which political party?
13. Wing Commander Rakesh Sharma is the only Indian citizen to travel in space till date.
14. Where do you live?

4.5 LOGICAL CONNECTIVES

Logical connectives are the operators used to join two or more atomic propositions (operands). The joining should be done in a way that the logic and truth value of the obtained compound proposition is dependent on the input atomic propositions and the connective used.

4.5.1 Conjunction

A proposition "A ∧ B"with connective ∧ is known as *conjunction* of A and B. It is a proposition (or operation) which is true only when both the constituent propositions are true. Even if one of the input propositions is false then the output is also false. It is also referred to as AND-ing the propositions. Example:

Ram is a playful boy and he loves to play football. It can be written as:

A = Ram is a playful boy.

B = Ram loves to play football.

A ∧ B = Ram is a playful boy and he loves to play football.

4.5.2 Disjunction

A proposition "A ∨ B"with connective ∨ is known as *disjunction* of A and B. It is a proposition (or operation) which is true when at least one of the constituent propositions are true. The output is false only when both the input propositions are false. It is also referred to as OR-ing the propositions. Example:

I will go to her house or she will come to my house. It can be written as:

A = I will go to her house.

B = She will come to my house.

A ∨ B = I will go to her house or she will come to my house.

4.5.3 Negation

The proposition ¬ A (or ~A) with ¬ (or ~) connective is known as *negation* of A. The purpose of negation is to negate the logic of given proposition. If A is true, its negation will be false, and if A is false, its negation will be true. Example:

University is closed. It can be written as:

A = University is closed.

¬ A = University is not closed.

4.5.4 Implication

The proposition A → B with → connective is known as *A implies B*. It is also called *if-then* proposition. Here, the second proposition is a logical consequence of the first proposition. For example, “If Mary scores good in examinations, I will buy a mobile phone for her”. In this case, it means that if Mary scores good, she will definitely get the mobile phone but it doesn’t mean that if she performs bad, she won’t get the mobile phone. In set notation, we can also say that A ⊆ B i.e., if something exists in the set A, then it necessarily exists in the set B. Another example:

If you score above 90%, you will get a mobile phone.

A = You score above 90%.

B = You will get a mobile phone.

A → B = If you score above 90%, you will get a mobile phone.

4.5.5 Bi-conditional

A proposition A ⟷ B with connective ⟷ is known as a *biconditional* or *if-and-only-if* proposition. It is true when both the atomic propositions are true or both are false. A classic example of biconditional is “A triangle is equivalent if and only if all its angles are60° each”. This statement means that if a triangle is an equivalent triangle, then all of its angles are 60° each. There is one more associated meaning with this statement which means that if all the interior angles of a triangle are of 60° each then it's an equivalent triangle. Example:

You will succeed in life if and only if you work hard.

A = You will succeed in life.

B = You work hard.

A ⟷ B = You will succeed in life if and only if you work hard.

☞ **Check Your Progress 2**

Which of the following propositions are atomic and which are compound?

1. The first battle of Panipat was fought in 1556.
2. Jack either plays cricket or football.

3. Posthumously, at the age of 22, Neerja Bhanot became the youngest recipient of the Ashok Chakra award which is India's highest peacetime gallantry decoration.
4. Chandigarh is the capital of the Indian states Haryana and Punjab.
5. Earth takes 365 days, 5 hours, 59 minutes and 16 seconds to complete one revolution around the Sun.
6. Dermatology is the branch of medical science which deals with the skin.
7. Indian sportspersons won 7 medals at the 2020 Summer Olympics and 19 medals at the 2020 Summer Paralympics both held at the Japanese city of Tokyo.
8. Harappan civilization is considered to be the oldest human civilization and it lies in the parts of present-day India, Pakistan and Afghanistan.
9. IGNOU is a central university and offers courses through the distance learning mode.
10. Uttarakhand was carved out of the Indian state of Uttar Pradesh in the year 2000.

4.6 SEMANTICS

You had already learned various of the concepts to be covered in this unit, in MCS-212 i.e., Discrete Mathematics, here is a quick revision to those concepts and we will extend our discussion to the advanced concepts, which are useful for our field of work i.e., Artificial Intelligence. In MCS – 212 i.e., Discrete Mathematics you learned that **Propositions are the** declarative sentence or statements which is either true or false, but not both, such sentences can either be **universally true** or **universally false**.

On the other hand, consider the declarative sentence 'Women are more intelligent than men'. Some people may think it is true while others may disagree. So, it is neither universally true nor universally false. Such a sentence is not acceptable as a statement or proposition in mathematical logic.

Note that a **proposition should be either uniformly true or uniformly false**.

In propositional logic, as mentioned earlier also, symbols are used to denote propositions. For instance, we may denote the propositions discussed above as follows:

P : The sun rises in the west,
Q : Sugar is sweet,
R : Ram has a Ph.D. degree.

The symbols, such as P, Q, and R, that are used to denote propositions, are called **atomic formulas, or atoms.,** in this case, the truth-value of P is False, the truth-value of Q is True and the truth-value of R, though not known yet, is exactly one of 'True' or 'False', depending on whether Ram is actually a Ph. D or not.

At this stage, it may be noted that once symbols are used in place of given statements in, say, English, then the propositional system, and, in general, a symbolic system is aware **only** of symbolic representations, and the associated truth values. The system operates only on these representations. And, except for possible final translation, is **not aware** of the original statements, generally given in some natural language, say, English.

When you're talking to someone, do you use very simple sentences only? Don't you use more complicated ones which are joined by words like 'and', 'or', etc? In the same way, most statements in mathematical logic are combinations of simpler statements joined by words and phrases like 'and'. 'or', 'if … then'. 'If and only if', etc. We can build, from atoms, *more complex propositions*, sometimes called***compound propositions***, by using logical **connectives,**

The Logical Connectives are used to frame compound propositions, and they are as follows:

a) Disjunction The **disjunction** of two propositions p and q is the compound statement **por q,** denoted by **p $\vee$ q**.

The **exclusive disjunction** of two propositions p and q is the statement **'Either of the two (i.e. p or q) can be true, but both can't be true'.** We denote this by **p $\oplus$ q .**

b) Conjunction We call the compound statement **'p and q'** the **conjunction** of the statements p and q. We denote this by **p $\wedge$ q**.

c) Negation The **negation** of a proposition p is **'not p'**, denoted by ~p.

d) Implication (Conditional Connectives) Given any two propositions p and q, we denote the statement **'If p, then q' by p $\rightarrow$ q.** We also read this as 'p **implies** q'. or 'p is sufficient for q', or 'p **only if** q'. We also call p the **hypothesis** and q the conclusion. Further, a statement of the form p $\rightarrow$ q is called a **conditional statement** or a **conditional proposition**.

Let p and q be two propositions. The compound statement (p $\rightarrow$ q) $\wedge$(q $\rightarrow$ p) is the **bi-conditional** of p and q. We denote it by **p $\leftrightarrow$ q,** and read it as 'p **if and only** q'

Note : The two connectives $\rightarrow$ and $\leftrightarrow$ are called **conditional connectives**

The rule of precedence: The order of preference in which the connectives are applied in a formula of propositions that has no brackets is

i) ~
ii) $\wedge$
iii) $\vee$ and $\oplus$
iv) $\rightarrow$ and $\leftrightarrow$

Note that the 'inclusive or' and 'exclusive or' are both third in the order of preference. However, if both these appear in a statement, we first apply the left most one. So, for instance, in p ∨ q ⊕ ~ p, we first apply ∨ and then ⊕. The same applies to the 'implication' and the 'biconditional', which are both fourth in the order of preference.

Let's see the working of the various concepts learned above, with the help of truth tables. In the following truth table, we write every TRUE value as Tand every FALSE value as F.

