DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNIT- 1- ARTIFICIAL INTELLIGENCE- 23CS402

UNIT 1

1.INTRODUCTIONTO ARTIFICIAL INTELLIGENCE

Introduction–Definition –Future of Artificial Intelligence –Characteristics of Intelligent Agents– Typical Intelligent Agents –Problem Solving Approach to Typical AI problems.

1 INTRODUCTION

INTELLIGENCE	ARTIFICIAL INTELLIGENCE
It is a natural process.	It is programmed by humans.
It is actually hereditary.	It is not hereditary.
Knowledge is required for intelligence.	KB and electricity are required to generate output.
No human is an expert. We may get better solutions from other humans.	Expert systems are made which aggregate many person's experience and ideas.

1.1 DEFINITION

The study of how to make computers do things at which at the moment, people are better. **"Artificial Intelligence is the ability of a computer to act like a human being".**

- Systems that think like humans
- Systems that act like humans
- Systems that think rationally.
- Systems that act rationally.

"The exciting new effort to make computers think ... *machines with minds*, in the full and literal sense" (Haugeland, 1985) "[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning ..." (Bellman, 1978)	"The study of mental faculties through the use of computational models" (Charniak and McDermott, 1985) "The study of the computations that make it possible to perceive, reason, and act" (Winston, 1992)
"The art of creating machines that perform functions that require intelligence when performed by people" (Kurzweil, 1990) "The study of how to make computers do things at which, at the moment, people are better" (Rich and Knight, 1991)	"A field of study that seeks to explain and emulate intelligent behavior in terms of computational processes" (Schalkoff, 1990) "The branch of computer science that is concerned with the automation of intelligent behavior" (Luger and Stubblefield, 1993)

Figure 1.1 Some definitions of artificial intelligence, organized into four categories

(a) **Intelligence -** Ability to apply knowledge in order to perform better in an environment.

(b) **Artificial Intelligence -** Study and construction of agent programs that perform well in a given environment, for a given agent architecture.

(c) **Agent -** An entity that takes action in response to precepts from an environment.

(d) **Rationality** - property of a system which does the "right thing" given what it knows.

(e) **Logical Reasoning -** A process of deriving new sentences from old, such that the new sentences are necessarily true if the old ones are true.

Four Approaches of Artificial Intelligence:

- Acting humanly: The Turing test approach.
- Thinking humanly: The cognitive modelling approach.
- Thinking rationally: The laws of thought approach.
- Acting rationally: The rational agent approach.

1.1.1 ACTING HUMANLY: THE TURING TEST APPROACH

The **Turing Test**, proposed by Alan Turing (1950), was designed to provide a satisfactory operational definition of intelligence. A computer passes the test if a human interrogator, after posing some written questions, cannot tell whether the written responses come from a person or from a computer.

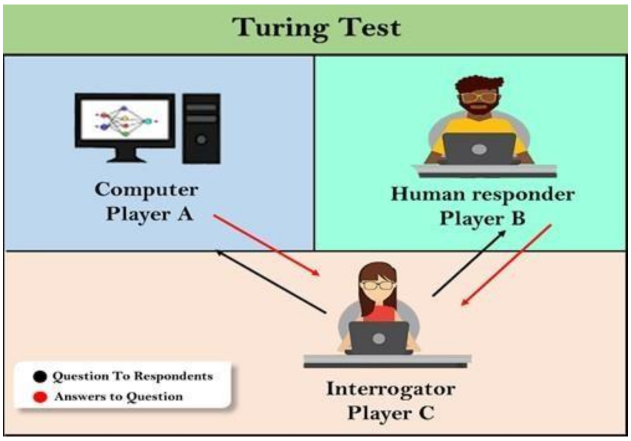


Figure 1.2 Turing Test

we note that programming a computer to pass a rigorously applied test provides plenty to work on. The computer would need the following capabilities:

- **natural language processing** to enable it to communicate successfully in English;
- **knowledge representation** to store what it knows or hears;
- **automated reasoning** to use the stored information to answer questions and to draw new conclusions
- **machine learning** to adapt to new circumstances and to detect and extrapolate patterns.

Total Turing Test which requires interaction with objects and people in the real world.To pass the total Turing Test, the computer will need

- **computer vision** to perceive objects
- **robotics** to manipulate objects and move about**.**

These six disciplines compose most of AI, and Turing deserves credit for designing a test that remains relevant 60 years later.

1.1.2 THINKING HUMANLY: THE COGNITIVE MODELLING APPROACH

Analyse how a given program thinks like a human, we must have some way of determining how humans think. We need to get *inside* the actual workings of human minds. There are three ways to do this: through introspection—trying to catch our own thoughts as they go by; through psychological experiments—observing a person in action; and through brain imaging—observing the brain in action. The interdisciplinary field of **cognitive science** brings together computer models from AI and experimental techniques from psychology to try to construct precise and testable theories of the workings of the human mind.

Although cognitive science is a fascinating field in itself, we are not going to be discussing it all that much in this book. We will occasionally comment on similarities or differences between AI techniques and human cognition. Real cognitive science, however, is necessarily based on experimental investigation of actual humans or animals, and we assume that the reader only has access to a computer for experimentation. We will simply note that AI and cognitive science continue to fertilize each other, especially in the areas of vision, natural language, and learning.

1.1.3 THINKING RATIONALLY: THE "LAWS OF THOUGHT" APPROACH

The Greek philosopher Aristotle was one of the first to attempt to codify ``right thinking," that is, irrefutable reasoning processes. His famous syllogisms provided patterns for argument structures that always gave correct conclusions given correct premises.

