



CHAPTER-2

COMPUTATIONAL THINKING & ALGORITHMS

Introduction

In this chapter, students learn the basic concepts of logic and computational thinking used in computer science. They understand how to solve problems in a clear and systematic way. Students explore computational thinking techniques such as decomposition and step-by-step problem solving. They also study propositional logic, truth values, truth tables, and logical equivalence.

2.1 Introduction to Computational Thinking

Ex.Q.1 Explain computational thinking and discuss its importance in solving problems with suitable examples.

Computational Thinking

Computational thinking is a **way of solving problems in a clear and logical way**. It helps us break a big problem into smaller and simpler parts. Computational thinking means thinking like a problem solver. It focuses on understanding a problem step by step. We look for patterns in problems and solutions. We remove unnecessary details and focus on important information. Then we create clear steps to solve the problem. These steps can be followed by both humans and computers.

Example:

If we want to plan a school trip, we divide the work into smaller tasks such as choosing the place, arranging transport, preparing a budget, and making a list of students.

Importance of Computational Thinking

- Computational thinking helps us solve problems in a logical and organized way.
- It makes complex problems easier to understand and solve.
- It helps us find patterns and focus on important information.
- It improves problem-solving and decision-making skills.
- It helps us create clear step-by-step solutions.
- It saves time and reduces mistakes.
- It is useful in computer science and everyday life. Scientists, engineers, teachers, and many other professionals use computational thinking to solve different types of problems.

Real-life Examples of Computational Thinking

- Making a daily timetable is an example of computational thinking because we divide our daily work into small tasks.
- Cooking a recipe also uses computational thinking because we follow step-by-step instructions.
- Planning a trip requires breaking the work into smaller tasks such as booking tickets, packing bags, and choosing a route.
- Sorting books by subject or size is an example of finding patterns.
- Fixing a mobile phone problem requires logical thinking to find the cause of the problem.
- Managing money needs planning and careful decision making.
- Playing strategy games also uses computational thinking because players think ahead before making decisions.

Computational Thinking in Daily Life

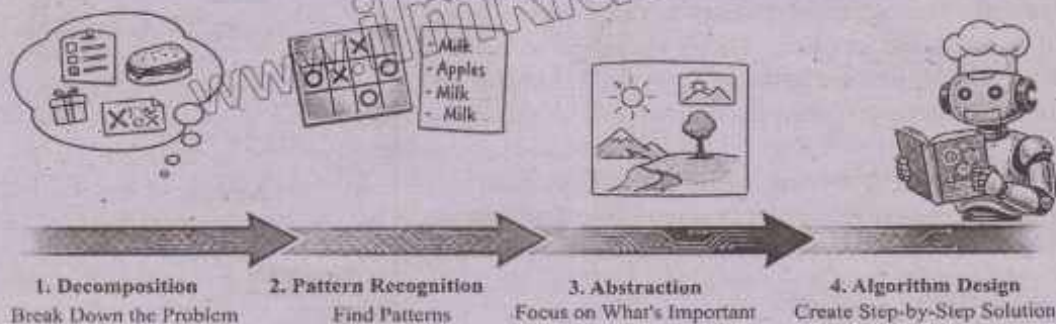


Figure 2.1: Computational Thinking in Daily Life

TIDBIT

➤ Computational thinking is used by scientists, engineers, and even chefs when planning complex tasks.

2.2 Problem Solving Using Computational Thinking

Ex.Q.2 Describe the steps involved in problem solving using computational thinking. Explain decomposition with an example.

Problem Solving

Problem solving is the process of finding a solution to a problem. It involves understanding the problem, planning a solution, and applying the correct steps.

Short Question

What is Problem solving?

Problem Solving Using Computational Thinking

Computational thinking helps us to solve problems in a logical and organized way. It makes problems easier to understand and solve. It reduces mistakes and confusion. It also improves our overall problem-solving ability.

Steps Involved in Problem Solving

1. Identifying a Problem Clearly

Identifying a problem clearly is the first and most important step in problem solving. We should understand what the problem is asking. We must know the given information and the expected result. Asking simple questions helps us understand the problem better. Unnecessary details should be ignored. A clear understanding of the problem leads to the correct solution. This step saves time and effort and makes problem solving easier.

2. Breaking a Problem into Smaller Parts (Decomposition)

Breaking a problem into smaller parts is called decomposition. A big problem can be difficult to solve at once. Smaller parts are easier to understand and manage. Each part can be solved one by one. This reduces confusion and stress. Decomposition helps us organize our thinking and focus on one task at a time. It is widely used in computer science and everyday life.

Example: Planning a school event is a big task. We can divide it into smaller tasks such as making a guest list, arranging the venue, preparing invitations, and managing the budget. Solving each task one by one makes the whole event easier to organize.

3. Creating Step-by-Step Solutions (Algorithms)

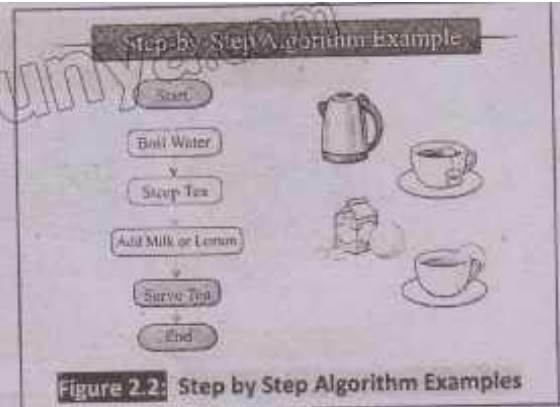
Creating step-by-step solutions helps us to solve problems in an organized way. These steps are known as algorithms. Each step should be clear and easy to follow. The steps must be arranged in the correct order. This reduces errors during problem solving. Anyone can understand and follow these steps. Computers also work by following step-by-step instructions. Good planning helps produce correct results.

4. Evaluating and Improving Solutions

After solving a problem, the solution should be evaluated. Evaluation checks whether the solution is correct. It helps find and correct mistakes. Testing shows whether the solution works properly. If needed, the solution can be improved to make it simpler or faster. This process helps us learn from mistakes and improve our work. Better solutions give better performance.

Example:

A student wants to score better marks in exams. Instead of studying everything at once, the student divides the work into smaller tasks, makes a timetable, studies one subject at a time, and practices past papers. This makes the problem easier to solve.



DO YOU KNOW?

Computers solve problems in the way humans do when they use computational thinking, but much faster!



Additional Multiple Choice Questions

- A way of solving problems in a clear and logical way is called:
A) Problem solving B) Computational thinking C) Algorithm D) Decomposition
- The process of finding a solution to a problem is called:
A) Observation B) Evaluation C) Problem solving D) Pattern finding
- Breaking a problem into smaller parts is called:
A) Evaluation B) Algorithm C) Pattern recognition D) Decomposition
- Creating step-by-step solutions helps us to solve problems in an organized way. These steps are known as:
A) Algorithms B) Variables C) Functions D) Patterns
- Which of the following is an example of computational thinking?
A) Sleeping early B) Planning a trip C) Watching TV D) Reading stories
- Which step comes after creating step-by-step solutions in problem solving?
A) Identifying a problem B) Breaking into parts
C) Evaluating and improving solutions D) Finding patterns
- Which statement best explains decomposition?
A) It creates logical expressions. B) It breaks a big problem into smaller parts.
C) It removes all information. D) It writes computer programs.
- Why should unnecessary details be ignored while identifying a problem?
A) To make the problem longer B) To increase confusion
C) To avoid planning D) To understand the problem clearly
- Evaluation checks whether the solution is:
A) Fast B) Simple C) Organized D) Correct

Answers:

1	B	2	C	3	D	4	A	5	B	6	C	7	B	8	D	9	D
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

2.3 Logic as a Knowledge Representation Framework

Ex.Q.3. What is logic as a knowledge representation framework? Explain its role in reasoning and decision making.