4.6.1 Negation Truth Table

α	~ α
_F(0)_	_T(1)_
_T(1)_	_F(0)_

4.6.2 Conjunction/Disjunction/Implication /Biconditional Truth Table

α_1	α_2	Conjunction α_1 Λα_2	Disjunction α_1Vα_2	Implication α_1 → α_2	Biconditional α_1↔α_2
_F(0)_	_F(0)_	_F(0)_	_F(0)_	_T(1)_	_T(1)_
_F(0)_	_T(1)_	_F(0)_	_T(1)_	_T(1)_	_F(0)_
_T(1)_	_F(0)_	_F(0)_	_T(1)_	_F(0)_	_F(0)_
_T(1)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_

4.6.3 Conjunction and Disjunction with three variables

α_1	α_2	α_3	(α_1 Λα_2)	(α_2 Λα_3)	(α_1 Vα_2)	(α_2 Vα_3)	(α_1 Λα_2) Λα_3 or α_1 Λ (α_2 Λα_3)	(α_1 Vα_2) Vα_3 or α_1 V (α_2 Vα_3)
_T(1)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_
_T(1)_	_T(1)_	_F(0)_	_T(1)_	_F(0)_	_T(1)_	_T(1)_	_F(0)_	_T(1)_
_T(1)_	_F(0)_	_T(1)_	_F(0)_	_F(0)_	_T(1)_	_T(1)_	_F(0)_	_T(1)_
_T(1)_	_F(0)_	_F(0)_	_F(0)_	_F(0)_	_T(1)_	_F(0)_	_F(0)_	_T(1)_
_F(0)_	_T(1)_	_T(1)_	_F(0)_	_T(1)_	_T(1)_	_T(1)_	_F(0)_	_T(1)_
_F(0)_	_T(1)_	_F(0)_	_F(0)_	_F(0)_	_T(1)_	_T(1)_	_F(0)_	_T(1)_
_F(0)_	_F(0)_	_T(1)_	_F(0)_	_F(0)_	_F(0)_	_T(1)_	_F(0)_	_T(1)_
_F(0)_	_F(0)_	_F(0)_	_F(0)_	_F(0)_	_F(0)_	_F(0)_	_F(0)_	_F(0)_

Using these logical connectives, we can transform any sentence in to its equivalent mathematical representation in Symbolic Logic and that representation is referred as Well

Form Formula (WFF), you had already learned a lot about WFF in MCS-212, lets briefly discuss it here also, as it has wide applications in Artificial Intelligence also.

A Well-formed formula, or *wff* or *formula* in short, in the propositional logic is defined recursively as follows:

1. An atom is a wff.
2. If A is a wff, then (~A) is a wff.
3. If A and B are wffs, then each of $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, and $(A \leftrightarrow B)$ is a wff.
4. Any wff is obtained only by applying the above rules.

From the above recursive definition of a wff it is not difficult to see that expression:

$((P \rightarrow (Q \wedge (\sim R))) \leftrightarrow S)$ is a wff; because , to begin with, each of P, Q , (~ R) and S, by definitions is a wff. Then, by recursive application, the expression: $(Q \wedge (\sim R))$ is a wff. Again, by another recursive application, the expression: $(P \rightarrow (Q \wedge (\sim R)))$ is a wff. And, finally the expression given initially is a wff.

Further, it is easy to see that according to the recursive definition of a wff, each of the expressions: $(P \rightarrow (Q \wedge))$ and $(P (Q \wedge R))$ is **not** a wff.

Some pairs of parentheses may be dropped, for simplification. For example,

$A \vee B$ and $A \rightarrow B$ respectively may be used instead of the given wffs $(A \vee B)$ and $(A \rightarrow B)$, respectively. We can omit the use of parentheses by assigning *priorities in increasing order* to the connectives as follows:

$$\leftrightarrow, \quad \rightarrow, \quad \vee, \quad \vee, \quad \sim.$$

Thus, '$\leftrightarrow$' has least priority and '~' has highest priority. Further, if in an expression, there are no parentheses and two connectives between three atomic formulas are used, then the operator with higher priority will be applied first and the other operator will be applied later.

For example: Let us be given the wff $P \rightarrow Q \wedge \sim R$ without parenthesis. Then among the operators appearing in wff, the operator '~' has highest priority. Therefore, ~ R is replaced by (~R). The equivalent expression becomes $P \rightarrow Q \wedge (\sim R)$. Next, out of the two operators viz '$\rightarrow$' and '$\wedge$', the operators '$\wedge$' has higher priority. Therefore, by applying parentheses appropriately, the new expression becomes $P \rightarrow (Q \wedge (\sim R))$. Finally, only one operator is left. Hence the *fullyparenthesized expression* becomes $(P \rightarrow (Q \wedge (\sim R)))$

Following are the rules of finding the truth value or meaning of a wff, when truth values of the atoms appearing in the wff are known or given.

1. The wff ~ A is *True* when A is *False*, and ~ A is *False* when A is *true*. The wff ~ A is called the ***negation*** of A.

2. The wff $(A \wedge B)$ is True if A and B are both True; otherwise, the wff $A \wedge B$ is False. The wff $(A \wedge B)$ is called the **conjunction** of A and B.
3. The wff $(A \vee B)$ is true if at least one of A and B is True; otherwise, $(A \vee B)$ is False. $(A \vee B)$ is called the **disjunction** of A and B.
4. The wff $(A \rightarrow B)$ is False if A is True and B is False; otherwise, $(A \rightarrow B)$ is True. The wff $(A \rightarrow B)$ is read as "If A, then B," or "**A implies B**." The symbol '$\rightarrow$' is called **implication**.
5. The wff $(A \leftrightarrow B)$ is True whenever A and B have the same truth values; otherwise $(A \leftrightarrow B)$ is False. The wff $(A \leftrightarrow B)$ is read as "**A if and only if B**."

☞ **Check Your Progress 3**

Q1. Draw the truth table for the following:

a) α_2⟷(~ α_1 → (α_1∨α_2))

b) (~α_1⟷ (α_2⟷α_3)) ∨ (α_3 ∧α_2)

c) ((α_1 ∧α_2) → α_3) ∨ ~α_4

d) ((α_1 → ~ α_2) ⟷α_3) → ~ (α_1 V α_1)

Q2. Verify the De Morgan's Laws using Truth Tables

Q3. Write WFF for the following statements:

a) Every Person has Mother

b) There is a woman and she is mother of Siya

4.7 PROPOSITIONAL RULES OF INFERENCE

We need intelligent computers based on the concept of artificial intelligence which are able to infer new "knowledge" or logic from the existing logic using the theory of inference. Inference rules help us to infer new propositions and conclusions based on existing propositions and logic. They act as templates to generate new arguments from the premises or predicates. We deduce new statements from the statements whose truthfulness is already known. These rules come to the rescue when we need to prove something logically. In general, inference rules preserve the truth. Depending on the problem, some or all of these rules may be applied to infer new propositions. The procedure of determining whether a proposition is a conclusion of the given propositions is known as inferring a proposition. The inference rules are described below.

4.7.1 Modus Ponens (MP)

It states that if the propositions A → B and A are true, then B is also true. Modus Ponens is also referred to as the implication elimination because it eliminates the implication A → B and results in only the proposition B. It also affirms the truthfulness of antecedent. It is written as:

$$\alpha\_1 \rightarrow \alpha\_2,\ \alpha\_1 \ => \alpha\_2$$

4.7.2 Modus Tollens (MT)

It states that if A → B and ¬ B are true then ¬ A is also true. Modus Tollens is also referred to as denying the consequent as it denies the truthfulness of the consequent. The rule is expressed as:

$$\alpha\_1 \rightarrow \alpha\_2,\ \sim\alpha\_2 \ => \sim\alpha\_1$$

4.7.3 Disjunctive Syllogism (DS)

Disjunctive Syllogism affirms the truthfulness of the other proposition if one of the propositions in a disjunction is false.

Rule 1:$\alpha\_1 \vee \alpha\_2, \sim\alpha\_1 => \alpha\_2$

Rule 2:$\alpha\_1 \vee \alpha\_2, \sim\alpha\_2 => \alpha\_1$

4.7.4 Addition

The rule states that if a proposition is true, then its disjunction with any other proposition is also true.

Rule 1:$\alpha\_1 => \alpha\_1 \vee \alpha\_2$

Rule 2:$\alpha\_2 => \alpha\_1 \vee \alpha\_2$

4.7.5 Simplification

Simplification means that if we have a conjunction, then both the constituent propositions are also true.

