SNS COLLEGE OF TECHNOLOGY

(Autonomous)

MCA- Internal Assessment –I (Feb 2024)

Academic Year 2023-2024(ODD) / Sixth Semester

19CSE303 – Artificial Intelligence

A

Time: $1^{1/2}$ Hours **Maximum Marks: 50**

Answer All Questions

Answer Key

PART - A (5 x 2 = 10 Marks)

CO

1 **Define an agent and explain the key components of a simple reflex agent.**

An agent is anything that can perceive its environment and take actions that affect that environment.

It has two key components: Percepts**,** Condition-action rules

CO1 **Und**

2 **Compare depth-first and breadth-first search in terms of completeness and optimality**

- Completeness:
  - DFS: Complete for finite graphs (guarantees finding a solution if it exists)
  - BFS: Complete for all graphs (finite or infinite)
- Optimality:
  - DFS: Not optimal in most cases (may explore unnecessary paths)
  - BFS: Optimal for finding shortest paths in unweighted graphs

CO1 **Ana**

3 **Give one advantage and one disadvantage of using beam search compared to best-first search.**

Advantage: Beam search uses less memory than best-first search by only considering a limited number of best options (beam width) at each step.

Disadvantage: Beam search might miss the optimal solution if it's pruned during the selection process.

CO1 **Eva**

4 **Describe the first-order logic model**

Predicate logic, or first order predicate logic, is another name for first order logic. First order logic is a powerful language that can express the relationship between objects and develop information about them more easily.

CO2 **Und**

5 **Define what is meant by a logical agent. Give an example.**

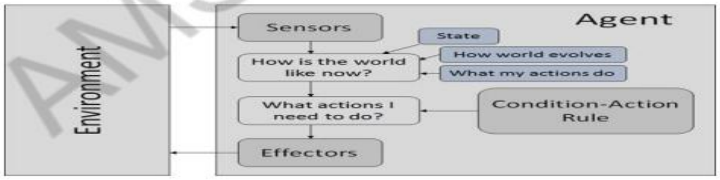
A logical agent is an entity that can reason about its environment and take actions based on those conclusions. For example, a thermostat is a simple logical agent. It perceives the temperature (senses its environment), compares it to a set point (uses logic), and turns the heating/cooling on or off (takes an action).

CO2 **Und**

PART - B (2 x 13 = 26 Marks, 1 X 14 = 14 Marks)

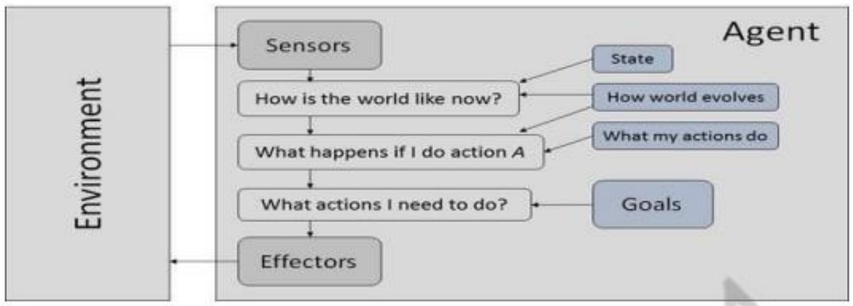
6 **(a)** **Analysis, the structure of different intelligent agents.** **CO1** **Ana**

Condition-Action Rule – It is a rule that maps a state (condition) to an action.




```
Procedure : SIMPLE - REFLEX - AGENT
Input : Percept
Output : An action.
Static : Rules, a set of condition - action rules.
1.     State ← INTERPRET - INPUT (percept)
2.     rule ←  RULE - MATCH (state, rules)
3.     action ← RULE - ACTION (rule)
4.     return action.
```

Goal – It is the description of desirable situations.

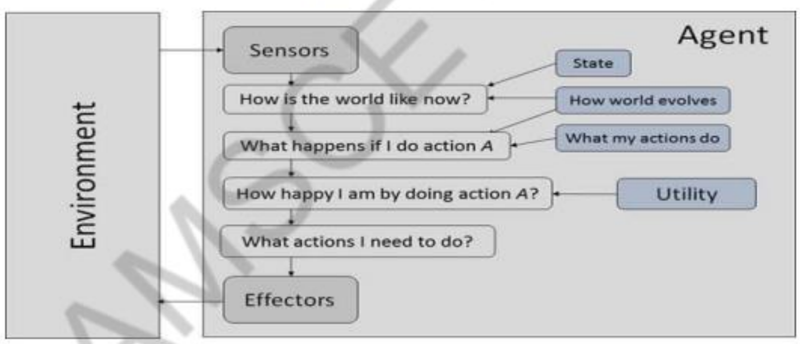


Example: **Searching solution for 8-queen puzzle**.