Logic in Computer Science

Logic is the science of true reasoning. It is used to represent facts, rules, and relationships in a clear and organized way. It helps in solving problems and reaching correct conclusions.

In computer science, logic means using **rules to make correct decisions**. It helps computers understand true and false conditions. It allows systems to process information correctly. Logical statements are used in programs and algorithms. These statements control the flow of a program. Logic is the basis of many computer operations. It helps computers to work in a structured and organized way. It is also used in databases, search systems, and software applications to process information accurately.

Example

An ATM checks whether the correct PIN is entered. If the PIN is correct, it allows the user to continue. If the PIN is incorrect, it denies access. This decision is made by using logic.

Role of Logic in Reasoning and Decision Making

Logic plays an important role in reasoning and decision making.

- It helps both computers and humans reach correct conclusions.
- Decisions are made by using facts and logical rules.
- Logic removes confusion and reduces guessing.
- It helps solve problems step by step. Computers use logic to choose the correct action.
- Logic makes decisions accurate and consistent.
- It also helps detect errors and improve the reliability of computer systems.
- Logic is widely used in expert systems, artificial intelligence, robotics, and automated decision-making systems.

Short Question

Describe Role of Logic in Reasoning and Decision Making?

TIDBIT

- Logic gives computers a way to "think" about facts and make decisions, almost like a brain.

CLASS ACTIVITY

- Discuss how logic helps in making decisions.
- Students make simple "if-then" statements about school life.
Example: "If it rains, then we will stay indoors."

Sol:

1. How does logic help in making decisions?

Logic helps us make correct decisions by using facts, conditions, and reasoning. It allows us to think clearly and choose the best action based on a given situation. In computational thinking, logic is used to create rules and instructions for solving problems.

2. Simple "If-Then" Statements about School Life

1. If the teacher gives homework, then students will complete it.
2. If a student is absent, then the student will ask for missed work.
3. If the exam is tomorrow, then students will revise their notes.
4. If the school bell rings, then students will go to their classes.
5. If it is raining, then students will stay indoors.
6. If a student scores good marks, then the student will be rewarded.
7. If the library is open, then students can borrow books.

2.4 Propositional Logic

Ex.Q.4. Explain propositional logic. Discuss simple and compound propositions with examples.

Propositional Logic

Propositional logic is a basic form of logic used in computer science. It deals with statements that can be **either true or false**. These statements are called propositions. Propositional logic helps computers make decisions and solve problems using logical rules.

Short Question

What is Propositional logic?

Proposition

A proposition is a statement that is either **true or false**. It cannot be both true and false at the same time. Questions and commands are not propositions because they do not have a true or false value.

For example, the statement "**The sun rises in the east**" is a proposition because it is true. A proposition must have a clear meaning. It should not be confusing. Propositions are the basic units of logic. They are used to form logical expressions. Understanding propositions is important in logic. A proposition is usually represented by a capital letter such as P, Q, or R.

Example: P = "The sky is blue"

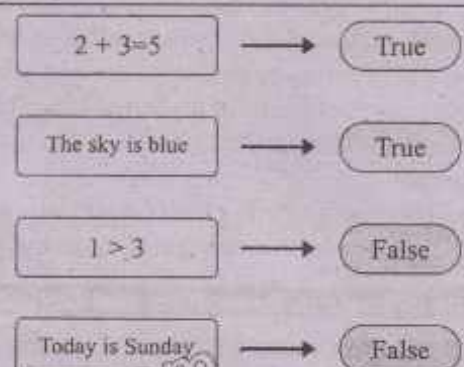


Figure 2.3: Propositional Logic Statements

Simple Propositions

A simple proposition contains only one statement. It cannot be divided into smaller statements.

Example: "It is raining."

Compound Propositions

A compound proposition is formed by joining two or more simple propositions. Logical connectives are used to join them.

Example: "It is raining and it is cold."

Compound propositions are used to express more complex ideas. They are common in logical reasoning. Both simple and compound propositions are important in propositional logic.

Short Question

What are simple propositions and compound propositions? Give example

Logical Connectives: AND, OR, NOT

Logical connectives are words used to join or modify propositions.

- **AND** is used when both statements must be true.
- **OR** is used when at least one statement is true.
- **NOT** is used to reverse the truth of a statement. It makes a true statement false and a false statement true.

Short Question

What are Logical Connectives and why they are used?

Logical connectives are used to create compound propositions. They are commonly used in computer programs and decision-making. They help to make logical expressions clear and easy to understand.

Example

P = "It is raining"

Q = "It is cold"

Then:

P AND Q = "It is raining and it is cold."

P OR Q = "It is raining or it is cold."

NOT P = "It is not raining."

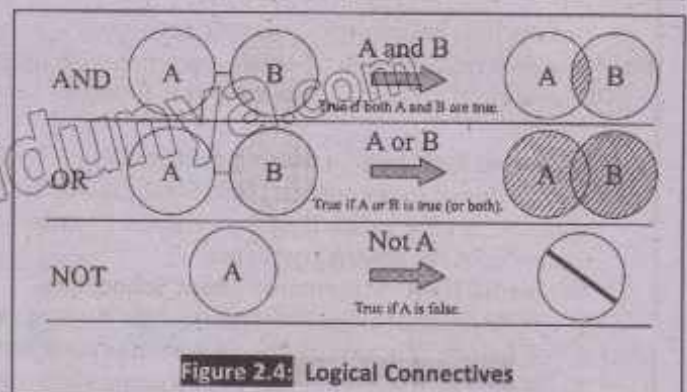


Figure 2.4: Logical Connectives

? DO YOU KNOW?

- Propositional logic is the simplest type of logic, but it forms the base of all computer reasoning!

2.5 Truth Tables

Ex.Q.5 What are truth tables? Explain how truth tables are created and used to evaluate logical expressions.

Truth Tables

A truth table is a table that shows the **truth values of logical statements**. They help us to understand whether a logical expression is true or false. They help us to understand how logical expressions work.

A truth table lists all possible combinations of truth values. It shows the final result for each combination. Truth tables are important in propositional logic. They are widely used in computer science and mathematics. They help check the correctness of logical expressions. Truth tables make logical thinking clear and simple.

A	B	A AND B	A OR B	NOT A
True	True	True	True	False
True	False	False	True	False
False	True	False	True	True
False	False	False	False	True

Figure 2.5: Truth Table for Logical Expressions

Purpose of Truth Tables

- They show how a logical expression behaves.
- They help us find when a statement is true or false.
- Truth tables list all possible combinations of input values.
- They remove confusion in logical reasoning. T
- They also help to compare different logical expressions.
- Truth tables are useful in decision making.
- They are important tools in logic and computer science.