Rule 1:$\alpha\_1 \wedge \alpha\_2 => \alpha\_1$

Rule 2:$\alpha\_1 \wedge \alpha\_2 => \alpha\_2$

4.7.6 Conjunction

Conjunction states if two propositions are true, then their conjunction is also true. It is written as:

$$\alpha\_1, \alpha\_2 \Rightarrow \alpha\_1 \wedge \alpha\_2$$

4.7.7 Hypothetical Syllogism (HS)

The rule says that the conclusion $\alpha\_1 \rightarrow \alpha\_3$ is true, whenever conditional statements $\alpha\_1 \rightarrow \alpha\_2$ and $\alpha_2 \rightarrow \alpha_3$hold the truth values. This rules also shows the transitive nature of implication operator.

$$\alpha\_1 \rightarrow \alpha\_2, \alpha\_2 \rightarrow \alpha\_3 \Rightarrow \alpha\_1 \rightarrow \alpha\_3$$

4.7.8 Absorption

The rule states that if the literal α_1 conditionally implies another literal α_2 i.e.,$\alpha\_1 \rightarrow \alpha\_2$ is true, then $\alpha\_1 \rightarrow (\alpha\_1 \wedge \alpha\_2)$ also holds.

$$\alpha\_1 \rightarrow \alpha\_2 \Rightarrow \alpha\_1 \rightarrow (\alpha\_1 \wedge \alpha\_2)$$

4.7.9 Constructive Dilemma (CD)

According to the rule, if proposition $(\alpha\_1 \vee \alpha\_3)$ and proposition $((\alpha_1 \rightarrow \alpha_2) \wedge (\alpha_3 \rightarrow \alpha_4))$have true values, then the well-formed formula $(\alpha\_2 \vee \alpha\_4)$ also holdsa true value.

$$(\alpha\_1 \vee \alpha\_3), (\alpha\_1 \rightarrow \alpha\_2) \wedge (\alpha\_3 \rightarrow \alpha\_4) \Rightarrow \alpha\_2 \vee \alpha\_4$$

4.8 PROPOSITIONAL RULES OF REPLACEMENT

We learned the concepts of Predicate and Propositional logic in MCS-212 (Discrete Mathematics), just to brief the understanding, it is to remind you here that "***a proposition is a specialized statement whereas Predicate is a generalized statement***". To be more specific the propositions use the logical connectives only and the predicates uses logical connectives and quantifiers (universal and existential), both.

Note :$\exists$ is the symbol used for the Existential quantifier and $\forall$ is used for the Universal quantifier.

In predicate logic, a replacement rule is used to replace an argument or a set of arguments with an equivalent argument. By equivalent arguments, we mean that the logical interpretation of the arguments is the same. These rules are used to manipulate the propositions. Also, the axioms and the propositional rules of inference are used as an aid to generate the replacement rules. Given below is the table summarizing the different replacement rules over the propositions α_1, α_2 and α_3.

Replacement Rule	**Proposition**	**Equivalent**
Tautology (Conjunction of a statement with itself always implies the statement)	$\alpha_1 \wedge \alpha_1$	α_1
Double Negation (DN) (Also Called negation of negation)	$(\sim(\sim\alpha_1))$	α_1
Commutativity (Valid for Conjunction and Disjunction)	$\alpha_1 \wedge \alpha_2$	$\alpha_2 \wedge \alpha_1$
	$\alpha_1 \vee \alpha_2$	$\alpha_1 \vee \alpha_2$
Associativity (Valid for Conjunction and Disjunction)	$(\alpha_1 \wedge \alpha_2) \wedge \alpha_3$	$\alpha_1 \wedge (\alpha_2 \wedge \alpha_3)$
	$(\alpha_1 \vee \alpha_2) \vee \alpha_3$	$\alpha_1 \vee (\alpha_2 \vee \alpha_3)$
DeMorgan's Laws	$\sim(\alpha_1 \wedge \alpha_2)$	$(\sim\alpha_1) \vee (\sim\alpha_2)$
	$\sim(\alpha_1 \vee \alpha_2)$	$(\sim\alpha_1) \wedge (\sim\alpha_2)$
Transposition (Defined over implication)	$\alpha_1 \rightarrow \alpha_2$	$(\sim\alpha_2) \rightarrow (\sim\alpha_1)$
Exportation	$\alpha_1 \rightarrow (\alpha_2 \rightarrow \alpha_3)$	$(\alpha_1 \wedge \alpha_2) \rightarrow \alpha_3$
Distribution (AND over OR, and OR over AND)	$\alpha_1 \wedge (\alpha_2 \vee \alpha_3)$	$(\alpha_1 \wedge \alpha_2) \vee (\alpha_1 \wedge \alpha_3)$
	$\alpha_1 \vee (\alpha_2 \wedge \alpha_3)$	$(\alpha_1 \vee \alpha_2) \wedge (\alpha_1 \vee \alpha_3)$
Material Implication (MI) (Defined over implication)	$\alpha_1 \rightarrow \alpha_2$	$\sim \alpha_1 \vee \alpha_2$
Material Equivalence (ME)(Defined over biconditional)	$\alpha_1 \leftrightarrow \alpha_2$	$(\alpha_1 \rightarrow \alpha_2) \wedge (\alpha_2 \rightarrow \alpha_1)$

4.9 VALIDITY AND SATISFIABILITY

An argument is called *valid* in propositional logic if the argument is a tautology for all possible combinations of the given premises. An argument is called *satisfiable* if the argument is true for at least one combination of the premises.

Consider the example given below. Kindly note, every TRUE value is written as _T(1)_, and every FALSE value is written as _F(0)_.

Example 1: Consider the expression $\alpha_1 \vee (\alpha_2 \vee \alpha_3) \vee (\sim\alpha_2 \wedge \sim \alpha_3)$, whose truth table is given below.

α_1	**α_2**	**α_3**	**$\sim\alpha_2$**	**$\sim\alpha_3$**	**$\alpha_1 \vee \alpha_2$**	**$\sim\alpha_2 \wedge \sim\alpha_3$**	**$\alpha_1 \vee (\alpha_2 \vee \alpha_3) \vee (\sim\alpha_2 \wedge \sim\alpha_3)$**
_F(0)_	_F(0)_	_F(0)_	_T(1)_	_T(1)_	_F(0)_	_T(1)_	_T(1)_
_F(0)_	_F(0)_	_T(1)_	_T(1)_	_F(0)_	_T(1)_	_F(0)_	_T(1)_
_F(0)_	_T(1)_	_F(0)_	_F(0)_	_T(1)_	_T(1)_	_F(0)_	_T(1)_
_F(0)_	_T(1)_	_T(1)_	_F(0)_	_F(0)_	_T(1)_	_F(0)_	_T(1)_
_T(1)_	_F(0)_	_F(0)_	_T(1)_	_T(1)_	_F(0)_	_T(1)_	_T(1)_
_T(1)_	_F(0)_	_T(1)_	_T(1)_	_F(0)_	_T(1)_	_F(0)_	_T(1)_
_T(1)_	_T(1)_	_F(0)_	_F(0)_	_T(1)_	_T(1)_	_F(0)_	_T(1)_
_T(1)_	_T(1)_	_T(1)_	_F(0)_	_F(0)_	_T(1)_	_F(0)_	_T(1)_

It is noteworthy from the above truth table that the argument $\alpha_1 \vee (\alpha_2 \vee \alpha_3) \vee (\sim\alpha_2 \wedge \sim\alpha_3)$ is a valid argument as it has true values for all possible combinations of the premises α_1, α_2 and α_3.

Example 2: Consider the expression $\alpha_1 \wedge ((\alpha_2 \wedge \alpha_3) \vee (\alpha_1 \wedge \alpha_3))$ for the

α_1	**α_2**	**α_3**	**$\alpha_1 \wedge \alpha_3$**	**$\alpha_2 \wedge \alpha_3$**	**$(\alpha_2 \wedge \alpha_3) \vee (\alpha_1 \wedge \alpha_3)$**	**$\alpha_1 \wedge ((\alpha_2 \wedge \alpha_3) \vee (\alpha_1 \wedge \alpha_3))$**
_F(0)_	_F(0)_	_F(0)_	_F(0)_	_F(0)_	_F(0)_	_F(0)_
_F(0)_	_F(0)_	_T(1)_	_T(1)_	_F(0)_	_T(1)_	_F(0)_
_F(0)_	_T(1)_	_F(0)_	_F(0)_	_F(0)_	_F(0)_	_F(0)_
_F(0)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_	_F(0)_
_T(1)_	_F(0)_	_F(0)_	_T(1)_	_F(0)_	_T(1)_	_T(1)_
_T(1)_	_F(0)_	_T(1)_	_T(1)_	_F(0)_	_T(1)_	_T(1)_
_T(1)_	_T(1)_	_F(0)_	_T(1)_	_F(0)_	_T(1)_	_T(1)_
_T(1)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_

propositions α_1,α_2, and $\alpha_3$whose truth table is given below.

