Course Code: CSE3002
Course Name: Artificial Intelligence

Faculty Name:
D Ramkumar
AP/SCOPE
VIT-AP

Course Objectives

- To have a thorough understanding of classical and modern AI applications
- To implement a wide range of AI concepts
- To understand non-classical AI approaches such as genetic algorithms and neural networks
- To be able to assess the potential of AI in research and real-world environments

CO's Mapping with PO's and PEO's

Course Outcomes	Course Outcome Statement	PO's / PEO's
CO1	Understand the basics of AI	PO1,PO2,PO5,PO8
CO2	Implement and debug core AI algorithms in a clean and structured manner	PO2,PO6,PO7
CO3	Describe AI algorithms and representations and explain their performance, in writing and orally	PO4,PO6,PO9
CO4	Analyze and design a real world AI application	PO1,PO2,PO6,PO7,PO12
		TOTAL HOURS OF INSTRUCTIONS: 45

Rubrics

- Grading policy

Topic	**Weightage**
Continuous Assessment Test-1	20%
Continuous Assessment Test-2	20%
FAT	20%
Continuous Evaluation	20%
Practical Assessment	20%

Course Logistics

- Approximately 3 assignments
- Assignments to be submitted within the given timeline → **Delay in submission will cause penalty (5%/day) → If any two copies look similar, both will be awarded a zero, no request will be considered.**
- **Attendance rules of the institute will be followed**

Class timings

Slot-C2

Venue: 230-CB

Tuesday: 5.00 pm – 5.50 pm

Thursday: 2.00 pm -2.50 pm

Friday: 3.00 pm – 3.50 pm

Contents

Module 1: Introduction To Al And Production Systems
Introduction to AI-Problem formulation, Problem Definition -Production systems, Control strategies, Search strategies. Problem characteristics, Production system characteristics -Specialized production system

Module 2: Problem Solving methods
Problem graphs, Matching, Indexing and Heuristic functions -Hill Climbing-Depth first and Breath first, Constraints satisfaction - Related algorithms, Measure of performance and analysis of search algorithms.

Module 3: Knowledge Representation
Knowledge based agents- Prepositional Logic- First Order logic- Inferences

Contents

Module 4: Knowledge Inference

Knowledge representation -Production based system, Frame based system. Inference - Backward chaining, Forward chaining, Rule value approach, Fuzzy reasoning - Certainty factors, Bayesian Theory-Bayesian Network-Dempster - Shafer theory

Module 5: Planning And Learning

Basic plan generation systems - Strips -Advanced plan generation systems – K strips -Strategic explanations – Explanation bases Learning- Machine learning, adaptive Learning. Reinforcement learning- Decision tree learning

Module 6: Expert Systems

Genetic algorithms Expert systems - Architecture of expert systems, Roles of expert systems - Knowledge Acquisition –Meta knowledge, Heuristics.

Books

Text Books

Stuart Russell and Peter Norvig, “Artificial Intelligence: A Modern Approach” Prentice Hall, Third Edition, 2009

References

- Deepak Khemani, “A First Course in Artificial Intelligence”, McGraw Hill Education (India), 2013.
- Nick Bostrom, “Superintelligence:Paths,Dangers,Strategies”, 1st edition, 2014.
- David Poole,Alan Mackworth, “Artificial Intelligence:Foundations of Computational Agents”, 2nd edition, 2017.
- Some Research Papers

Module-1

Module No. 1	Introduction To Al And Production Systems	7 Hours

Introduction to AI-Problem formulation, Problem Definition -Production systems, Control strategies, Search strategies. Problem characteristics, Production system characteristics -Specialized production system-