Short Question

Why truth tables are used?

Steps to Create a Truth Table for Logical Expressions

Step 1: Identify the Propositions

Identify all the propositions used in the logical expression. These are usually represented by letters such as P, Q, and R.

Step 2: Assign Truth Values

Give each proposition a truth value of either **True (T)** or **False (F)**. These truth values are used to evaluate the expression.

Step 3: List All Possible Combinations

Write all possible combinations of truth values for the propositions. The number of rows depends on the number of propositions.

Step 4: Apply the Logical Connectives

Evaluate the logical connectives such as **NOT ($\neg$)**, **AND ($\wedge$)**, **OR ($\vee$)**, and **IF-THEN ($\rightarrow$)** one step at a time.

Step 5: Find the Final Result

Calculate the final truth value for each row of the logical expression. Write the result in the last column of the table.

Step 6: Draw the Table Neatly

Arrange all columns and rows in a clear and organized way. A neat truth table is easier to read and understand.

Step 7: Check the Results

Review the completed truth table to make sure all values are correct.

Truth Values of Propositions

Truth values are the possible results of a logical statement. There are two truth values: **True** and **False**. A statement is True if it is correct and False if it is incorrect. Truth values are used in logical expressions and decision-making in programming. Each proposition, such as **P** or **Q**, can have only one of two truth values: **True (T)** or **False (F)**. For n propositions, a truth table contains 2^n possible combinations of truth values. The truth value of a compound proposition depends on the truth values of the simple propositions that form it.

Example: Truth table for **P AND Q**.

P = "It is raining"

Q = "It is cold"

P	Q	P AND Q
T	T	T
T	F	F
F	T	F
F	F	F



Additional Multiple Choice Questions

Logic in Computer Science

1. **Logic is the science of:**
A) Programming B) True reasoning C) Data storage D) Networking
2. **In computer science, logic helps computers to understand:**
A) Pictures only B) True and false conditions C) Internet speed D) File size
3. **Which system uses logic to allow or deny access after checking a PIN?**
A) Printer B) ATM C) Scanner D) Monitor
4. **Logic is widely used in expert systems, artificial intelligence, robotics, and:**
A) Video editing B) Web designing
C) Automated decision-making systems D) Computer games
5. **Which of the following is an example of using logic?**
A) Entering the correct PIN in an ATM B) Charging a mobile phone
C) Printing a document D) Typing a letter

Propositional Logic

6. **Propositional logic deals with statements that can be either:**
A) Long or short B) True or false C) Easy or difficult D) New or old
7. **A statement that is either true or false is called a:**
A) Variable B) Algorithm C) Proposition D) Function
8. **Which of the following is NOT a proposition?**
A) The sun rises in the east. B) It is raining C) Close the door. D) The sky is blue.
9. **A proposition is usually represented by:**
A) Numbers B) special character C) small letter D) capital letter
10. **A compound proposition is formed by joining:**
A) Numbers B) Variables C) Two or more simple propositions D) Algorithms
11. **Which logical connective is true only when both statements are true?**
A) OR B) NOT C) IF D) AND
12. **Which logical connective reverses the truth of a statement?**
A) AND B) OR C) NOT D) IF
13. **Which expression represents "It is raining and it is cold"?**
A) $P \text{ OR } Q$ B) $\text{NOT } P$ C) $P \text{ AND } Q$ D) $P \rightarrow Q$

Truth Tables

14. **A truth table is a table that shows the truth values of:**
A) Numbers B) Programs C) Logical statements D) Pictures
15. **Truth tables help us to understand whether a logical expression is:**
A) Large or small B) True or false C) Fast or slow D) Easy or difficult

16. The first step in creating a truth table is to:
 A) Find the final result B) Apply logical connectives C) Identify the propositions D) Draw the table
17. Which truth values are assigned to propositions?
 A) High and Low B) On and Off C) True and False D) Open and Close
18. For n propositions, a truth table contains:
 A) n rows B) n^2 rows C) 2^n rows D) $n + 2$ rows

Answers:

1	B	2	B	3	B	4	C	5	A	6	B	7	C	8	C	9	D	10	C
11	D	12	C	13	C	14	C	15	B	16	C	17	C	18	C				

2.6 Propositional Equivalence

Ex. Q.6: Explain propositional equivalence. Describe how equivalent logical expressions are identified and why equivalence is important.

Propositional Equivalence

Propositional equivalence is a concept in propositional logic. It explains when two logical expressions have the same meaning. Two expressions may look different but still produce the same result. Propositional equivalence is checked by using truth tables.

If both expressions have the same truth value in every possible case, they are equivalent. The symbol " $\equiv$ " is used to show equivalence. Propositional equivalence makes logical reasoning easier and clearer.

Short Question

What is Propositional equivalence?

The following two propositional expressions are logically equivalent.

Expression 1

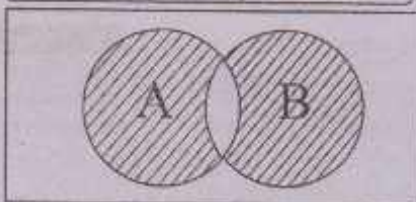
Expression 2

$(A \text{ AND NOT } B) \text{ OR } (B \text{ AND NOT } A)$

$(A \text{ OR } B) \text{ AND } (\text{NOT } A \text{ OR NOT } B)$

Both expressions are true in the same cases (XOR).

Venn Diagram Representation



☒ A AND NOT B ☒ B AND NOT A
☐ Neither A nor B ☒ A AND B (not included)

Truth Table

A	B	$(A \text{ AND NOT } B) \text{ OR } (B \text{ AND NOT } A)$	$(A \text{ OR } B) \text{ AND } (\text{NOT } A \text{ OR NOT } B)$
F	F	F	F
F	T	T	T
T	F	T	T
T	T	F	F

Therefore: $(A \text{ AND NOT } B) \text{ OR } (B \text{ AND NOT } A) \equiv (A \text{ OR } B) \text{ AND } (\text{NOT } A \text{ OR NOT } B)$

Identifying Equivalent Logical Expressions

Equivalent logical expressions are identified by using truth tables. Each expression is tested with all possible truth values. If both expressions produce the same result in every row, they are equivalent. Logical laws can also be used to identify equivalent expressions. These laws help simplify logical expressions. Identifying equivalence makes complex expressions easier to understand.

Importance of Equivalence in Logic

- Equivalence is important because it helps simplify logic problems.
- It allows complex logical expressions to be replaced with simpler ones. This saves time and effort.
- In programming, equivalent expressions can improve code efficiency.
- Equivalence also helps to avoid logical errors. It supports correct reasoning and decision making.
- This concept is useful in both mathematics and computer science.
- Understanding equivalence improves logical thinking.

Example

The logical expressions **NOT (NOT P)** and **P** always have the same truth value. Therefore, they are equivalent. A truth table can be used to verify that both expressions give the same result in every case.

P	NOT P	NOT (NOT P)
T	F	T
F	T	F

Short Question

Describe importance of Propositional equivalence in logic?