In this example, as the argument $\alpha_1 \wedge ((\alpha_2 \wedge \alpha_3) \vee (\alpha_1 \wedge \alpha_3))$ is true for a few combinations of the premises α_1, α_2, and α_3, hence it is a satisfiable argument.

4.10 INTRODUCTION TO PREDICATE LOGIC

Now it's time to understand the difference between the Proposition and the Predicate(also known as propositional function). In short, a proposition is a specialized statement whereas Predicate is a generalized statement. To be more specific the propositions use the logical connectives only and the predicates uses logical connectives and quantifiers (universal and existential), both.

Note : $\exists$ is the symbol used for the Existential quantifier and $\forall$ is used for the Universal quantifier.

Let's understand the difference through some more detail, as given below.

A propositional function, or a **predicate,** a variable x in a sentence p(x) involving x becomes a proposition when we give x a definite value from the set of values it can take. We usually denote such functions by p(x), q(x), etc. The set of values x can take is called the universe of discourse.

So, if p(x) is '$x > 5$', then p(x) is not a proposition. But when we give x particular values, say $x = 6$ or $x = 0$, then we get propositions. Here, p(6) is a true proposition and p(0) is a false proposition.

Similarly, if q(x) is 'x has gone to Patna.', then replacing x by 'Taj Mahal' gives us a false proposition.

Note that a predicate is usually not a proposition. But, of course, every proposition is a prepositional function in the same way that every real number is a real-valued function, namely, the constant function.

Now, can all sentences be written in symbolic from by using only the logical connectives? What about sentences like 'x is prime and $x + 1$ is prime for some x.'? How would you symbolize the phrase 'for some x', which we can rephrase as 'there exists an x'? You must have come across this term often while studying mathematics. **We use the symbol '$\exists$' to denote this quantifier, 'there exists'.** The way we use it is, for instance, to rewrite 'There is at least one child in the class.' as'($\exists$ x in U)p(x)',

where p(x) is the sentence 'x is in the class.' and U is the set of all children.

Now suppose we take the negative of the proposition we have just stated. Wouldn't it be 'There is no child in the class.'? We could symbolize this as 'for all x in U, q(x)' where x ranges over all children and q(x) denotes the sentence 'x is not in the class.', i.e., $q(x) \equiv \sim p(x)$.

We have a **mathematical symbol for the quantifier 'for all', which is '$\forall$'.** So, the proposition above can be written as

'$(\forall x \in U)q(x)$', or '$q(x), \forall x \in U$'.

An example of the use of the existential quantifier is the true statement.

$(\exists x \in \mathbf{R})(x + 1 > 0)$, which is read as 'There exists an x in **R** for which $x + 1 > 0$.'.

Another example is the false statement

$(\exists x \in \mathbf{N})(x - \frac{1}{2} = 0)$, which is read as 'There exists an x in **N** for which $x - \frac{1}{2} = 0$.'.

An example of the use of the universal quantifier is $(\forall x \notin \mathbf{N})(x^2 > x)$, which is read as 'for every x not in **N**, $x^2 > x$.'. Of course, this is a false statement, because there is at least one $x \notin \mathbf{N}$, $x \in \mathbf{R}$, for which it is false.

As you have already read in the example of a child in the class,

$(\forall x \in U)p(x)$ is logically equivalent to $\sim(\exists x \in U)(\sim p(x))$. Therefore,

$\sim(\forall x \in U)p(x) \equiv \sim\sim(\exists x \in U)(\sim p(x)) \equiv (\exists x \in U)(\sim p(x))$.

This is one of the rules for negation that relate $\forall$ and $\exists$. The two rules are

$\sim(\forall \mathbf{x} \in \mathbf{U})\mathbf{p}(\mathbf{x}) \equiv (\exists x \in \mathbf{U})(\sim \mathbf{p}(\mathbf{x}))$, and

$\sim(\exists \mathbf{x} \in \mathbf{U})\mathbf{p}(\mathbf{x}) \equiv (\forall x \in \mathbf{U})(\sim \mathbf{p}(\mathbf{x}))$

Where U is the set of values that x can take.

To Sum up "***a proposition is a specialized statement whereas Predicate is a generalized statement***". To be more specific the propositions use the logical connectives only and the predicates uses logical connectives and quantifiers (universal and existential), both.

Note :$\exists$ is the symbol used for the Existential quantifier and $\forall$ is used for the Universal quantifier.

By interpretation of symbolic logic, we mean assigning the meaning to the symbols of any formal language. By default, the statements do not have any meaning associated with them but when we assign the symbols to them, their meaning is automatically associated with them i.e., TRUE (T) or FALSE (F).

Example 1: Consider the following statements

a) Weather is not hot today and it is windy than yesterday.
b) Kids will go to park only if it is not hot.
c) If kids do not go to park, then they can play near pool.
d) If kids play near pool, then we can offer them a juice.

Let the statements be represented using literals as following:

H: Weather is not hot today
W: It is windy than yesterday
K: Kids will go to park
P: They can play near pool.
J: We can offer them juice.

The given statements can be interpreted as:

a) H ∧W
b) K -> ~H
c) ~K -> P
d) P -> J

These interpretations may be verified with the help of truth tables.

4.11 INFERENCING IN PREDICATE LOGIC

In general, we are given a set of arguments in predicate logic. Now, using the rules of inference, we can deduce other arguments (or predicates) based on the given arguments (or predicates). This process is known as entailment as we entail new arguments (or predicates). The inference rules you learned in MCS-212 and also in section 4.7 above, of this unit are also applicable here for the process of entailment or making inferences. Now, with the help of the following example we will learn how the rules of inference, discussed above, can be used to solve the problems.

Example : There is a village that consists of two types of people – those who always tell the truth, and those who always lie. Suppose that you visit the village and two villagers A and B come up to you. Further, suppose

A says, "B always tells the truth," and

B says, "A and I are of opposite types".

What types are A and B ?

Solution: Let us start by assuming A is a truth-teller.

∴ What A says is true.

∴ B is a truth-teller.

∴ What B says is true.

∴ A and B are of opposite types.

This is a contradiction, because our premises say that A and B are both truth-tellers.

$\therefore$ The assumption we started with is false.

$\therefore$ A always tells lies.

$\therefore$ What A has told you is lie.

$\therefore$ B always tells lies.

$\therefore$ A and B are of the same type, i.e., both of them always lie.

Let us now consider the problem of showing that a statement is false. I.e.,**Counter examples :** A common situation in which we look for counterexamples is to disprove statements of the form p → q needs to be an example where p ∧ ~ q. Therefore, a counterexample to p → q needs to be an example where p ∧ ~ q is true, i.e., p is true and ~ q is true, i.e., the hypothesis p holds but the conclusion q does not hold.

For instance, to disprove the statement 'If n is an odd integer, then n is prime.', we need to look for an odd integer which is not a prime number. 15 is one such integer. So, n = 15 is a counterexample to the given statement.

Notice that a **counter example to a statement p proves that** p is false, i.e., ~ **p is true**.

Example: Following statements are to be Symbolized and thereafter construct a proof for the following valid argument:

(i) If the BOOK_X is literally true, then the Earth was made in six days.
(ii) If the Earth was made in six days, then carbon dating is useless and Scientists/Researchers are liars.
(iii) Scientists/Researchers are not liars.
(iv) The BOOK_X is literally true, Hence
(v) God does not exist.

Solution: Let us symbolize as follows:

B : BOOK_X is literally true
E : The Earth was created in six days
C : Carbon_dating techniques are useless
S : Scientists/Researchers are frauds
G : God exists

Therefore, the statements in the given arguments are symbolically represented as :

(i) $B \rightarrow E$

(ii) $E \rightarrow C \wedge S$

(iii) $\sim$ S

(iv) B

(v) $\sim$ G (*to show*)

Using ModusPonens on (i) and (iv), we get expression (vi) E

Using ModusPonens on (ii) & (vi) we get expression (vii) $C \wedge S$

Using Simplification on (vii), we get expression (viii) S

Using Addition on (viii), we get expression (ix) $S \vee \sim G$

Using DisjunctiveSyllogism(D.S.) on (iii) & (ix) we get expression (x) $\sim$ G

The last statement is what is to be proved.