For example, ``Socrates is a man; all men are mortal; therefore Socrates is mortal."

These laws of thought were supposed to govern the operation of the mind, and initiated the field of ***logic.***

Logicians in the 19th century developed a precise notation for statements about all kinds of objects in the world and the relations among them. By 1965, programs existed that could, in principle, solve *any* solvable problem described in logical notation. The so-called **logicist** tradition within artificial intelligence hopes to build on such programs to create intelligent systems. There are two main obstacles to this approach. First, it is not easy to take informal knowledge and state it in the formal terms required by logical notation, particularly when the knowledge is less than 100% certain. Second, there is a big difference between solving a problem "in principle" and solving it in practice.

1.1.4 ACTING RATIONALLY: THE RATIONAL AGENT APPROACH

An **agent** is just something that acts (*agent* comes from the Latin *agere*, to do). Of course, all computer programs do something, but computer agents are expected to do more: operate autonomously, perceive their environment, persist over a prolonged time period, adapt to change, and create and pursue goals. A **rational agent** is one that acts so as to achieve the best outcome or, when there is uncertainty, the best expected outcome.

In the "laws of thought" approach to AI, the emphasis was on correct inferences. Making correct inferences is sometimes *part* of being a rational agent. All the skills needed for the Turing Test also allow an agent to act rationally.

The rational-agent approach has two advantages over the other approaches. First, it is more general than the "laws of thought" approach because correct inference is just one of several possible mechanisms for achieving rationality. Second, it is more amenable to scientific development than are approaches based on human behavior or human thought.

One important point to keep in mind: We will see before too long that achieving perfect rationality—always doing the right thing—is not feasible in complicated environments. The computational demands are just too high.

1.2 FUTURE OF ARTIFICIAL INTELLIGENCE

- **Transportation:** Although it could take a decade or more to perfect them, autonomous cars will one day ferry us from place to place.
- **Manufacturing:** AI powered robots work alongside humans to perform a limited range of tasks like assembly and stacking, and predictive analysis sensors keep equipment running smoothly.
- **Healthcare:** In the comparatively AI-nascent field of healthcare, diseases are more quickly and accurately diagnosed, drug discovery is speed up and streamlined, virtual nursing assistants monitor patients and big data analysis helps to create a more personalized patient experience.
- **Education:** Textbooks are digitized with the help of AI, early-stage virtual tutors assist human instructors and facial analysis gauges the emotions of students to help determine who's struggling or bored and better tailor the experience to their individual needs.
- **Media:** Journalism is harnessing AI, too, and will continue to benefit from it. Bloomberg uses Cyborg technology to help make quick sense of complex financial reports. The Associated Press employs the natural language abilities of Automated Insights to produce 3,700 earning reports stories per year — nearly four times more than in the recent past
- **Customer Service:** Last but hardly least, Google is working on an AI assistant that can place human-like calls to make appointments at, say, your neighborhood hair salon. In addition to words, the system understands context and nuance.

1.3 AGENTS AND ENVIRONMENTS

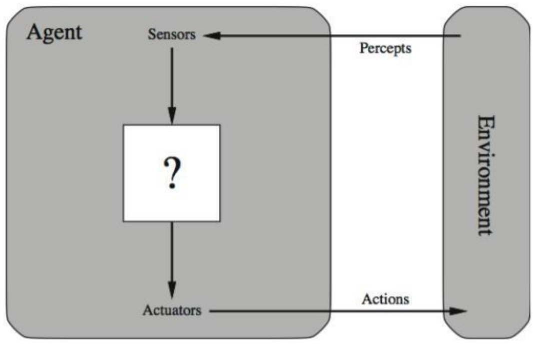


Figure 1.3 Agent types

An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.

- Human Sensors:
- Eyes, ears, and other organs for sensors.
- Human Actuators:
- Hands, legs, mouth, and other body parts.
- Robotic Sensors:
- Mic, cameras and infrared range finders for sensors
- Robotic Actuators:
- Motors, Display, speakers etc

An agent can be:

Human-Agent: A human agent has eyes, ears, and other organs which work for sensors and hand, legs, vocal tract work for actuators.

Robotic Agent: A robotic agent can have cameras, infrared range finder, NLP for sensors and various motors for actuators.

Software Agent: Software agent can have keystrokes, file contents as sensory input and act on those inputs and display output on the screen.

Hence the world around us is full of agents such as thermostat, cell phone, camera, and even we are also agents. Before moving forward, we should first know about sensors, effectors, and actuators.

Sensor: Sensor is a device which detects the change in the environment and sends the information to other electronic devices. An agent observes its environment through sensors.

Actuators: Actuators are the component of machines that converts energy into motion.

The actuators are only responsible for moving and controlling a system. An actuator can be an electric motor, gears, rails, etc.

Effectors: Effectors are the devices which affect the environment. Effectors can be legs, wheels, arms, fingers, wings, fins, and display screen.

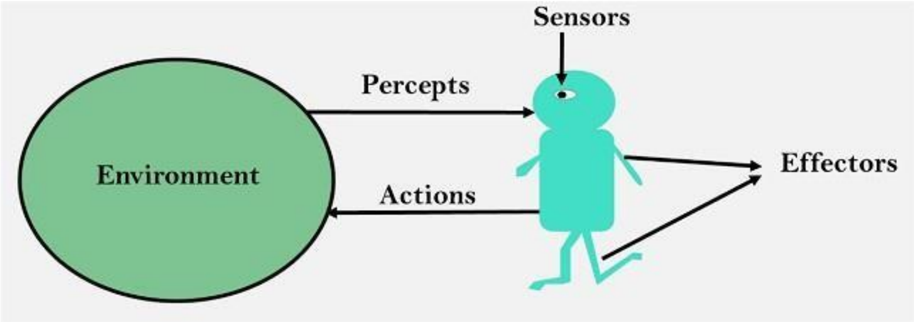


Figure 1.4 Effectors

GOOD BEHAVIOUR:The Concept of Rationality

Definition of a Rational Agent: - A system is rational if it does the “right thing” given what it knows.