TIDBIT

- Equivalent expressions may look different but always give the same result; they help simplify complex

2.7 Propositional Satisfiability

Ex. Q. 7. What is propositional satisfiability? Explain satisfiable and unsatisfiable propositions with examples.

Propositional Satisfiability

Propositional satisfiability is a concept in logic that checks whether a **statement can be true**. It helps us know if a logical expression has at least one situation where it is true. This concept is widely used in computer science, especially in problem solving and programming. It helps test logical statements and supports decision making. Satisfiability is also important for designing algorithms and verifying programs. Understanding satisfiability improves logical reasoning and helps solve complex problems efficiently.

Satisfiable Propositions

A satisfiable proposition is a logical expression that is true for at least one combination of truth values.

Example: $P \vee Q$ is satisfiable because it is true when $P = \text{True}$ or $Q = \text{True}$.

Unsatisfiable Propositions

An unsatisfiable proposition is a logical expression that is false for all combinations of truth values.

Example: $P \wedge \neg P$ is unsatisfiable because it is always false.

Checking Satisfiability

The satisfiability of a proposition depends on its structure and the logical connectives used, such as AND, OR, and NOT. A truth table can be used to check all possible combinations of truth values. If the truth table contains at least one True (T) result, the proposition is satisfiable. If all results are False (F), the proposition is unsatisfiable.

Examples of Satisfiable and Unsatisfiable Expressions

Example 1:

Expression: Bring pizza OR (burger AND do NOT bring soda).

Result: Bring pizza, no soda = satisfiable expression because a solution exists. This expression is true when you bring pizza, or when you bring a burger and do not bring soda.

Expression: You must bring pizza AND you must NOT bring pizza.

Result: No solution exists because both conditions cannot be true at the same time. Therefore, it is an unsatisfiable expression.

Short Question

Describe satisfiable proposition and unsatisfiable proposition? Give example.

Example 2:

Satisfiable Expression: $P \text{ OR } Q$

This expression is true if P is true, Q is true, or both are true. Therefore, it is satisfiable.

Satisfiable Expression: $P \text{ AND } Q$

This expression is true when both P and Q are true. Therefore, it is satisfiable.

Unsatisfiable Expression: $P \text{ AND NOT } P$

This expression can never be true because P cannot be both true and false at the same time. Therefore, it is unsatisfiable.



TIDBIT

➤ A logical expression is satisfiable if there is at least one way to make it true.



CLASS ACTIVITY

Provide several expressions and ask students to classify them as satisfiable or unsatisfiable.

Sol:

Instruction: Check each expression and classify it as Satisfiable or Unsatisfiable.

Expressions:

- | | |
|--|-----------------------|
| 1. P : It is raining.
(It can be true when it rains.) | Answer: Satisfiable |
| 2. $P \text{ AND NOT } P$
(A statement cannot be true and false at the same time.) | Answer: Unsatisfiable |
| 3. $P \text{ OR } Q$
(It is true if at least one statement is true.) | Answer: Satisfiable |
| 4. $P \text{ AND } Q$
(Both statements can be true together.) | Answer: Satisfiable |
| 5. $P \text{ AND NOT } Q$
(P can be true and Q can be false.) | Answer: Satisfiable |
| 6. $P \text{ AND NOT } P \text{ AND } Q$
(P cannot be true and false at the same time.) | Answer: Unsatisfiable |
| 7. If a student studies, then the student passes.
(The condition can be true in some situations.) | Answer: Satisfiable |
| 8. A student is present AND a student is not present.
(Both conditions cannot happen at the same time.) | Answer: Unsatisfiable |
| 9. The computer is on OR the computer is off.
(One of the two conditions can be true.) | Answer: Satisfiable |
| 10. The light is on AND the light is off.
(A light cannot be on and off at the same time.) | Answer: Unsatisfiable |

2.8 Predicate Logic

Q. Explain predicate logic and also differentiate between Propositional Logic and Predicate Logic

Predicate Logic

Predicate logic is an extension of propositional logic. It allows us to express more detailed statements about objects and their properties. In predicate logic, we can talk about individuals, groups, and relationships between them.

Short Question

What is predicate logic?

Predicate logic studies statements that contain predicates and variables. A predicate is a statement that can be true or false depending on the value of a variable. For example, " x is greater than 5" is a predicate because its truth depends on the value of x .

Predicate logic can express statements such as "All students are hardworking" and "Some books are interesting." It is used to represent knowledge in a formal way.

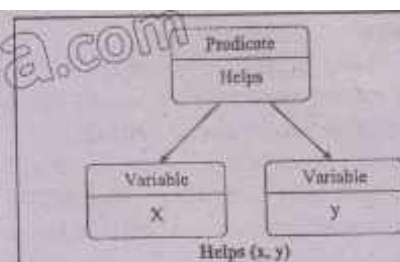


Figure 2.6: Predicate Logic Structure

Difference Between Propositional Logic and Predicate Logic

Propositional Logic	Predicate Logic
Propositional logic deals with simple statements that are either true or false.	Predicate logic uses predicates and variables to express detailed statements.
It cannot describe the properties of objects or the relationships between them.	It can describe the properties of objects and the relationships between different objects.
It is simpler to understand and is used for basic logical expressions.	It is more flexible and powerful because it can represent complex conditions.
It has limited expressive power because it only deals with complete statements.	It has greater expressive power because it can describe many objects and different conditions.
It uses simple propositions without variables.	It uses variables, predicates, and quantifiers to represent detailed information.

Predicate

A predicate is a statement that describes a property of an object or the relationship between objects. It contains one or more variables. A predicate becomes a complete statement when the variables are replaced with specific values.

Example: isEven(x)

This statement is true if x is an even number.

Variables

Variables represent objects or values. They allow us to write general statements.

Example: For all x , $x > 0$

Predicates and variables work together to form predicate logic expressions. They help express complex ideas clearly and formally. Understanding predicates and variables is important for reasoning and problem solving.

Example

Propositional Logic: $A = \text{"Ali is Tall"}$

This statement only tells whether it is true or false. It does not show the subject or the property.

Predicate Logic: Tall(Ali)

This statement clearly shows the subject **Ali** and the property **Tall**. Therefore, predicate logic can express more detailed information than propositional logic.

Short Question

What are variables in predicate logic?