Remarks: (iii) and (viii) are contradicts with each other in the above deduction. In general, if we come across two statements (like S and ~ S) that contradict each other during the process of deduction, we can deduce any statement, even if the statement can never be True in any way. So, we can assume that any statement is true if both S and ~ S have already happened in the process of derivation.

4.12 PROOF SYSTEMS

A sequence of statements that follows logically from the previous set of statements or observations are the mathematical proofs in propositional logic. The last statements in the proof becomes the theorem. This proof system symbolizes the science of valid inference.

There are more than two main styles of proof system for propositional logic, but the two main styles are:

a) *Axiomatic Proof Systems and*

b) *Natural deduction Systems*

Axiomatic Proof Systems: This is a system where conclusion is derived from either the given hypothesis or using assumed premise, which are considered as truth. Such hypothesis or assumed premise is an *axiom*.

Example 1: Show that (α_1 ∨α_2) is a logical consequence of (α_2 ∧α_1) using proof systems.

Proof:

Step 1: $(\alpha_2 \wedge \alpha_1)$ (Premise)

Step 2: α_1 (Simplification, Step 1)

Step 3: α_2 (Simplification, Step 1)

Step 4: $\alpha_1 \vee \alpha_2$ (Addition on either Step 2 or Step 3)

We observe here that each step is either a truth or a logical consequence of previously established truths. In general, there can be more than one proof for establishing a given conclusion.

Thus, the proof system in propositional logic is quite similar to the proofs in mathematics, which follows a step-wise derivation of consequent from the given hypothesis. However, in propositional logic, we use well-formed formulas obtained from literals and connectors, rather than using English statements. We use the rules of inference (Section 4.7) and replacement rules (Section 4.8) to prove our conclusion.

Example 2: Show that $\alpha_1 \rightarrow \sim\alpha_4$ can be derived from the given premises:

$$(\alpha_1 \rightarrow (\alpha_2 \vee \alpha_3)),\ \alpha_2 \rightarrow \sim\alpha_1,\ \alpha_4 \rightarrow \sim\alpha_3$$

Proof:

Step 1: $(\alpha_1 \rightarrow (\alpha_2 \vee \alpha_3))$ (Premise)

Step 2: α_1 (Assumed Premise)

Step 3: $\alpha_2 \vee \alpha_3$ (Modus Ponens, Step 1, 2)

Step 4: $\alpha_2 \rightarrow \sim\alpha_1$ (Premise)

Step 5: $\sim\alpha_2$ (Modus Tollens, Step 2, step 4)

Step 6: α_3 (Disjunctive Syllogism, Step 1, 2, 4)

Step 7: $\alpha_4 \rightarrow \sim\alpha_3$ (Premise)

Step 8: $\sim\alpha_4$ (Modus Tollens, Step 1, 2, 4, 7)

Step 9: $\alpha_1 \rightarrow \sim\alpha_4$ (Step 2, 4)

Hence proved.

The discussion over Natural deduction Systems is given in section 4.13 below

4.13 NATURAL DEDUCTIONS

So far, we have discussed methods, of solving problems requiring reasoning of propositional logic, that were based on

i) Truth-table construction

ii) Use of inference rules,

and follow, directly or indirectly, **natural deduction approach.**

In order to determine whether or not a conclusion C in an argument is valid or invalid based on a given set of facts or axioms $A_1, A_2, \ldots, A_n$, the only thing we currently know is that either a truth table should be constructed for the formula P: $A_1 \wedge A_2 \wedge \ldots \wedge A_n \rightarrow C$,, or this formula should be converted to CNF or DNF by substituting equivalent formulas and simplifying it. There are other possible options available as well. The trouble with these methods, on the other hand, is that as the number n of axioms increases, the formula becomes more complicated (imagine n being equal to 50), and the number of variables involved, say k, also normally increases. When there are k different variables to consider in an argument, the size of the truth table grows to 2^k. For big values of k, the number of rows, denoted by 2^k, approaches an almost unmanageable level. As a result, it is necessary for us to look for alternative approaches that, rather than processing the entire argument as a single formula, process each of the individual formulas A_1, A_2, and C of the argument as well as their derivatives by applying some principles that ensure the validity of the results.

In an earlier section, we introduced eight different inference rules that can be used in propositional logic to help derive logical inferences. The methods of drawing valid conclusions that have been discussed up until this point are examples of an approach to drawing valid conclusions that is called the **natural deduction approach** of making inferences. This is an approach to drawing valid conclusions in which the reasoning system starts the reasoning process from the axioms, uses inferencing rules, and, if the conclusion can be validly drawn, then it ultimately reaches the conclusion that was intended. On the other hand, there is a method of obtaining legitimate conclusions that is known as the **Refutation approach**. This method, which will be covered in the following part, will be addressed.

The normal forms (CNF and DNF) also play a vital role in both Natural deductions and Resolution approach. To understand the normal forms, we need to start with the basic concepts of clauses, literals etc.

Some Definitions: A **clause** is a disjunction of literals. For example, $(E \vee \sim F \vee \sim G)$ is a clause. But $(E \vee \sim F \wedge \sim G)$ is not a clause. A **literal** is either an atom, say A, or its negation, say ~A.

Definition: A formula E is said to be in a **Conjunctive Normal Form (CNF)** if and only if E has the form $E : E_1 \wedge \ldots . \wedge E_n$, $n \geq 1$, where each of $E_1, \ldots ., E_n$ is a **disjunction** of literals.

Definition: A formula E is said to be in **Disjunctive Normal Form (DNF)** if and only if E has the form $E: E_1 \vee E_2 \vee \ldots . E_n$, where each E_i is a **conjunction** of literals.

Examples: Let A, B and C be atoms. Then F:$(\sim A \wedge B) \vee (A \wedge \sim B \wedge \sim C)$ is a formula in a disjunctive normal form.

Example: Again G: $(\sim A \vee B) \wedge (A \vee \sim B \vee \sim C)$ is a formula in Conjunctive Normal Form, because it is a conjunction of the two disjunctions of literals viz of $(\sim A \vee B)$ and $(A \vee \sim B \vee \sim C)$

Example: Each of the following is neither in CNF nor in DNF

(i) $(\sim A \vee B) \vee (A \wedge \sim B \vee C)$
(ii) $(A \rightarrow B) \wedge (\sim B \wedge \sim A)$

Using table of equivalent formulas given above, any valid Propositional Logic formula can be transformed into CNF as well as DNF.

The steps for conversion to DNF are as follows

Step 1: Use the equivalences to remove the logical operators '$\leftrightarrow$' and '$\rightarrow$':

(i) $E \leftrightarrow G = (E \rightarrow g) \wedge (G \rightarrow E)$

(ii) $E \rightarrow G = \sim E \vee G$

Step 2 Remove ~'s, if occur consecutively more than once, using

(iii) $\sim(\sim E) = E$

(iv) Use De Morgan's laws to take '~' nearest to atoms

(v) $\sim(E \vee G) = \sim E \wedge \sim G$

(vi) $\sim(E \wedge G) = \sim E \vee \sim G$

Step 3 Use the distributive laws repeatedly

(vii) $E \vee (G \wedge H) = (E \vee G) \wedge (E \vee H)$

(viii) $E \wedge (G \vee H) = (E \wedge G) \vee (E \wedge H)$

Example : Obtain a disjunctive normal form for the formula $\sim (A \rightarrow (\sim B \wedge C))$.
Consider $A \rightarrow (\sim B \wedge C) = \sim A \vee (\sim B \wedge C)$ *(Using $(E \rightarrow F) = (\sim E \vee F)$)*

Hence, $\sim(A \rightarrow (\sim B \wedge C)) = \sim(\sim A \vee (\sim B \wedge C))$

$= \sim(\sim A) \wedge (\sim(\sim B \wedge C))$ *(Using $\sim(E \vee F) = \sim E \wedge \sim F$)*

$= A \wedge (B \vee (\sim C))$ *(Using $\sim(\sim E) = E$ and*

$\sim(E \wedge F) = \sim E \vee \sim F$

$= (A \wedge B) \vee (A \wedge (\sim C))$ *(Using $E \wedge (F \vee G) = (E \wedge F) \vee (E \wedge G)$)*

However, if we are to obtain CNF of $(\sim A \rightarrow (\sim B \wedge C))$, in the last but one step, we obtain

$\sim(A \rightarrow (\sim B \wedge C)) = A \wedge (B \vee \sim C)$, which is in CNF, because, each of A and

$(B \vee \sim C)$ is a disjunct.