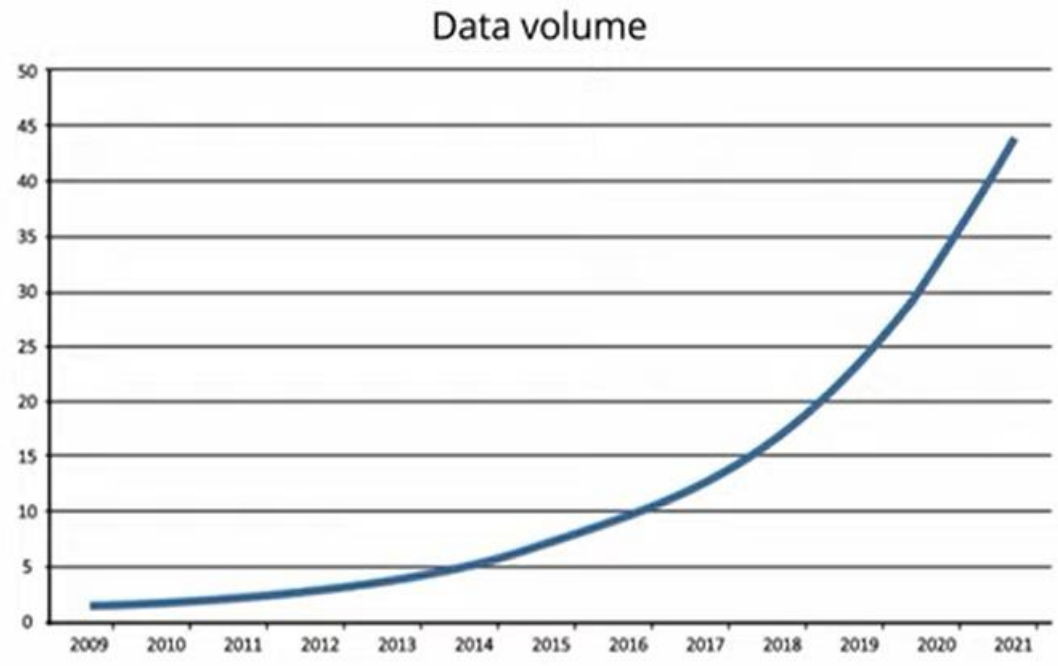
Learning Objectives

By the end of this lesson, you will be able to:

- Define artificial intelligence (AI)
- Describe the relationship between artificial intelligence (AI) and data science
- Define machine learning (ML)

Data Economy

Data economy refers to how much data has grown over the past few years and how much more it can grow in the coming years.