Additional Multiple Choice Questions

Propositional Equivalence

- Propositional equivalence is a concept in:
A) Predicate logic B) Propositional logic C) Flowcharts D) Algorithms
- Propositional equivalence explains when two logical expressions have the same:
A) Size B) Shape C) Meaning D) Color
- Propositional equivalence is checked by using:
A) Algorithms B) Flowcharts C) Truth tables D) Programs
- Two logical expressions are equivalent if they have the same truth value in:
A) One case B) Two cases C) Every possible case D) No case
- The symbol used to show equivalence is:
A) + B) $\equiv$ C) $\times$ D) $\div$
- Which pair of expressions is equivalent?
A) P AND Q, P OR Q B) P, NOT P C) NOT (NOT P), P D) P, P AND Q

Propositional Satisfiability

- Propositional satisfiability checks whether a statement can be:
A) Printed B) Stored C) True D) Deleted
- A satisfiable proposition is true for at least:
A) One combination of truth values B) Two combinations only C) Every combination D) No combination
- An unsatisfiable proposition is false for:
A) One combination B) Two combinations C) Some combinations D) All combinations
- Which expression is satisfiable?
A) $P \wedge \neg P$ B) $P \vee Q$ C) $P \text{ AND NOT } P$ D) False AND True
- Which expression is unsatisfiable?
A) $P \vee Q$ B) $P \text{ AND } Q$ C) $P \wedge \neg P$ D) $P \vee \text{ NOT } Q$
- The satisfiability of a proposition can be checked by using a:
A) Database B) Compiler C) Truth table D) Network
- If a truth table contains at least one True (T), the proposition is:
A) Equivalent B) Unsatisfiable C) Satisfiable D) Invalid
- If all results in a truth table are False (F), the proposition is:
A) Satisfiable B) Unsatisfiable C) Equivalent D) Compound

Predicate Logic

- Predicate logic is an extension of:
A) Boolean algebra B) Flowcharts C) Propositional logic D) Algorithms
- Predicate logic allows us to express more detailed statements about objects and their:
A) Prices B) Properties C) Colors D) Sizes
- A predicate is a statement that can be true or false depending on the value of a:
A) Constant B) Variable C) Number D) Operator
- A predicate becomes a complete statement when the variables are replaced with:
A) Pictures B) Symbols C) Specific values D) Colors
- Which statement is an example of predicate logic?
A) A = "Ali is Tall" B) Tall(Ali) C) It is raining. D) Close the door.
- Predicate logic is more flexible because it can describe:
A) Only one object B) Many objects and different conditions C) Only numbers D) Only pictures

Answers:

1	B	2	C	3	C	4	C	5	B	6	C	7	C	8	A	9	D	10	B
11	C	12	C	13	C	14	B	15	C	16	B	17	B	18	C	19	B	20	B

2.9 Quantifiers in Predicate Logic

Q. What are Quantifiers? Explain the Universal Quantifier, Existential Quantifier, and the use of quantifiers in logical statements.

Quantifiers

Quantifiers are symbols in predicate logic to describe how many objects satisfy a condition. They help us to make general statements or statements about specific objects.

They make predicate logic more powerful than propositional logic. With quantifiers, we can write statements such as "All students are hardworking" or "Some books are interesting." Quantifiers help represent knowledge clearly and accurately.

Universal Quantifier (for all)

The universal quantifier is used when a statement is true for all objects in a given set. It is represented by the symbol $\forall$.

Example: $\forall x, x > 0$

This means "For all x , x is greater than 0."

The universal quantifier is used to make general statements about every object in a set. It is important in mathematics and logic. It helps express general rules and facts clearly.

Existential Quantifier (there exists)

The existential quantifier is used when a statement is true for at least one object in a set. It is represented by the symbol $\exists$.

Example: $\exists x, x$ is even

This means "There exists at least one x that is even."

The existential quantifier is used when a condition is true for one or more objects, but not necessarily for all. It helps express possibilities and specific examples. It is useful in reasoning and problem solving.

Using Quantifiers in Logical Statements

Quantifiers are combined with predicates to form logical statements.

Example: $\forall x, P(x)$

This means $P(x)$ is true for all x .

$\exists x, P(x)$

This means $P(x)$ is true for at least one x .

Quantifiers can also be used together in one statement.

Example: $\forall x \exists y, x < y$

This means For every x , there exists a y such that y is greater than x .

Using quantifiers helps express rules, conditions, and relationships in a clear and logical way.

Example

Statement: $\forall x, \text{Students}(x) \rightarrow \text{Hardworking}(x)$

Meaning: All students are hardworking.

Statement: $\exists x, \text{Book}(x) \wedge \text{Interesting}(x)$

Meaning: There exists at least one interesting book.

Short Question

What are Quantifiers in predicate logic?

Short Question

What is universal quantifier in predicate logic? Give example

Short Question

What is existential quantifier in predicate logic? Give example

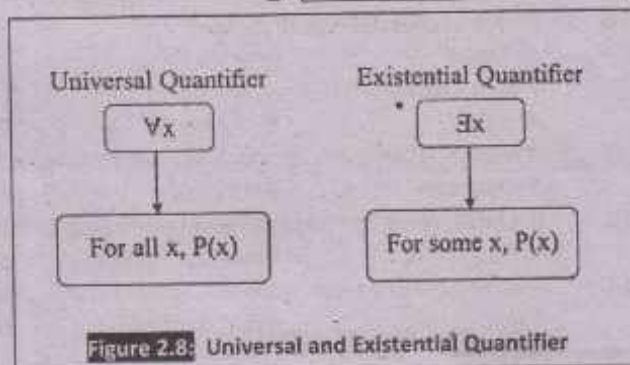


Figure 2.8: Universal and Existential Quantifier



TIDBIT

Quantifiers like "for all" or "there exists" help us express general rules or possibilities clearly.

2.10 Applying Predicate Logic to Real-World Problems

Q. Explain the application of Predicate Logic to real-world problems.

Application of Predicate Logic to Real-World Problems

Predicate logic is used to solve real-world problems by representing knowledge clearly and systematically. It helps describe objects, their properties, and their relationships. Using predicate logic, we can analyze situations, make decisions, and find logical solutions.

Representing Relationships Using Predicates

Predicates are used to show relationships between objects.

Examples:

Borrows(x, y) means student x borrows book y .

Teaches(t, s) means teacher t teaches student s .

These predicates clearly describe the relationships between students, books, and teachers. They can be used to model the working of a library system. This makes it easier to organize information and understand how different objects are connected.

Writing Logical Statements with Quantifiers

Quantifiers help write general rules and conditions.

Example: $\forall x, \text{Student}(x) \rightarrow \text{Borrows}(x, \text{Book})$

This means every student borrows a book.

$\exists y, \text{Book}(y) \wedge \text{Available}(y)$

This means there exists at least one book that is available.

These logical statements describe both general rules and specific situations in a clear and precise way.

Reasoning and Drawing Conclusions

Predicate logic helps us draw conclusions from given facts.

Example:

$\forall x, \text{Student}(x) \rightarrow \text{HasLibraryCard}(x)$

$\text{Student}(\text{Ali})$

From these statements, we conclude:

$\text{HasLibraryCard}(\text{Ali})$

This means Ali has a library card.

Logical reasoning helps in decision making. For example, it can be used to check whether a student is allowed to borrow a book. It helps make decisions correctly and consistently.

Example

In a library system, the rule $\forall x, \text{Student}(x) \rightarrow \text{HasLibraryCard}(x)$ is stored in the system. If $\text{Student}(\text{Ali})$ is true, the system concludes $\text{HasLibraryCard}(\text{Ali})$. Based on this result, Ali is allowed to borrow a book.

DO YOU KNOW?

- Predicate logic can model family trees, social networks, and even computer programs!



CLASS ACTIVITY

Ask students to write rules like "All students must attend class" using universal quantifiers.

Sol: Rule: Universal quantifier ($\forall$) is used to represent statements that are true for all objects in a group.

Examples:

1. Rule: All students must attend class.

Predicate Logic:

$\forall x (\text{Student}(x) \rightarrow \text{AttendClass}(x))$

Meaning: For every x , if x is a student, then x must attend class.