Example: Obtain conjunctive Normal Form (CNF) for the formula: $D \rightarrow (A \rightarrow (B \wedge C))$

Consider

$D \rightarrow (A \rightarrow (B \wedge C))$ (using $E \rightarrow F = \sim E \vee F$ for the inner implication)

$= D \rightarrow (\sim A \vee (B \wedge C))$ (using $E \rightarrow F = \sim E \vee F$ for the outer implication)

$= \sim D \vee (\sim A \vee (B \wedge C))$

$= (\sim D \vee \sim A) \vee (B \wedge C)$ (using Associative law for disjunction)

$= ((\sim D \vee \sim A \vee B) \wedge (\sim D \vee \sim A \vee C)$

The last line denotes the conjunctive Normal Form of $D \rightarrow (A \rightarrow (B \wedge C))$

(Using distributivity of $\vee$ over $\wedge$)

Note: If we stop at the last but one stop, then we obtain $(\sim D \vee \sim A) \vee (B \wedge C) = \sim D \vee \sim A \vee (B \wedge C)$ is a Disjunctive Normal Form for the given formula : $D \rightarrow (A \rightarrow (B \wedge C))$

Sum up of the technique of natural deduction - Apart from constructing truth table to show that the conclusion follows from the given set of premises, another method exists known as *natural deduction.*In this case, we assume that the conclusion is not valid. We consider negated conclusion as a premise along with other given premises. We apply certain implications, equivalences and replacement rules to derive a contradiction to our assumption. Once, we obtain a contradiction, this proves that the given argument is true.

Example 1: Show that Z is a valid conclusion from the premises X,X->Y and Y->Z.

Proof: Step 1: ~Z (Negated conclusion as Premise)

Step 2: X (Premise)

Step 3:X-> Y (Premise)

Step 4: Y (using Step 2,3andModus Ponens)

Step 5: Y -> Z (Premise)

Step 6: Z (using Step 4, 5 and Modus Ponens)

We obtain the conclusion Z from the given premises, which is a contradiction to our assumption ~Z. Thus, the given conclusion is valid.

Example 2: Show that ~P is concluded from RV S, S-> ~Q, P -> Q and R ->~Q.

Proof: Step 1: ~(~P) (Negated conclusion as Premise)

Step 2: P (Step 1, Double negation)

Step 3: P -> Q (Premise)

Step 4: Q (Step 2, 3, Modus Ponens)

Step 5: S -> ~Q (Premise)

Step 6: ~S (Step 4, 5,Modus Tollens)

Step 7: R V S (Premise)

Step 8: R (Step 1, 3, 5, 7, Disjunctive Syllogism)

Step 9: R -> ~Q (Premise)

Step 10: ~Q (Step 1, 3, 5, 7, 9and Modus Ponens)

We obtain ~Q in Step 10and Q in step 4, which is a contradiction. Thus, our assumption is not valid. Hence, the conclusion follows from the set of premises.

4.14 PROPOSITIONAL RESOLUTION

For the most part, there are two distinct strategies that can be implemented in order to demonstrate the correctness of a theorem or derive a valid conclusion from a given collection of axioms:

i) natural deduction
ii) the method of refutation

In the method known as **natural deduction**, one begins with a given set of axioms, applies various rules of inference, and ultimately arrives at a conclusion. This method is strikingly similar to the intuitive reasoning that is characteristic of humans.

When using a **refutation approach**, one begins with the denial of the conclusion that is to be drawn and then proceeds to deduce a contradiction or the word "false." We are able to deduce a contradiction as a result of having presupposed that the conclusion is incorrect; hence, the premise that the conclusion is incorrect is itself incorrect. Therefore, the argument concerning the technique of resolution leads to the correctness of the conclusion. In this part of the article, we will talk about a different method known as the **Resolution Method**, which was proposed by Robinson in 1965 and is based on the **refutation approach**. The Robinson technique, which has served as the foundation for numerous computerised theorem provers, highlights the significance of the method in question.

Propositional resolution is a sound, complete and powerful rule of inference used in Propositional Logic. It is used prove the unsatisfiability of the given set of statements. This is done using a strategy called Resolution Refutation that uses Resolution rule as described below.

Resolution Rule: The rule states given two statements as $\{\alpha_1, \alpha_2, \alpha_3 \ldots \alpha, \alpha_m\}$ and $\{\gamma_1, \gamma_2, \gamma_3 \ldots . \gamma_n\}$,then the conclusion is $\{\alpha_1, \alpha_2, \alpha_3 \ldots \alpha_m, \gamma_1, \gamma_2, \gamma_3, \ldots . \gamma_n\}$.

For example, the statements {A, B} and {C, ~B} leads to the conclusion {A, C}.

***Resolution Refutation*:**

a) Convert all the given statements to Conjunctive Normal Form (CNF).It is also described as AND of ORs'.For eg., $(A \lor B) \land (\sim A \lor B) \land (\sim B \lor A)$ is a CNF.
b) Obtain the negation of the given conclusion
c) Apply the resolution rule until either the contradiction is obtained or the resolution rule cannot be applied anymore.

4.14.1 Clausal Form

Any atomic sentence or its negation is called as a *literal*. A literal or the disjunction of at least two literals is called as the *clausal form* or *clause expression*. Next, a *clause* is defined as the set of literals in the clause form or clause expression. For example, consider two atomic statements represented using the literals X and Y. Their clausal expressions are X, ~X and (X V Y) and the clauses for these expressions are {X}, {~X} and {X, Y}. It is noteworthy that the empty set {} is always a clause as it represents an empty disjunction and hence, is unsatisfiable. Kindly note, conjunctive normal form (CNF) of an expression represents the corresponding clausal form.

Now, we shall first understand certain rules for converting the statements to the clause form as given below.

i) Operator: $(\beta_1 V \beta_2 V \beta_3 V \ldots . V \beta_m) \Rightarrow \{\beta_1, \beta_2, \beta_3, \ldots . \beta_m\}$
$(\beta_1 \land \beta_2 \land \beta_3 \land \ldots . \land \beta_m) \Rightarrow \{\beta_1\}, \{\beta_2\}, \{\beta_3\}, \ldots ., \{\beta_m\}$

ii) Negation: same as Double Negation and De Morgan's Law in section 4.8
iii) Distribution: as in sub-section 4.8
iv) Implications: as Material implication and Material Equivalence (section 4.8)

Example 1: Convert A∧ (B->C) to clausal expression.

Step 1: A ∧ (~B V C) (using rule Material Implication to eliminate ->)

Step 2: {A}, {~B V C} (using rule Operator to eliminate ∧)

Example 2: Derive the Clausal form or Conjunctive normal form of X <-> Y.

Step 1: Replace bi-condition using Material equivalence rule:

$(X \rightarrow Y) \wedge (Y \rightarrow X)$

Step 2: Use Material Implication replacement rule to replace the implication:

$(\neg X \vee Y) \wedge (\neg Y \vee X)$

Example 3: Derive the CNF of $\sim Z \wedge \sim((\sim X) \rightarrow (\sim Y))$

Step 1: Replace implication using Material Implication(MI):

$\sim(\sim Z \wedge (\sim(\sim X) \vee \sim Y))$

Step 2: Use double negation (DN):

$\sim(\sim Z \wedge (X \vee \sim Y))$

Step 3: Apply DeMorgan's Law:

$\sim\sim Z \vee \sim(X \vee \sim Y)$

Step 4: Apply Double Negation:

$Z \vee \sim(X \vee \sim Y)$

Step 5: Apply DeMorgan's Law again:

$Z \vee (\sim X \wedge \sim\sim Y)$

Step 6: Apply Double Negation on ~~Y:

$Z \vee (\sim X \wedge Y)$

Step 7: Lastly, apply Distributive law to obtain CNF:

$(Z \vee \sim X) \wedge (Z \vee Y)$, which is the AND of OR's form.

$(Z \vee \sim X)$, $(Z \vee Y)$ are the clausal forms for the given expression.