Characteristic of Rational Agent

- The agent's prior knowledge of the environment.
- The performance measure that defines the criterion of success.
- The actions that the agent can perform.
- The agent's percept sequence to date.

For every possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.

- An **omniscient agent** knows the actual outcome of its actions and can act accordingly; but omniscience is impossible in reality.

- **Ideal Rational Agent** precepts and does things. It has a greater performance measure.

  Eg. Crossing road. Here first perception occurs on both sides and then only action. No perception occurs in **Degenerate Agent**.

  Eg. Clock. It does not view the surroundings. No matter what happens outside. The clock works based on inbuilt program.

- **Ideal Agent** describes by ideal mappings. “Specifying which action an agent ought to take in response to any given percept sequence provides a design for ideal agent”.
- **Eg.** SQRT function calculation in calculator.

- Doing actions in order to modify future precepts-sometimes called **information gathering**- is an important part of rationality.

- A rational agent should be **autonomous**-it should learn from its own prior knowledge (experience).

The Structure of Intelligent Agents

Agent = Architecture + Agent Program

Architecture = the machinery that an agent executes on. (Hardware)

Agent Program = an implementation of an agent function. (Algorithm, Logic – Software)

1.4 NATURE or PROPERTIES OF ENVIRONMENT

An **environment** is everything in the world which surrounds the agent, but it is not a part of an agent itself. An environment can be described as a situation in which an agent is present.

The environment is where agent lives, operate and provide the agent with something to sense and act upon it.

Fully observable vs Partially Observable:

If an agent sensor can sense or access the complete state of an environment at each point of time then it is **a fully observable** environment, else it is **partially observable**.

A fully observable environment is easy as there is no need to maintain the internal state to keep track history of the world.

An agent with no sensors in all environments then such an environment is called as unobservable.

Example: chess – the board is fully observable, as are opponent's moves. Driving – what is around the next bend is not observable and hence partially observable.

2 Deterministic vs Stochastic

- If an agent's current state and selected action can completely determine the next state of the environment, then such environment is called a deterministic environment.

- A stochastic environment is random in nature and cannot be determined completely by an agent.
- In a deterministic, fully observable environment, agent does not need to worry about uncertainty.

3. **Episodic vs Sequential**

- In an episodic environment, there is a series of one-shot actions, and only the current percept is required for the action.
- However, in Sequential environment, an agent requires memory of past actions to determine the next best actions.

4. **Single-agent vs Multi-agent**

- If only one agent is involved in an environment, and operating by itself then such an environment is called single agent environment.
- However, if multiple agents are operating in an environment, then such an environment is called a multi-agent environment.
- The agent design problems in the multi-agent environment are different from single agent environment.

5. **Static vs Dynamic**

- If the environment can change itself while an agent is deliberating then such environment is called a dynamic environment else it is called a static environment.
- Static environments are easy to deal because an agent does not need to continue looking at the world while deciding for an action.
- However for dynamic environment, agents need to keep looking at the world at each action.
- Taxi driving is an example of a dynamic environment whereas Crossword puzzles are an example of a static environment.

6. **Discrete vs Continuous**

- If in an environment there are a finite number of precepts and actions that can be performed within it, then such an environment is called a discrete environment else it is called continuous environment.
- A chess game comes under discrete environment as there is a finite number of moves that can be performed.
- A self-driving car is an example of a continuous environment.

7. Known vs Unknown

- Known and unknown are not actually a feature of an environment, but it is an agent's state of knowledge to perform an action.
- In a known environment, the results for all actions are known to the agent. While in unknown environment, agent needs to learn how it works in order to perform an action.
- It is quite possible that a known environment to be partially observable and an Unknown environment to be fully observable.

8. Accessible vs. Inaccessible

- If an agent can obtain complete and accurate information about the state's environment, then such an environment is called an Accessible environment else it is called inaccessible.
- An empty room whose state can be defined by its temperature is an example of an accessible environment.
- Information about an event on earth is an example of Inaccessible environment.

Task environments, which are essentially the "problems" to which rational agents are the "solutions."

PEAS: Performance Measure, Environment, Actuators, Sensors

Performance

The output which we get from the agent. All the necessary results that an agent gives after processing comes under its performance.

Environment

All the surrounding things and conditions of an agent fall in this section. It basically consists of all the things under which the agents work.

Actuators

The devices, hardware or software through which the agent performs any actions or processes any information to produce a result are the actuators of the agent.

Sensors

The devices through which the agent observes and perceives its environment are the sensors of the agent.

Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments, referrals	Keyboard entry of symptoms, findings, patient's answers
Satellite image analysis system	Correct image categorization	Downlink from orbiting satellite	Display of scene categorization	Color pixel arrays
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts; bins	Jointed arm and hand	Camera, joint angle sensors
Refinery controller	Purity, yield, safety	Refinery, operators	Valves, pumps, heaters, displays	Temperature, pressure, chemical sensors
Interactive English tutor	Student's score on test	Set of students, testing agency	Display of exercises, suggestions, corrections	Keyboard entry

Figure 1.5 Examples of agent types and their PEAS descriptions

1.5 TYPES AND STRUCTURE OF AGENTS

Agents can be grouped into four classes based on their degree of perceived intelligence and capability :

- Simple Reflex Agents
- Model-Based Reflex Agents
- Goal-Based Agents
- Utility-Based Agents
- Learning Agent

The Simple reflex agents

- The Simple reflex agents are the simplest agents. These agents take decisions on the basis of the current percepts and ignore the rest of the percept history (**past State).**
- These agents only succeed in the fully observable environment.
- The Simple reflex agent does not consider any part of percepts history during their decision and action process.
- The Simple reflex agent works on Condition-action rule, which means it maps the current state to action. Such as a Room Cleaner agent, it works only if there is dirt in the room.
- Problems for the simple reflex agent design approach:

- o They have very limited intelligence
- o They do not have knowledge of non-perceptual parts of the current state
- o Mostly too big to generate and to store.
- o Not adaptive to changes in the environment.

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

Ex: if car-in-front-is-braking then initiate- braking.

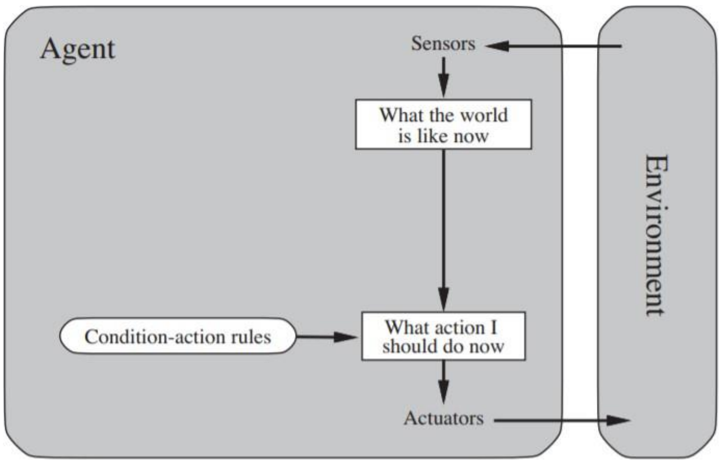


Figure 1.6 A simple reflex agent

Model Based Reflex Agents

- The Model-based agent can work in a partially observable environment, and track the situation.
- A model-based agent has two important factors:
  - o **Model:** It is knowledge about "how things happen in the world," so it is called a Model-based agent.
  - o **Internal State:** It is a representation of the current state based on percept history.
- These agents have the model, "which is knowledge of the world" and based on the model they perform actions.
- Updating the agent state requires information about:
  - o How the world evolves
  - o How the agent's action affects the world.

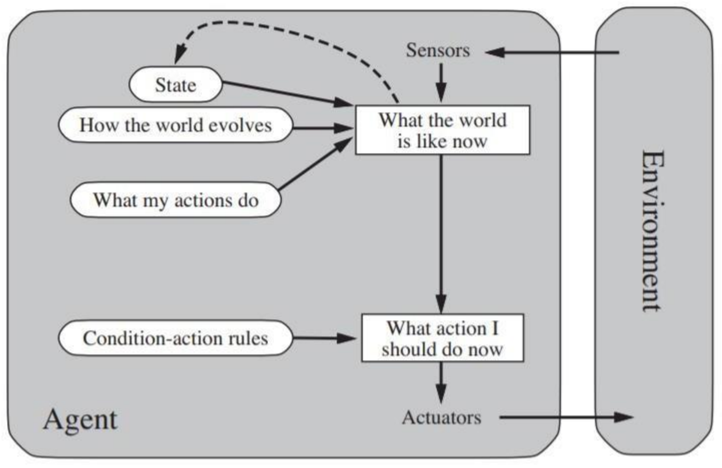


Figure 1.7 A model-based reflex agent

Goal Based Agents

- The knowledge of the current state environment is not always sufficient to decide for an agent to what to do.
- The agent needs to know its goal which describes desirable situations.
- Goal-based agents expand the capabilities of the model-based agent by having the "goal" information.
- They choose an action, so that they can achieve the goal.
- These agents may have to consider a long sequence of possible actions before deciding whether the goal is achieved or not. Such considerations of different scenario are called searching and planning, which makes an agent proactive.

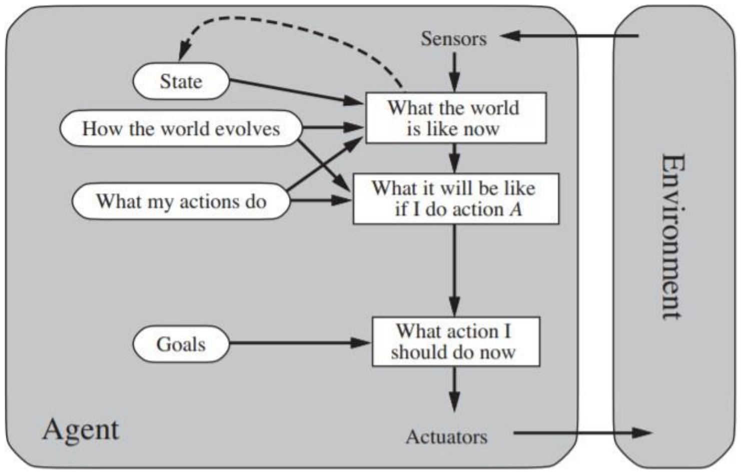


Figure 1.8 A goal-based agent

Utility Based Agents

- These agents are similar to the goal-based agent but provide an extra component of utility measurement **("Level of Happiness")** which makes them different by providing a measure of success at a given state.
- Utility-based agent act based not only goals but also the best way to achieve the goal.
- The Utility-based agent is useful when there are multiple possible alternatives, and an agent has to choose in order to perform the best action.
- The utility function maps each state to a real number to check how efficiently each action achieves the goals.