2. Rule: All students must submit assignments.

Predicate Logic:

$\forall x (\text{Student}(x) \rightarrow \text{SubmitAssignment}(x))$

3. Rule: All teachers prepare lessons.

Predicate Logic:

$\forall x (\text{Teacher}(x) \rightarrow \text{PrepareLesson}(x))$

4. Rule: All books in the library are available for students.

Predicate Logic:

$\forall x (\text{Book}(x) \rightarrow \text{Available}(x))$

5. Rule: All computers in the lab are working.

Predicate Logic:

$\forall x (\text{Computer}(x) \rightarrow \text{Working}(x))$

6. Rule: All players follow game rules.

Predicate Logic:

$\forall x (\text{Player}(x) \rightarrow \text{FollowRules}(x))$

7. Rule: All employees have an ID card.

Predicate Logic:

$\forall x (\text{Employee}(x) \rightarrow \text{HasIDCard}(x))$

8. Rule: All students respect teachers.

Predicate Logic:

$\forall x (\text{Student}(x) \rightarrow \text{RespectTeacher}(x))$

Activity Answer:

Students can create similar rules using the format:

$\forall x (\text{Condition}(x) \rightarrow \text{Action}(x))$

Example:

All students read books.

$\forall x (\text{Student}(x) \rightarrow \text{ReadBooks}(x))$

2.11 Logic-Based Reasoning and Inference

Q. What is Logic-Based Reasoning and Inference? Also describe Logical Inference and Deduction using Logical Rules.

Logic-Based Reasoning and Inference

Logic-based reasoning and inference are ways of thinking that use logical rules to draw correct conclusions from known facts. They use the rules of logic to reach conclusions step by step. Reasoning helps to solve problems, make decisions, and verify information. Logical inference ensures that conclusions are correct and reliable.

Understanding this concept improves critical thinking skills. It is important for solving complex real-world problems.

Short Question

What is Logic-based reasoning and inference?

Logical Inference

Logical inference is the process of deriving new facts from known information. It follows logical rules to reach a correct conclusion.

Example: All humans are mortal.

Socrates is a human.

From these facts, we conclude:

Socrates is mortal.

If the given facts are true, the conclusion will also be true. Logical inference is widely used in mathematics, computer science, and artificial intelligence. It helps make decisions based on facts and evidence.

Short Question

What is Logical Inference?
Give example

Deduction Using Logical Rules

Deduction is a type of reasoning that starts with general facts and reaches a specific conclusion. It uses logical rules such as if-then statements.

Example: If it rains, the ground gets wet.

It is raining.

Therefore:

The ground is wet.

Deduction is reliable because the conclusion is correct when the given facts are true. Computers use deduction to solve problems and answer questions. It is an important method in logical reasoning and problem solving.

Example:

A school has the rule: If a student passes all subjects, the student is promoted.

All passes all subjects.

Therefore: Ali is promoted.

Short Question

What is deduction in logical reasoning? Give example



Additional Multiple Choice Questions

Quantifiers and Logic-Based Reasoning and Inference

- Symbols in predicate logic to describe how many objects satisfy a condition are called:
A) Predicates B) Quantifiers C) Variables D) Operators
- A quantifier that is used when a statement is true for all objects in a given set is called:
A) Universal quantifier B) Existential quantifier C) Logical statement D) Predicate
- A quantifier that is used when a statement is true for at least one object in a set is called:
A) Universal quantifier B) Logical rule C) Existential quantifier D) General statement
- Which symbol represents the universal quantifier?
A) $\exists$ B) $\forall$ C) $\rightarrow$ D) $\wedge$
- The universal quantifier is used when a statement is true for:
A) One object B) Some objects C) All objects D) No objects
- Which statement represents the existential quantifier?
A) $\exists x, x$ is even B) $\forall x, x > 0$ C) $P(x)$ is false D) All objects are equal
- A type of reasoning that starts with general facts and reaches a specific conclusion is called:
A) Deduction B) Prediction C) Simulation D) Observation
- If the given facts are true, the conclusion in logical inference will also be:
A) False B) Unknown C) True D) Possible
- The process of deriving new facts from known information is called:
A) Logical inference B) Logical statement C) Predicate logic D) Deduction
- A type of reasoning that starts with general facts and reaches a specific conclusion is called:
A) Inference B) Deduction C) Quantifier D) Predicate

Answers:

- | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|----|---|
| 1 | B | 2 | A | 3 | C | 4 | B | 5 | C | 6 | A | 7 | A | 8 | C | 9 | A | 10 | B |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|----|---|

TEXTBOOK EXERCISE (SOLVED)



MULTIPLE CHOICE QUESTIONS

- Computational thinking is mainly used for:
 - Playing computer game
 - Solving problems in a logical way
 - Writing essays
 - Drawing pictures
- The first step in problem solving is:
 - Writing code
 - Evaluating the solution
 - Identifying the problem
 - Breaking the problem
- Breaking a large problem into smaller parts is called:
 - Abstraction
 - Decomposition
 - Evaluation
 - Automation
- A step-by-step solution to a problem is known as an:
 - Program
 - Flowchart
 - Algorithm
 - Function
- The logic that uses only true or false values is:
 - Predicate logic
 - Fuzzy logic
 - Propositional logic
 - Natural logic
- A compound proposition is:
 - (P)
 - (Q)
 - (P ∨ Q)
 - True
- The purpose of a truth table is to:
 - Write programs
 - Store data
 - Evaluate logical expressions
 - Design algorithms
- The proposition that is true for at least one case is called:
 - Unsatisfiable
 - Satisfiable
 - Invalid
 - Equivalent
- The symbol for the universal quantifier is:
 - $\exists$
 - $\rightarrow$
 - $\wedge$
 - $\forall$
- Predicate logic is mainly used to:
 - Draw diagrams
 - Represent relationships using variables
 - Store numbers
 - Design networks

Answers:

1	B	2	C	3	B	4	C	5	C	6	C	7	C	8	C	9	D	10	B
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	----	---



SHORT QUESTIONS

- What is meant by computational thinking?
 Ans: Computational thinking is a way of solving problems in a clear and logical way. It helps us break a big problem into smaller and simpler parts. Computational thinking means thinking like a problem solver. It focuses on understanding a problem step by step.
- Why is computational thinking important in problem solving?
 Ans:
 - It helps us to solve problems in a logical and organized way.
 - It makes complex problems easier to understand and solve.
 - It helps us to create clear step-by-step solutions.
 - It saves time and reduces mistakes.
- What is decomposition in problem solving?
 Ans: Breaking a problem into smaller parts is called decomposition. A big problem can be difficult to solve at once. Smaller parts are easier to understand and manage. Each part can be solved one by one. This reduces confusion and stress.

4. What is an algorithm?

Ans: An algorithm is a step-by-step set of instructions to solve a problem or perform a task. It takes input, processes it, and produces the required output. Each step of an algorithm is clear, simple, and in the correct order.

5. What is logic in computer science?

Ans: In computer science, logic means using rules to make correct decisions. It helps computers to understand true and false conditions. It allows systems to process information correctly. Logical statements are used in programs and algorithms.

6. What is a proposition?

Ans: A proposition is a statement that is either true or false. It cannot be both true and false at the same time. A proposition must have a clear meaning. It should not be confusing. Propositions are the basic units of logic. They are used to form logical expressions.