4.14.2 Determining Unsatisfiability

If the obtained set of clauses is not satisfiable, then one can derive an empty clause using resolution principle as described above. In other words, to determine whether a set of propositions or premises {P} logically entails a conclusion C, write P ∨ {¬C} in clausal form and try to derive the empty clause as explained in examples below.

Example 1: Given a set of propositions : X -> Y, Y -> Z. Prove X -> Z.

Proof: To prove the conclusion, we add the negation of conclusion i.e ~(X -> Z) to the set of premises, and derive an empty clause.

Step 1: X -> Y (Premise)
Step 2: ~X ∨ Y (Premise, Material Implication)
Step 3: Y -> Z (Premise)
Step 4: ~Y ∨ Z (Premise, Material Implication)
Step 5: ~(X -> Z) (Premise, Negated Conclusion)
Step 6: ~(~X ∨ Z) (Premise, Material Implication)
Step 7: X ∧ ~Z (Premise, DeMorgans)
Step 8: X (Clausal form, Operator)
Step 9: ~Z (Clausal form, Operator)
Step 10: Y (Resolution rule on Premises in Step 2 and Step 8)
Step 11: Z (Resolution rule on Step 10 and Step 4)
Step 12: {} (Conjunction on Step 11 and Step 9)

Thus, the given set of premises entail the conclusion.

Example 2:Use propositional resolution to derive the goal from the given knowledge base.

a) Either it is a head, or Lisa wins.
b) If Lisa wins, then Mary will go.
c) If it is a head, then the game is over.
d) The game is not over.

Conclusion: Mary will go.

Proof: First consider propositions to represent each of the statement in knowledge base.

Let H: It is a head

L: Lisa wins

M: Mary will go

G: Game is over.

Re-writing the given knowledge base using the propositions defined.

a) H ∨ L
b) L -> M
c) H -> G
d) ~G

Conclusion: M

Step 1: H∨L (Premise)
Step 2: L -> M (Premise)
Step 3: ~L ∨ M (Step 4, Material Implication)
Step 4: H -> G (Premise)
Step 5: ~H ∨ G (Step 4, Material Implication)
Step 6: ~G (Premise)
Step 7: ~M (Negated conclusion as Premise)
Step 8: H ∨ M (Resolution principle on Step 1 and 3)
Step 9: M ∨ G (Resolution principle on Step 8 and 5)
Step 10: M (Resolution principle on Step 9 and 6)
Step 11: {} (Sep 10 and 7)

After applying Proof by Refutation i.e., contradicting the conclusion, the problem is terminated with an empty clause ({}). Hence, the conclusion is derived.

Example 3: Show that ~S1 follows from S1 -> S2 and ~(S1 ∧ S2).

Proof:

Step 1: S1 -> S2 (Premise)
Step 2: ~S1 ∨ S2 (Material Implication, Step 1)
Step 3: ~(S1 ∧ S2) (Premise)
Step 4: ~S1 ∨~S2 (De Morgan's, Step 3)
Step 5: ~S1 (Resolution, Step 2, 4)

The resolution mechanism in PL is not used until after the given statements or wffs have been converted into **clausal forms**. To obtain the clasual form of a wff, one must first convert the wff into the Conjuctive Normal Form (CNF). We are already familiar with the fact that a phrase is a formula (and only a formula) of the form: $A_1 \vee A_2 \vee \ldots\ldots \vee A_n$,where A_i might be either any atomic formula or its negation.

The method of resolution is actually generalization of Modus Ponens, whose expression is

$\frac{P, P \rightarrow Q}{Q}$ which can be written in the equivalent form as $\frac{P, \sim P \vee Q}{Q}$ (i.e. by using the relation $P \rightarrow Q \Rightarrow \sim P \vee Q$).

If we are provided that both P and $\sim P \vee Q$ are true, then we may safely assume that Q is also true. This is a straightforward application of a general resolution principle that will be covered in more detail in this unit.

The construction of a truth table can be used to demonstrate the validity of a resolution process (generally). In order to talk about the resolution process, we will first talk about some of the applications of that method.

Example: Let $C_1: Q \vee R$ and $C_2: \sim Q \vee S$ be two given clauses, so that, one of the literals i.e., Q occurs in one of the clauses (in this case C_1)and its negation $(\sim Q)$ occurs in the other clause C_2. Then application of resolution method in this case tells us to take disjunction of the remaining parts of the given clause C_1 and C_2, i.e., to take $C_3: R \vee S$ as **deduction** from C_1 and C_2. Then C_3 is called a **resolvent** of C_1 and C_2.

The two literals Q and $(\sim Q)$ which occur in two different clauses are called **complementary literals**.

In order to illustrate resolution method, we consider another example.

Example: Let us be given the clauses $C_1: \sim S \vee \sim Q \vee R$ and $C_2: \sim P \vee Q$.

In this case, complementary pair of literals viz. Q and $\sim Q$ occur in the two clause C_1 and C_2.

Hence, the resolution method states: Conclude $C_3: \sim S \vee R \vee (\sim P)$

Example: Let us be given the clauses $C_1: \sim Q \vee R$ and $C_2: \sim Q \vee S$

Then, in this case, the clauses do not have any complementary pair of literals and hence,

resolution method cannot be applied.

Example: Consider a set of three clauses

$C_1: R$

$C_2: \sim R \vee S$

$C_3: \sim S$

Then, from C_1 and C_2 we conclude, through resolution:

C_4:S

From C_3 and C_4, we conclude,

C_5:FALSE

However, a resolvent FALSE can be deduced only from an***unstatisfiable set of clauses*.**
Hence, the set of clauses C_1, C_2 and C_3 is an unsatisfiable set of clauses.

Example: Consider the set of clauses

C_1:$R \vee S$
C_2:$\sim R \vee S$
C_3:$R \vee \sim S$
C_4:$\sim R \vee \sim S$
Then, from clauses C_1 and C_2 we get the resolvent
$C_5 : S \vee S = S$

From C_3 and C_4 we get the resolvent

C_6:~S

From C_5 and C_6 we get the resolvent

C_7:FALSE

Thus, again the set of clauses C_1, C_2, C_3 and C_4 is unsatisfiable.

Note: We could have obtained the resolvent FALSE from only two clauses, viz., C_2 and C_3. Thus, out of the given four clauses, even set of only two clauses viz, C_2 and C_3 is unsatisfiable. Also, a superset of any unsatisfiable set is unsatisfiable.

Example: Show that the set of clauses:

C_1:$R \vee S$
C_2:$\sim S \vee W$
C_3:$\sim R \vee S$
C_4:$\sim W$ is unsatisfiable.
From clauses C_1 and C_3 we get the resolvent
C_7:S
From the clauses C_7 and C_2 we get the resolvent
C_8:W
From the clauses C_8 and C_4 we get
C_9:FALSE
Hence, the given set of clauses is unsatisfiable.
Solution of the Problem Using the Resolution Method As was discussed before, the resolution process can also be understood as a refutation approach. The following is an example of a proving technique that can be used to solve problems:

After the symbolic representation of the issue at hand, an additional premise in the form of the negation of the wff, which stands for conclusion, should be added. You can infer either false or a contradiction from this improved set of premises and axioms. If we are able to get to the conclusion that the statement is not true, then the conclusion that was required to be formed is correct, and the issue has been resolved. If, despite our best efforts, we are unable to arrive at the conclusion that the hypothesis is false, then we are unable to determine whether or not the conclusion is correct. As a result, the predicament cannot be solved using the axioms that have been provided and the conclusion that has been drawn.

Let's go on to the next step and apply Resolution Method to the issues we discussed earlier.

Example: If the interest rate goes up, the stock prices might go down. Also, let's say that most people are unhappy when the price of stocks goes down. Let's say that the rate of interest goes up. Show that most people are unhappy and that we can draw that conclusion.

To show the above conclusion, let us denote the statements as follows:

- A : Interest rate goes up,
- S : Stock prices go down
- U : Most people are unhappy

The problem has the following four statements
1) If the interest rate goes up, stock prices go down.
2) If stock prices go down, most people are unhappy.
3) The interest rate goes up.
4) Most people are unhappy. (to conclude)

These statements are first symbolized as wffs of PL as follows:
(1′) $A \rightarrow S$
(2′) $S \rightarrow U$
(3′) A
(4′) U *(to conclude)*
Converting to clausal form, we get
(i) $\sim A \vee S$
(ii) $\sim S \vee U$
(iii) A
(iv) U *(to be concluded)*
As per resolution method, assume (iv) as false, i.e., *assume ~U as initially given statement, i.e., an axiom.*
Thus, the set of axioms in clasual form is:

(i)$\sim A \vee S$
(ii)$\sim S \vee U$
(iii)A
(iv)$\sim U$

Then from (i) and (iii), through resolution, we get the clause
(v) S.