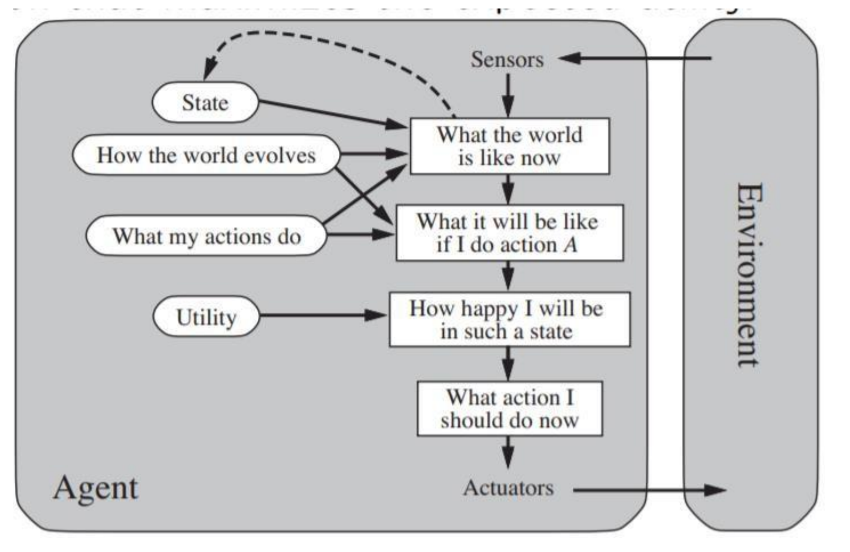


Figure 1.9 A utility-based agent

Learning Agents

- A learning agent in AI is the type of agent which can learn from its past experiences, or it has learning capabilities.
- It starts to act with basic knowledge and then able to act and adapt automatically through learning.
- A learning agent has mainly four conceptual components, which are:

  a. **Learning element:** It is responsible for making improvements by learning from environment

  b. **Critic:** Learning element takes feedback from critic which describes that how well the agent is doing with respect to a fixed performance standard.

  c. **Performance element:** It is responsible for selecting external action

d. **Problem generator:** This component is responsible for suggesting actions that will lead to new and informative experiences.

- Hence, learning agents are able to learn, analyze performance, and look for new ways to improve the performance.

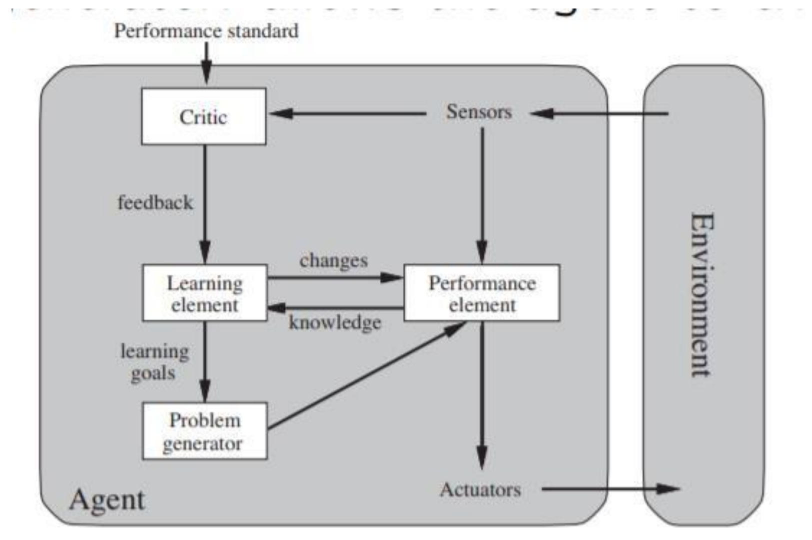


Figure 1.10 Learning Agents

1.6 PROBLEM SOLVING APPROACH TO TYPICAL AI PROBLEMS

1.6.1Problem-solving agents

In Artificial Intelligence, Search techniques are universal problem-solving methods. **Rational agents** or **Problem-solving agents** in AI mostly used these search strategies or algorithms to solve a specific problem and provide the best result. Problem- solving agents are the goal-based agents and use atomic representation. In this topic, wewill learn various problem-solving search algorithms.

- Some of the most popularly used problem solving with the help of artificial intelligence are:

  1. Chess.
  2. Travelling Salesman Problem.
  3. Tower of Hanoi Problem.
  4. Water-Jug Problem.
  5. N-Queen Problem.

1.6.2 Problem Searching

- In general, searching refers to as finding information one needs.

- Searching is the most commonly used technique of problem solving in artificial intelligence.
- The searching algorithm helps us to search for solution of particular problem.

Problem: Problems are the issues which comes across any system. A solution is needed to solve that particular problem.