7. What are truth values?

Ans: Truth values are the possible results of a logical statement. There are two truth values: True and False. A statement is True if it is correct and False if it is incorrect. Truth values are used in logical expressions and decision-making in programming.

8. What is a truth table?

Ans: A truth table is a table that shows the truth values of logical statements. It helps us to understand whether a logical expression is true or false. They help us to understand how logical expressions work.

9. What is propositional satisfiability?

Ans: Propositional satisfiability is a concept in logic that checks whether a statement can be true. It helps us know if a logical expression has at least one situation where it is true. This concept is widely used in computer science, especially in problem solving and programming.

10. What is a predicate in predicate logic?

Ans: A predicate is a statement that describes a property of an object or the relationship between objects. It contains one or more variables. A predicate becomes a complete statement when the variables are replaced with specific values.



LONG QUESTIONS

All Questions Solved in chapter

- Q.1.** Explain computational thinking and discuss its importance in solving problems with suitable examples.
- Q.2.** Describe the steps involved in problem solving using computational thinking. Explain decomposition with an example.
- Q.3.** What is logic as a knowledge representation framework? Explain its role in reasoning and decision making.
- Q.4.** Explain propositional logic. Discuss simple and compound propositions with examples.
- Q.5.** What are truth tables? Explain how truth tables are created and used to evaluate logical expressions.
- Q.6.** Explain propositional equivalence. Describe how equivalent logical expressions are identified and why equivalence is important.
- Q.7.** What is propositional satisfiability? Explain satisfiable and unsatisfiable propositions with examples.
- Q.8.** Explain predicate logic and quantifiers. Discuss how predicate logic is applied to real-world problems.

ADDITIONAL SHORT QUESTIONS

Computational Thinking

1. Explain how computational thinking is useful in everyday life.

Ans: Computational thinking is useful in many daily activities. It helps us make a daily timetable, plan trips, and cook by following step-by-step instructions. It is also used in managing money, fixing mobile phone problems, and playing strategy games. It makes everyday tasks easier and more organized.

2. List any four real-life examples of computational thinking.

Ans:

- Making a daily timetable.
- Cooking a recipe.
- Planning a trip.
- Sorting books by subject or size.

Problem Solving Using Computational Thinking

3. List the steps involved in problem solving using computational thinking.

Ans:

- Identifying a problem clearly.
- Breaking a problem into smaller parts (Decomposition).
- Creating step-by-step solutions (Algorithms).
- Evaluating and improving solutions.

4. Explain the steps involved in problem solving using computational thinking.

Ans: Problem solving begins by identifying the problem clearly. Then the problem is divided into smaller parts using decomposition. After that, step-by-step solutions called algorithms are created. Finally, the solution is evaluated and improved if needed.

5. Why is identifying a problem clearly the first step in problem solving?

Ans: Identifying a problem clearly helps us understand what the problem is asking. We know the given information and the expected result. Unnecessary details are ignored to avoid confusion. This step saves time and makes problem solving easier.

6. Why are algorithms important in problem solving?

Ans: Algorithms are step-by-step solutions used to solve problems in an organized way. Each step should be clear and arranged in the correct order. This reduces errors and makes the solution easy to follow. Computers also work by following algorithms.

7. Why is it important to evaluate and improve a solution?

Ans: Evaluation checks whether the solution is correct. It helps find and correct mistakes. If needed, the solution can be improved to make it simpler or faster. This process helps produce better results and improves future problem solving.

Logic in Computer Science

8. Why is logic important in reasoning and decision-making?

Ans: Logic helps both computers and humans reach correct conclusions. It removes confusion and reduces guessing. It helps solve problems step by step and makes decisions accurate and consistent. It also helps detect errors and improve the reliability of computer systems.

9. List any four areas where logic is used.

Ans:

- Databases.
- Search systems.
- Software applications.
- Artificial Intelligence (AI).

10. How is logic used in computer systems? Explain with an example.

Ans: Logic is used in programs, databases, search systems, and software applications to process information correctly. It helps computers choose the correct action. For example, an ATM checks whether the correct PIN is entered. If the PIN is correct, it allows access; otherwise, it denies access.

11. List the logical connectives used in propositional logic.

Ans: i) AND ii) OR iii) NOT

Propositional Logic

12. Differentiate between simple proposition and compound proposition.

Simple Proposition	Compound Proposition
A simple proposition contains only one statement.	A compound proposition is formed by joining two or more simple propositions.
It cannot be divided into smaller statements.	Logical connectives are used to join the statements.

Truth Tables

13. Describe the steps to create a truth table.

Ans: First, identify the propositions used in the logical expression. Then assign True (T) or False (F) values and list all possible combinations. After that, apply the logical connectives and calculate the final truth value for each row. Finally, draw the table neatly and check all the results for correctness.

Propositional Equivalence

14. How are equivalent logical expressions identified?

Ans: Equivalent logical expressions are identified by using truth tables. Each expression is tested with all possible truth values. If both expressions produce the same result in every row, they are equivalent. Logical laws can also be used to identify equivalent expressions.

Propositional Satisfiability

15. Differentiate between satisfiable and unsatisfiable propositions.

Satisfiable Proposition	Unsatisfiable Proposition
A satisfiable proposition is true for at least one combination of truth values.	An unsatisfiable proposition is false for all combinations of truth values.
Example: $P \text{ OR } Q$ is satisfiable because it is true when P or Q is true.	Example: $P \text{ AND NOT } P$ is unsatisfiable because it can never be true.

16. How can we check whether a proposition is satisfiable?

Ans: A truth table is used to check all possible combinations of truth values. If the truth table contains at least one True (T) result, the proposition is satisfiable. If all results are False (F), the proposition is unsatisfiable. This helps test logical expressions correctly.

Predicate Logic

17. Differentiate between propositional logic and predicate logic.

Propositional Logic	Predicate Logic
Propositional logic deals with simple statements that are either true or false.	Predicate logic uses predicates and variables to express detailed statements.
It cannot describe the properties of objects or the relationships between them.	It can describe the properties of objects and the relationships between different objects.
It is simpler to understand and is used for basic logical expressions.	It is more flexible and powerful because it can represent complex conditions.
It uses simple propositions without variables.	It uses variables, predicates, and quantifiers to represent detailed information.

18. List the main components of predicate logic.

Ans: i) Predicates, ii) Variables, iii) Quantifiers.

Quantifiers

19. List the types of quantifiers.

Ans: i) Universal Quantifier ($\forall$), ii) Existential Quantifier ($\exists$)

20. What is the universal quantifier?

Ans: The universal quantifier is used when a statement is true for all objects in a given set. It is represented by the symbol $\forall$. It helps express general rules and facts clearly. It is important in mathematics and logic.

21. What is the existential quantifier?

Ans: The existential quantifier is used when a statement is true for at least one object in a set. It is represented by the symbol $\exists$. It helps express possibilities and specific examples. It is useful in reasoning and problem solving.

22. How are quantifiers used in logical statements?

Ans: Quantifiers are combined with predicates to form logical statements. They help express general rules, conditions, and relationships. They make logical statements clear and precise. Quantifiers improve logical reasoning and knowledge representation.