Utility Based Agents

They choose actions based on a preference (utility) for each state. Goals are inadequate when –

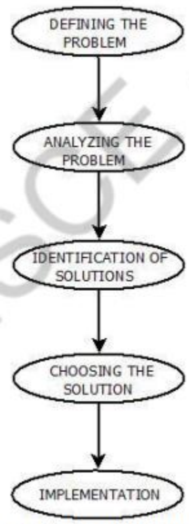
- There are conflicting goals, out of which only few can be achieved.
- Goals have some uncertainty of being achieved and you need to weigh likelihood of success against the importance of a goal.
- Example: Millitary planning robot which provides certain plan of action to be taken.



(Or)

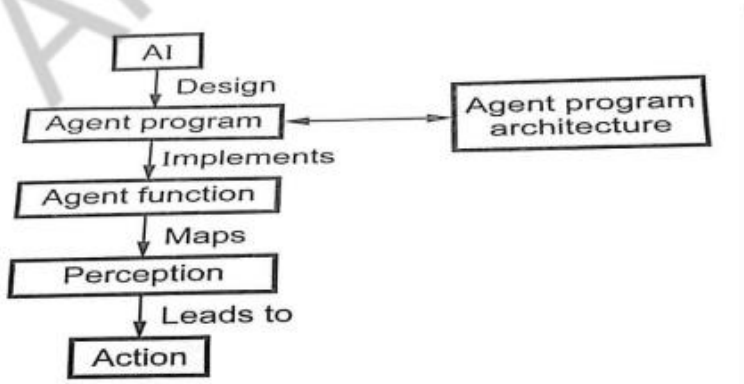
(b) **How a problem is formally defined? List down the components of it?** **CO1** **App**

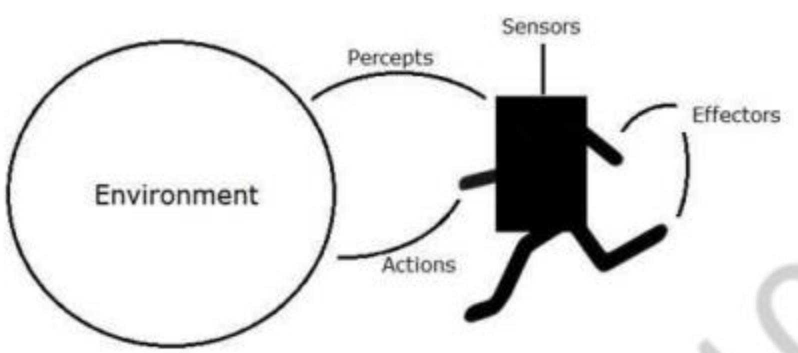
Problems are the issues which comes across any system. A solution is needed to solve that particular problem.
The process of solving a problem consists of five steps. These are:



Problem Solving in Artificial Intelligence

Following diagram illustrated the agent action process, a specified by architecture.





7	(a)	**Compare the propositional Vs. first order inference with examples.** Propositional logic deals with statements (propositions) as true or false, without considering the "what" or "who". Imagine building blocks - you can combine them (AND, OR, NOT) but not describe them. First-order logic (FOL) builds on this by introducing objects, variables, and quantifiers ("all" or "some"). It allows reasoning about things and	**CO1**	**App**

their relationships. Think of FOL like LEGOs - you can build things and describe them.

Here's an example:

- Propositional Logic: "It is raining (R) OR I will go for a walk (W)" (True or False)
- First-Order Logic: "For all x (if x is a cat (C(x)) then x has fur (F(x)))"

(Or)

(b) **Analysis the concept of inference in propositional logic, using an example.**

Inference in Propositional Logic

Inference in propositional logic is the process of deriving new logical statements (conclusions) from existing ones (premises) based on a set of predefined rules. These rules guarantee that if the premises are true, then the conclusion must also be true. It allows us to reason logically and extract new knowledge from a given set of propositions.

Components of Inference

- Propositions (p, q, r): Basic statements that can be true or false.
- Connectives (∧, ∨, ¬, →): Operators that combine propositions to form more complex statements.
- Premises: The starting statements we base our reasoning on.
- Conclusion: The statement we derive based on the premises and inference rules.
- Inference Rules: A set of well-defined rules that dictate how to move from premises to conclusions.

Example: Rain and Slippery Roads

Let's analyze the following scenario:

- Premise 1 (p): It is raining today.
- Premise 2 (q): When it rains, the roads are slippery.

CO2 **Ana**

We want to infer whether the roads are slippery today based on these premises.