Steps : Solve Problem Using Artificial Intelligence

- The process of solving a problem consists of five steps. These are:

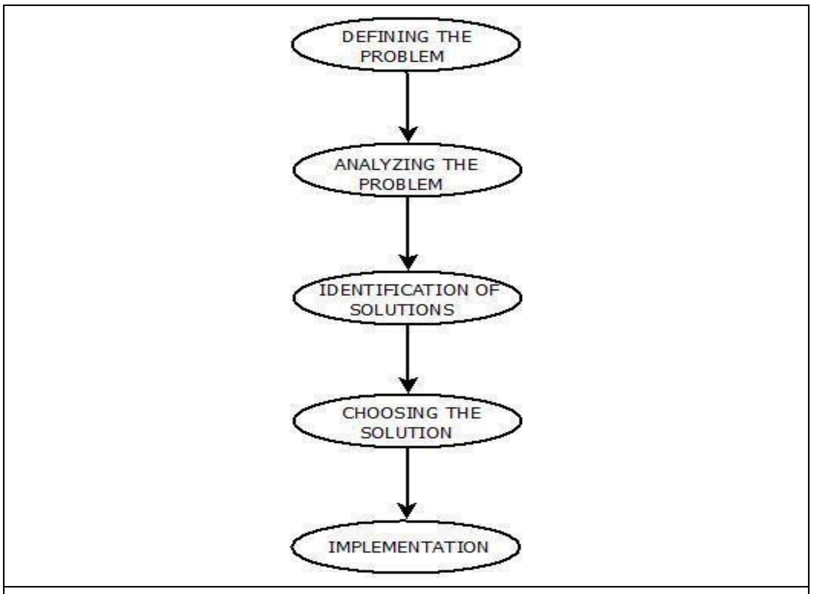


Figure 1.11 Problem Solving in Artificial Intelligence

Defining The Problem: The definition of the problem must be included precisely. It should contain the possible initial as well as final situations which should result in acceptable solution.

1. **Analyzing The Problem**: Analyzing the problem and its requirement must be done as few features can have immense impact on the resulting solution.
2. **Identification Of Solutions**: This phase generates reasonable amount of solutions to the given problem in a particular range.
3. **Choosing a Solution**: From all the identified solutions, the best solution is chosen basis on the results produced by respective solutions.
4. **Implementation**: After choosing the best solution, its implementation is done.

Measuring problem-solving performance

We can evaluate an algorithm's performance in four ways:

Completeness: Is the algorithm guaranteed to find a solution when there is one?

Optimality: Does the strategy find the optimal solution?

Time complexity: How long does it take to find a solution?

Space complexity: How much memory is needed to perform the search?

Search Algorithm Terminologies

- Search: Searching is a step by step procedure to solve a search-problem in a given search space. A search problem can have three main factors:
  1. Search Space: Search space represents a set of possible solutions, which a system may have.
  2. Start State: It is a state from where agent begins the search.
  3. Goal test: It is a function which observe the current state and returns whether the goal state is achieved or not.
- Search tree: A tree representation of search problem is called Search tree. The root of the search tree is the root node which is corresponding to the initial state.
- Actions: It gives the description of all the available actions to the agent.
- Transition model: A description of what each action do, can be represented as a transition model.
- Path Cost: It is a function which assigns a numeric cost to each path.
- Solution: It is an action sequence which leads from the start node to the goal node. Optimal Solution: If a solution has the lowest cost among all solutions.

1.6.3 Example Problems

A **Toy Problem** is intended to illustrate or exercise various problem-solving methods. A **real- world problem** is one whose solutions people actually care about.

1.6.3.1 Toy Problems

1)Vacuum World

States: The state is determined by both the agent location and the dirt locations. The agent is in one of the 2 locations, each of which might or might not contain dirt. Thus there are 2*2^2=8 possible world states.

Initial state: Any state can be designated as the initial state.

Actions: In this simple environment, each state has just three actions: *Left*, *Right*, and *Suck*. Larger environments might also include *Up* and *Down*.

Transition model: The actions have their expected effects, except that moving *Left* in the leftmost squ are, moving *Right* in the rightmost square, and *Suck*ing in a clean square have no effect. The complete state space is shown in Figure.

Goal test: This checks whether all the squares are clean.

Path cost: Each step costs 1, so the path cost is the number of steps in the path.

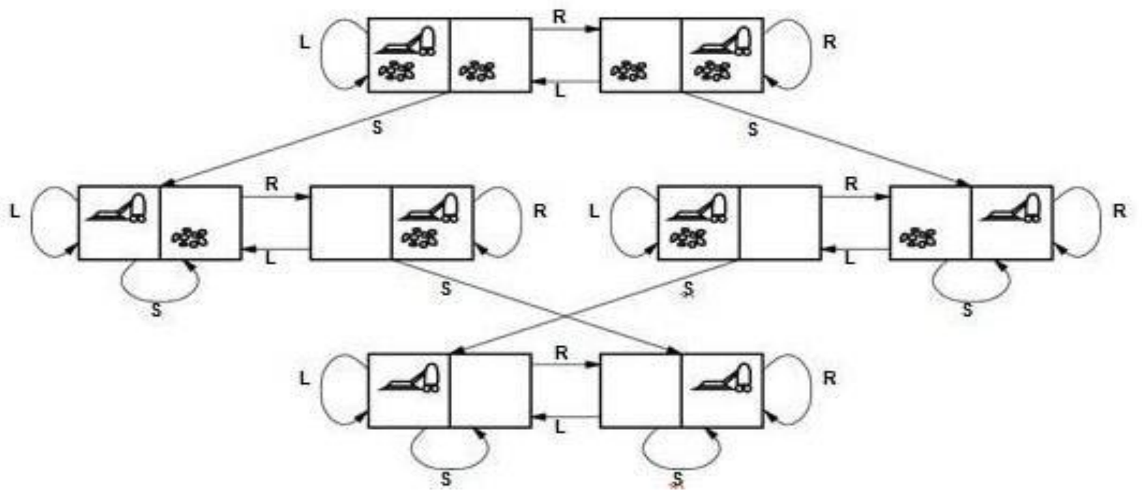


Figure 1.12 Vacuum World State Space Graph

2) 8- Puzzle Problem

1. States

- Representation: Each state in the 8-puzzle is represented by a 3x3 grid containing numbers from 1 to 8 and an empty space.
- Example:
  1 2 3
  4 5 6
  7 8

2. Initial State

- The initial state is the starting configuration of the puzzle, which can vary.
  - Example:
    2 8 3
    1 6 4
    7 0 5

(where '0' represents the empty space)

3. Transition Model

- Actions: The possible actions in the 8-puzzle are:
  - Move the empty space up: If the empty space is not in the top row.
  - Move the empty space down: If the empty space is not in the bottom row.
  - Move the empty space left: If the empty space is not in the leftmost column.
  - Move the empty space right: If the empty space is not in the rightmost column.