Applying Predicate Logic to Real-World Problems

23. How is predicate logic applied to real-world problems?

Ans: Predicate logic is used to represent real-world problems in a clear and organized way. It describes objects, their properties, and their relationships. It helps analyze situations, make decisions, and find logical solutions. It represents knowledge clearly and systematically.

24. How are relationships represented using predicates?

Ans: Predicates are used to show relationships between objects. They describe how different objects are connected with each other. This helps organize information in a clear way. It also makes real-world systems easier to understand.

25. How do quantifiers help in writing logical statements?

Ans: Quantifiers help write general rules and specific conditions. They are combined with predicates to form logical statements. These statements describe both general rules and particular situations. This makes logical expressions clear and accurate.

26. How does predicate logic help in reasoning and drawing conclusions?

Ans: Predicate logic helps draw conclusions from given facts. It uses logical rules to reach correct decisions. It helps solve problems in a logical way. This makes decision making correct and consistent.

Logic-Based Reasoning and Inference

27. List the two main methods of logical reasoning.

Ans: i) Logical Inference, ii) Deduction.

28. List the uses of logic-based reasoning and inference.

Ans:

- Solving problems.
- Making decisions.
- Verifying information.
- Drawing correct conclusions.
- Improving critical thinking.

CONCEPTUAL SHORT QUESTIONS

Computational Thinking

1. A school wants to organize a science exhibition. Which computational thinking technique should be used first and why?

Ans: The school should first use decomposition. The work should be divided into smaller tasks such as making a guest list, arranging the venue, and preparing displays. Each task can then be completed one by one. This makes the event easier to manage.

Logic in Computer Science

2. A student enters the correct password to log in to the college portal. How does logic help the computer make a decision?

Ans: The computer checks whether the entered password is correct. If it matches the stored password, access is allowed. If it does not match, access is denied. This decision is made using logical rules.

3. A shopping website checks whether an online payment is successful before confirming an order. How is logic used in this situation?

Ans: The website checks the payment status. If the payment is successful, the order is confirmed. If the payment fails, the order is not confirmed. Logic helps the system make the correct decision.

Propositional Logic and Truth Tables

4. A smart door opens only when the correct card is scanned and the correct PIN is entered. Which logical connective is used? Explain.

Ans: The AND logical connective is used. Both conditions must be true before the door opens. If one condition is false, the door remains locked. This ensures security.

Propositional Equivalence

5. A programmer replaces a complex logical expression with a simpler one that gives the same result. Which concept is being used?

Ans: The programmer is using propositional equivalence. Equivalent expressions always produce the same truth values. This makes the program easier to understand. It also improves code efficiency.

Propositional Satisfiability

6. A school rule says, "Students must wear the uniform and must not wear the uniform at the same time." Is this satisfiable or unsatisfiable? Why?

Ans: This is an unsatisfiable proposition. A student cannot wear and not wear the uniform at the same time. Both conditions cannot be true together. Therefore, no solution exists.

7. A canteen rule says, "You may buy tea or juice." Is this satisfiable? Explain.

Ans: Yes, it is satisfiable. The expression is true if a student buys tea, juice, or both. At least one condition can be true. Therefore, a valid solution exists.

Predicate Logic and Quantifiers

8. A school rule states, "Every student must wear an ID card." Which quantifier should be used? Explain.

Ans: The universal quantifier ($\forall$) should be used. The rule applies to every student in the school. It represents a general statement. Therefore, the universal quantifier is correct.

9. A library has at least one science book available. Which quantifier should be used? Explain.

Ans: The existential quantifier ($\exists$) should be used. It means at least one science book is available. The statement does not apply to every book. Therefore, the existential quantifier is suitable.

10. A library system stores the rule: "Every student has a library card." If Ali is a student, what conclusion will the system make?

Ans: The system will conclude that Ali has a library card. This conclusion is based on the stored rule. Predicate logic applies the rule to Ali. This helps the system make a correct decision.

Logic-Based Reasoning and Inference

11. A school rule says, "If a student has 75% attendance, the student can take the exam." Ahmed has 80% attendance. What conclusion can be drawn?

Ans: Ahmed can take the exam. The condition of the rule is satisfied. Deduction is used to reach this conclusion. The decision is based on the given facts.

12. A company rule says, "All permanent employees receive medical insurance." Sara is a permanent employee. What conclusion can be made?

Ans: Sara receives medical insurance. This conclusion follows the company rule. Logical inference is used to derive the new fact. The conclusion is correct because the given facts are true.

13. A school wants to represent the rule "Every student has a roll number." Which type of logic is most suitable and why?

Ans: Predicate logic is the most suitable. It represents objects, their properties, and relationships. It can express rules about all students clearly. It is more powerful than propositional logic.

14. The statement "x is less than 20" becomes true or false only after assigning a value to x. What is this called?

Ans: It is called a predicate. A predicate contains a variable. Its truth depends on the value assigned to the variable. It can be either true or false.

15. A company wants to express the rule "Every employee must wear an ID card." Which quantifier should be used?

Ans: The universal quantifier ($\forall$) should be used. It represents statements that are true for every object in a set. It is used to express general rules. It applies to all employees.

TEST YOUR SKILLS

Total Marks: 40

Time: 1:30 Hour

Q.1 MULTIPLE CHOICE QUESTIONS

8 x 1 = 8

1. Breaking a problem into smaller parts is called:
A) Evaluation B) Algorithm C) Pattern recognition D) Decomposition
2. A statement that is either true or false is called a:
A) Variable B) Algorithm C) Proposition D) Function
3. Which expression represents "It is raining and it is cold"?
A) $P \text{ OR } Q$ B) $\text{NOT } P$ C) $P \text{ AND } Q$ D) $P \rightarrow Q$
4. Which truth values are assigned to propositions?
A) High and Low B) On and Off C) True and False D) Open and Close
5. The symbol used to show equivalence is:
A) $+$ B) $\equiv$ C) $\times$ D) $\div$
6. The symbol for the universal quantifier is:
a) $\exists$ b) $\rightarrow$ c) $\wedge$ d) $\forall$
7. The proposition that is true for at least one case is called:
a) Unsatisfiable b) Invalid c) Satisfiable d) Equivalent
8. The process of deriving new facts from known information is called:
A) Logical inference B) Logical statement C) Predicate logic D) Deduction

Q.2 SHORT QUESTIONS:

Note: Attempt any 8 short questions

8 x 2 = 16

- i. Why is computational thinking important in problem solving?
- ii. Describe Role of Logic in Reasoning and Decision Making?
- iii. What are Logical Connectives and why they are used?
- iv. What are truth values?
- v. What is a predicate in predicate logic?
- vi. What is logic in computer science?
- vii. Why truth tables are used?
- viii. Describe Importance of Propositional equivalence in logic?
- ix. What is universal quantifier in predicate logic? Give example
- x. What is Logical Inference? Give example

LONG QUESTIONS:

Note: Attempt any 2 long questions

2 x 8 = 16

3. Describe the steps involved in problem solving using computational thinking. Explain decomposition with an example.
4. What is propositional satisfiability? Explain satisfiable and unsatisfiable propositions with examples.
5. Explain predicate logic and quantifiers. Discuss how predicate logic is applied to real-world problems.