From (ii) and (iv), through resolution, we get the clause

(vi)~S

From (vi) and (v), through resolution we get,

(viii)FALSE

Hence, the conclusion, i.e., (iv)*U:Most people are unhappy* **is valid.**

From the above solution using the resolution method, we might have noticed that clausal conversion is a major step that takes a lot of time after translation to wffs. Most of the time, once the clause form is known, proof is easy to see, at least by a person.

☞ **Check Your Progress 4**

Ques 1. Prove that given set of premises are unsatisfiable:

a) {X, Y}, {~X, Z}, {~X,~Z}, {X, ~Y}

Ques 2. Consider the given knowledge base to prove the conclusion.

KB1. If Mary goes to school, then Mary eats lunch.

KB 2. If it is Friday, then Mary goes to school or eats lunch.

Conclusion: If it is Friday, then Mary eats lunch.

4.15 ANSWERS/SOLUTIONS

Check Your Progress 1

Ques No	Ans	Ques No	Ans
1	No	8	No
2	No	9	No
3	Yes	10	Yes
4	Yes	11	No
5	No	12	No
6	Yes	13	Yes
7	Yes	14	No

Check Your Progress 2

Ques No	Ans	Ques No	Ans
1	Atomic	6	Atomic
2	Compound	7	Compound
3	Compound	8	Compound
4	Compound	9	Compound
5	Atomic	10	Atomic

Check Your Progress 3

1. (~α_1 → (α_1 ∨α_2)) ↔α_2

α_1	α_2	α_1 ∨α_2	~ α_1 → (α_1 ∨α_2)	(~α_1 → (α_1 ∨α_2)) ↔α_2
_F(0)_	_F(0)_	_F(0)_	_F(0)_	_T(1)_
_F(0)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_
_T(1)_	_F(0)_	_T(1)_	_T(1)_	_F(0)_
_T(1)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_

2. (~α_1 ↔ (α_2 ↔α_3)) ∨ (α_3 ∧α_2)

α_1	α_2	α_3	α_2 ↔α_3	(~α_1 ↔ (α_2 ↔ S3)	α_3 ∧α_2	(~α_1 ↔ (α_2 ↔α_3)) ∨ (α_3 ∧α_2)
_F(0)_	_F(0)_	_F(0)_	_T(1)_	_T(1)_	_F(0)_	_T(1)_
_T(1)_	_F(0)_	_F(0)_	_T(1)_	_F(0)_	_F(0)_	_F(0)_
_F(0)_	_F(0)_	_T(1)_	_F(0)_	_F(0)_	_F(0)_	_F(0)_
_T(1)_	_F(0)_	_T(1)_	_F(0)_	_T(1)_	_F(0)_	_T(1)_
_F(0)_	_T(1)_	_F(0)_	_F(0)_	_F(0)_	_F(0)_	_F(0)_
_T(1)_	_T(1)_	_F(0)_	_F(0)_	_T(1)_	_F(0)_	_T(1)_
_F(0)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_
_T(1)_	_T(1)_	_T(1)_	_T(1)_	_F(0)_	_T(1)_	_T(1)_

3. ((α_1 ∧α_2) → α_3) ∨ ~α_4

α_1	α_2	α_1∧α_2	α_3	((α_1 ∧α_2) → α_3	α_4	((α_1 ∧α_2) → α_3) ∨ ~α_4
_F(0)_	_F(0)_	_F(0)_	_F(0)_	_T(1)_	_F(0)_	_T(1)_
_F(0)_	_F(0)_	_F(0)_	_F(0)_	_T(1)_	_T(1)_	_T(1)_
_F(0)_	_F(0)_	_F(0)_	_T(1)_	_T(1)_	_F(0)_	_T(1)_
_F(0)_	_F(0)_	_F(0)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_
_F(0)_	_T(1)_	_F(0)_	_F(0)_	_T(1)_	_F(0)_	_T(1)_
_F(0)_	_T(1)_	_F(0)_	_F(0)_	_T(1)_	_T(1)_	_T(1)_
_F(0)_	_T(1)_	_F(0)_	_T(1)_	_T(1)_	_F(0)_	_T(1)_
_F(0)_	_T(1)_	_F(0)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_
_T(1)_	_F(0)_	_F(0)_	_F(0)_	_T(1)_	_F(0)_	_T(1)_
_T(1)_	_F(0)_	_F(0)_	_F(0)_	_T(1)_	_T(1)_	_T(1)_
_T(1)_	_F(0)_	_F(0)_	_T(1)_	_T(1)_	_F(0)_	_T(1)_
_T(1)_	_F(0)_	_F(0)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_
_T(1)_	_T(1)_	_T(1)_	_F(0)_	_F(0)_	_F(0)_	_T(1)_
_T(1)_	_T(1)_	_T(1)_	_F(0)_	_F(0)_	_T(1)_	_F(0)_
_T(1)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_	_F(0)_	_T(1)_
_T(1)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_

4. ((α_1 → ~ α_2) ↔α_3) → ~ (α_1 V α_1)

α_1	**A_2**	**α_1 → ~α_2**	**α_3**	**~α_1**	**((α_1 → ~α_2) ↔ α_3)**	**((α_1 → ~α_2) ↔α_3) → ~ (α_1 V α_1)**
_F(0)_	_F(0)_	_F(0)_	_F(0)_	_T(1)_	_T(1)_	_T(1)_
_F(0)_	_F(0)_	_F(0)_	_T(1)_	_T(1)_	_F(0)_	_F(0)_
_F(0)_	_T(1)_	_T(1)_	_F(0)_	_T(1)_	_F(0)_	_F(0)_
_F(0)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_	_T(1)_
_T(1)_	_F(0)_	_T(1)_	_F(0)_	_F(0)_	_F(0)_	_T(1)_
_T(1)_	_F(0)_	_T(1)_	_T(1)_	_F(0)_	_T(1)_	_F(0)_
_T(1)_	_T(1)_	_T(1)_	_F(0)_	_F(0)_	_F(0)_	_F(0)_
_T(1)_	_T(1)_	_T(1)_	_T(1)_	_F(0)_	_T(1)_	_T(1)_

Check Your Progress 4

1. For the given set of premises, derivse an empty clause {} using proposition resolution to show that the given set of premises are unsatisfiable.

Step 1: XV Y (Premise)
Step 2: ~X V Z (Premise)
Step 3: ~XV ~Z (Premise)
Step 4: X V ~Y (Premise)
Step 5: X (Resolution on Step 1 and 4)
Step 6: ~X (Resolution on Step 2 and 3)
Step 7:{} (Using Step 5 and 6)

Thus, the given set of premises are unsatisfiable.

2. First introduce notation set for the given knowledge base as:
S: Mary goes to school
L: Mary eats lunch
F: It is Friday
Corresponding knowledge base is:
KB1: S -> L
KB2:F -> (S V L)
Conclusion: F -> L

Proof: Step 1: ~S V L (Premise, Material Implication)
Step 2: ~F V (S V L) (Premise, Material Implication)
Step 3: F (Negated conclusion)
Step 4: ~L (Negated conclusion)
Step 5: (S V L) (Resolution on Step 2 and 3)
Step 6: L (Resolution on Step 1 and 5)
Step 7:{}(Resolution on Step 4 and 6)

4.16 FURTHER READINGS

1. C. L. Liu & D. P. Mohapatra, Elements of Discrete Mathematics: A Computer Oriented Approach, Fourth Edition, 2017, Tata McGraw Hill Education.
2. David J. Hunter, Essentials of Discrete Mathematics, Third Edition, 2016, Jones and Bartlett Publishers.
3. James L. Hein, Discrete Structures, Logic and Computability, Fourth Edition, 2015, Jones and Bartlett Publishers.
4. Kenneth H. Rosen, Discrete Mathematics and Its Applications, Eighth Edition, 2021, Tata McGraw Hill Education.
5. Thomas Koshy, Discrete Mathematics with Applications, 2012, Elsevier Academic Press.