- Resulting State: Applying an action to a given state results in a new state where the empty space has moved accordingly.

4. Goal Test

- Objective: The goal of the 8-puzzle is to reach a specific goal state.
- Example:
  1 2 3
  4 5 6
  7 8 0

5. Path Cost

- Definition: The path cost represents the "distance" traveled to reach the goal state.
- Common Metric:
  Unit Cost: Each move (up, down, left, right) has a cost of 1.
  Total Cost: The total path cost is the sum of the costs of all the moves made to reach the goal state.

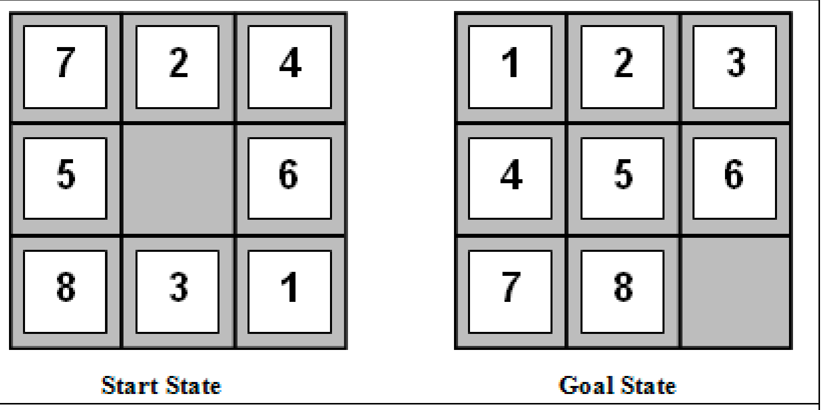


Figure 1.13 8- Puzzle Problem

3)8-Queens Problem

1. States

- A state is represented by the configuration of queens on the chessboard.
- Typically, it can be represented as a tuple or list where each element specifies the column position of a queen in a particular row. For example, [1, 3, 5] means a queen is placed at row 1, column 1; row 2, column 3; and row 3, column 5.
- Partial configurations are also considered states in search problems.

2. Initial State

- The board is empty, i.e., no queens are placed on the board.
- Represented as an empty list, [].

3. Actions

- Place a queen in a valid position on the next row.
- The action is defined as adding a queen to a specific column in the current row, provided it does not conflict with already placed queens.

  Example:

- If the current state is [1, 3] (queens in row 1, column 1; row 2, column 3), valid actions for row 3 might be columns [0, 2, 4, 6].

4. Transition Model

- The transition model updates the state by adding a queen in the next row.
- For example, if the current state is [1, 3] and the action is to place a queen in column 5 of row 3, the next state would be [1, 3, 5].

5. Goal Test

- The goal test checks whether all 8 queens are placed on the board and no two queens attack each other.
- Conditions for a valid configuration:
    - No two queens are in the same column.
    - two queens are on the same diagonal.

6. Path Cost

- The path cost is typically irrelevant in this problem because it is a constraint satisfaction problem, not an optimization problem.
- If a cost is defined, it could simply be the number of steps (queens placed) to reach the goal, which would be 8 for a valid solution.

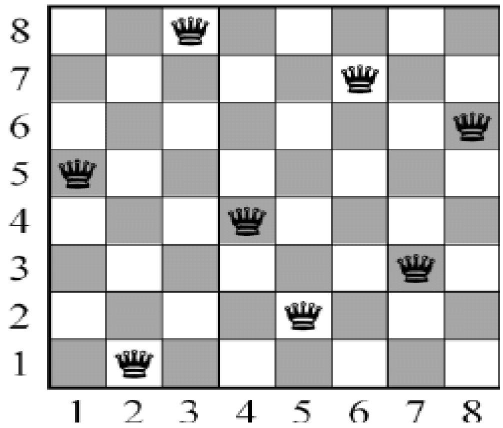


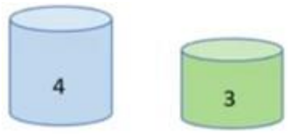
Figure 1.14 Queens Problem

4)Water jug Problem

Consider the given problem. Describe the operator involved in it. Consider the water jug problem: You are given two jugs, a 4-gallon one and 3-gallon one. Neither has any measuring marker on it. There is a pump that can be used to fill the jugs with water. How can you get exactly 2 gallon of water from the 4-gallon jug ?

Explicit Assumptions: A jug can be filled from the pump, water can be poured out of a jug on to the ground, water can be poured from one jug to another and that there are no other measuring devices available.

- 4-gallon one and a 3-gallon Jug



- 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?

1. Problem Characteristics:

- **States:** Each state is represented by the amount of water in each jug. For example, if you have a 5-liter jug and a 3-liter jug, a state could be represented as (x, y), where x is the amount of water in the 5-liter jug and y is the amount in the 3-liter jug.

- Initial **State:** The initial state is where both jugs are empty. This is typically represented as (0, 0).