Here, we can use the modus ponens rule:

- Modus Ponens: If P implies Q (P → Q), and P is true, then Q must be true.

In our example:

- P: It is raining today (premise 1).
- Q: The roads are slippery (premise 2).

Since "When it rains, the roads are slippery" can be rewritten as "If it is raining today, then the roads are slippery" (P → Q), and it is raining today (premise 1), then according to modus ponens, the roads must be slippery today (conclusion).

Importance of Inference

Inference is crucial in propositional logic because it allows us to:

- Draw well-founded conclusions from existing knowledge.
- Analyze the logical relationships between propositions.
- Build complex arguments and proofs.
- Develop logical reasoning skills applicable in various fields like computer science, mathematics, and philosophy.

8 (a) Enumerate Classical "Water jug Problem". Evaluate the state space for this problem and also give the solution.

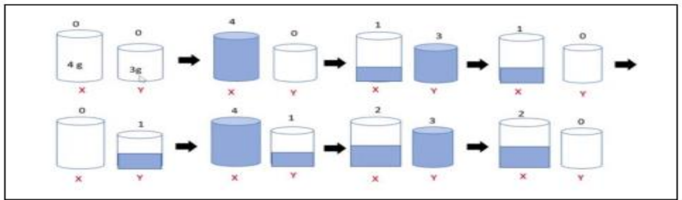


Figure 1.15 Solution

Table 1.2

Solution

S.No.	Gallons in 4-gel jug(x)	Gallons in 3-gel jug (y)	Rule Applied
1.	0	0	Initial state
2..	4	0	1. Fill 4
3	1	3	6. Poor 4 into 3 to fill
4.	1	0	4. Empty 3
5.	0	1	8. Poor all of 4 into 3
6.	4	1	1. Fill 4
7.	2	3	6. Poor 4 into 3

- 4-gallon one and a 3-gallon Jug

CO1 Eva

- No measuring mark on the jug.
- There is a pump to fill the jugs with water.
- How can you get exactly 2 gallon of water into the 4-gallon jug?

Sr.	Current State	Next State	Descriptions
1	(x,y) if x < 4	(4,y)	Fill the 4 gallon jug
2	(x,y) if x < 3	(x,3)	Fill the 3 gallon jug
3	(x,y) if x > 0	(x – d, y)	Pour some water out of the 4 gallon jug
4	(x,y) if y > 0	(x, y – d)	Pour some water out of the 3 gallon jug
5	(x,y) if y > 0	(0, y)	Empty the 4 gallon jug
6	(x,y) if y > 0	(x 0)	Empty the 3 gallon jug on the ground
7	(x,y) if x + y > = 4 and y > 0	(4, y – (4 – x))	Pour water from the 3 gallon jug into the 4 gallon jug until the 4 gallon jug is full
8	(x,y) if x + y > = 3 and x > 0	(x – (3 – x), 3)	Pour water from the 4 gallon jug into the 3 gallon jug until the 3 gallon jug is full
9	(x,y) if x + y < = 4 and y > 0	(x + y, 0)	Pour all the water from the 3 gallon jug into the 4 gallon jug
10	(x,y) if x + y < = 3 and x > 0	(0, x + y)	Pour all the water from the 4 gallon jug into the 3 gallon jug
11	(0, 2)	(2, 0)	Pour the 2 gallons from 3 gallon jug into the 4 gallon jug
12	(2, y)	(0, y)	Empty the 2 gallons in the 4 gallon jug on the ground

(Or)

(b) You are trying to solve a puzzle where you need to get from an initial state to a goal state. There are multiple possible states and moves CO2 Eva

between states, but no information about which states are closer to the goal. Which uninformed search strategy would you use that expands the fewest nodes? Explain your choice.

In the case study you described, where you want to find the solution to a puzzle with no guidance about proximity to the goal, the best uninformed search strategy to use is Breadth-First Search (BFS). Here's why:

- BFS prioritizes exploring all neighboring states of the current state before moving on to deeper levels. This systematic exploration ensures that all states at a given distance from the starting point are considered before going further.
- Guaranteed to find the shortest path: As long as a solution exists, BFS will eventually explore the level containing the goal state. Since it expands all neighbors at a specific depth before moving deeper, the first solution encountered will be the one with the fewest number of moves from the starting point.
- Unaffected by misleading branches: Unlike Depth-First Search (DFS) which can get stuck going down long, irrelevant branches, BFS explores all possibilities at a given depth. This prevents it from wasting effort on dead ends that might appear closer to the goal initially.

While BFS might seem computationally expensive for very large search spaces, its guarantee of finding the shortest path and systematic exploration make it ideal for this scenario where you don't have any information about which states are closer to the goal.