- **Goal** State**:** The goal is to reach a state where one of the jugs contains the target amount of water. For example, if the goal is to measure 4 liters, the goal state might be (4, 0) or (0, 4), depending on the problem definition.

2. State Transition Model:

The state transition model defines how the system can move from one state to another. The possible actions include:

- **Fill a jug:** Fill one jug completely (e.g., fill the 5-liter jug).

- **Empty a jug:** Empty one jug completely.

- **Pour water from one jug to the other:** Pour water from one jug into the other until one jug is either empty or full, depending on which jug you're pouring into.

3. Actions:

- **Fill** either of the two jugs completely (e.g., fill the 5-liter jug to its capacity).

- **Empty** either of the two jugs completely (e.g., empty the 3-liter jug).

- **Pour** water from one jug into the other, where you can either fill one jug completely or empty the other jug, depending on the situation.

4. Goal Test:

The goal test checks if the current state satisfies the goal condition. In this case, the goal is to find a state where 4-galaon jug contains the exact target amount of water (e.g., 2 liters). For instance, the goal is, to get exactly 2 litres, the test will check if 4 litres jug contains 2 liters of water

5. Path Cost:

The path cost refers to the cost associated with reaching a particular state from the initial state. In the basic water jug problem, each action (such as filling, emptying, or pouring water) is typically assigned a cost of 1 unit. Therefore, the total path cost would be the number of actions taken to reach the goal state.

Example Problem:

- Jug **capacities:** 4 liters and 3 liters

- **Goal:** Measure exactly 2 liters in one of the jugs.

In this case:

- **Initial state:** (0, 0) — both jugs are empty.

- **Goal state:** (2, 0) or (0, 2) — one jug has 4 liters of water.

- **State Transition Model:** You can perform actions like filling either jug, emptying either jug, or pouring from one jug to the other.

- **Action example:** Fill the 4-liter jug (state becomes (4, 0)), then pour it into the 3-liter jug (state becomes (1, 3)), and so on.

- **Path cost:** Each action (such as filling, emptying, or pouring) has a cost of 1, so the total path cost will be the number of steps you take to reach the goal.

This problem is often solved using search techniques such as **Breadth-First Search** (BFS), **Depth-First Search** (DFS), or *A Search**, depending on the problem's complexity and the desired solution.

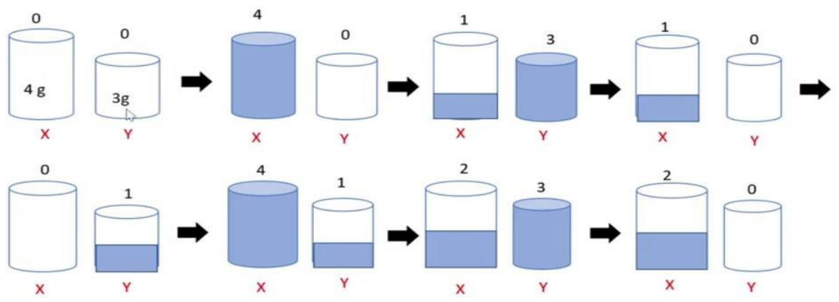


Figure 1.15 Solution

Table 1.2

Solution

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

1.6.3.2 REAL-WORLD PROBLEMS

1.ROUTE-FINDING PROBLEM

Route-finding problem is defined in terms of specified locations and transitions along links between them. Route-finding algorithms are used in a variety of applications, such as routing in computer networks, military operations planning, and airline travel planning systems.

a)AIRLINE TRAVEL PROBLEM

The **airline travel problem** is specifies as follows**:**

- **States:** Each is represented by a location (e.g., an airport) and the current time.
- **Initial state:** This is specified by the problem.
- **Successor function:** This returns the states resulting from taking any scheduled flight (further specified by seat class and location),leaving later than the current time plus the within-airport transit time, from the current airport to another.
- **Goal Test:** Are we at the destination by some prespecified time?
- **Path cost:** This depends upon the monetary cost, waiting time, flight time, customs and immigration procedures, seat quality, time of date, type of air plane, frequent-flyer mileage awards, and so on.

b)TOURING PROBLEMS

Touring problems are closely related to route-finding problems, but with an important difference. Consider for example, the problem, "Visit every city at least once" as shown in Romania map.

As with route-finding the actions correspond to trips between adjacent cities. The state space, however, is quite different.

The initial state would be "In Bucharest; visited{Bucharest}".

A typical intermediate state would be "In Vaslui;visited {Bucharest, Urziceni,Vaslui}".

The goal test would check whether the agent is in Bucharest and all 20 cities have been visited.

c)THE TRAVELLING SALESPERSON PROBLEM(TSP)

Is a touring problem in which each city must be visited exactly once. The aim is to find the shortest tour. The problem is known to be **NP-hard**. Enormous efforts have been expended to improve the capabilities of TSP algorithms. These algorithms are also used in tasks such as planning movements of **automatic circuit-board drills** and of **stocking machines** on shop floors.

d)VLSI layout

A **VLSI layout** problem requires positioning millions of components and connections on a chip to minimize area, minimize circuit delays, minimize stray capacitances, and maximize manufacturing yield. The layout problem is split into two parts: **cell layout** and **channel routing**.

e)ROBOT navigation

ROBOT navigation is a generalization of the route-finding problem. Rather than a discrete set of routes, a robot can move in a continuous space with an infinite set of possible actions and states. For a circular Robot moving on a flat surface, the space is essentially two-dimensional. When the robot has arms and legs or wheels that also must be controlled, the search space becomes multi-dimensional. Advanced techniques are required to make the search space finite.