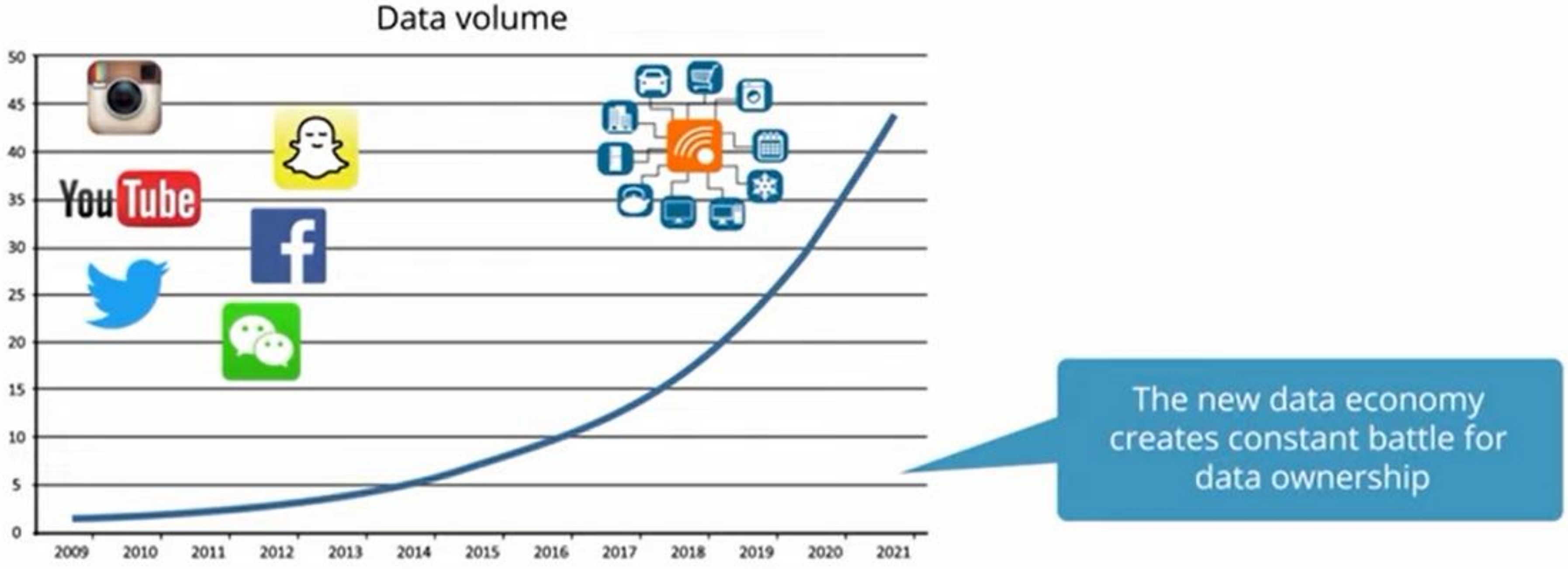


Sources: IDC, Azeem Azhar analysis

Data Economy

Data economy refers to how much data has grown over the past few years and how much more it can grow in the coming years.

44x in 11 years



Sources: IDC, Azeem Azhar analysis

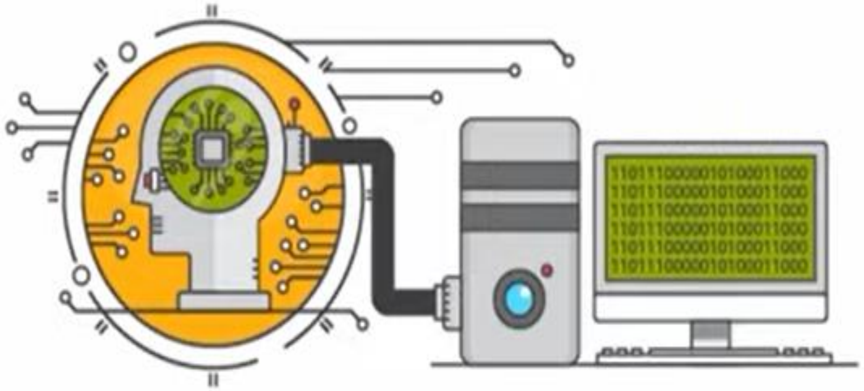
Emergence of Artificial Intelligence

You can teach machines to learn from data and derive a variety of useful insights giving rise to artificial intelligence.

Definition of Artificial Intelligence

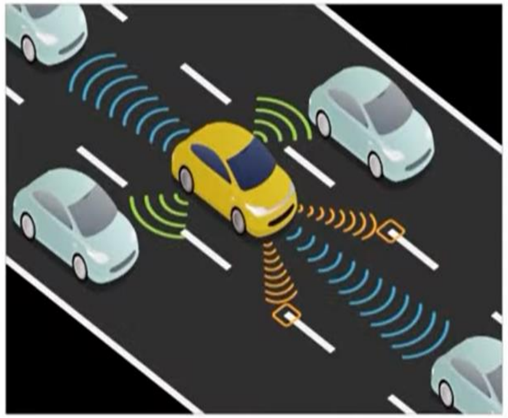
VIT-AP UNIVERSITY

Artificial Intelligence refers to intelligence displayed by machines that simulates human and animal intelligence.



It enables computers to mimic human intelligence using logic.

Artificial Intelligence in Practice



Self-driving cars

Artificial Intelligence in Practice

Siri (iPhone)

VIT-AP UNIVERSITY

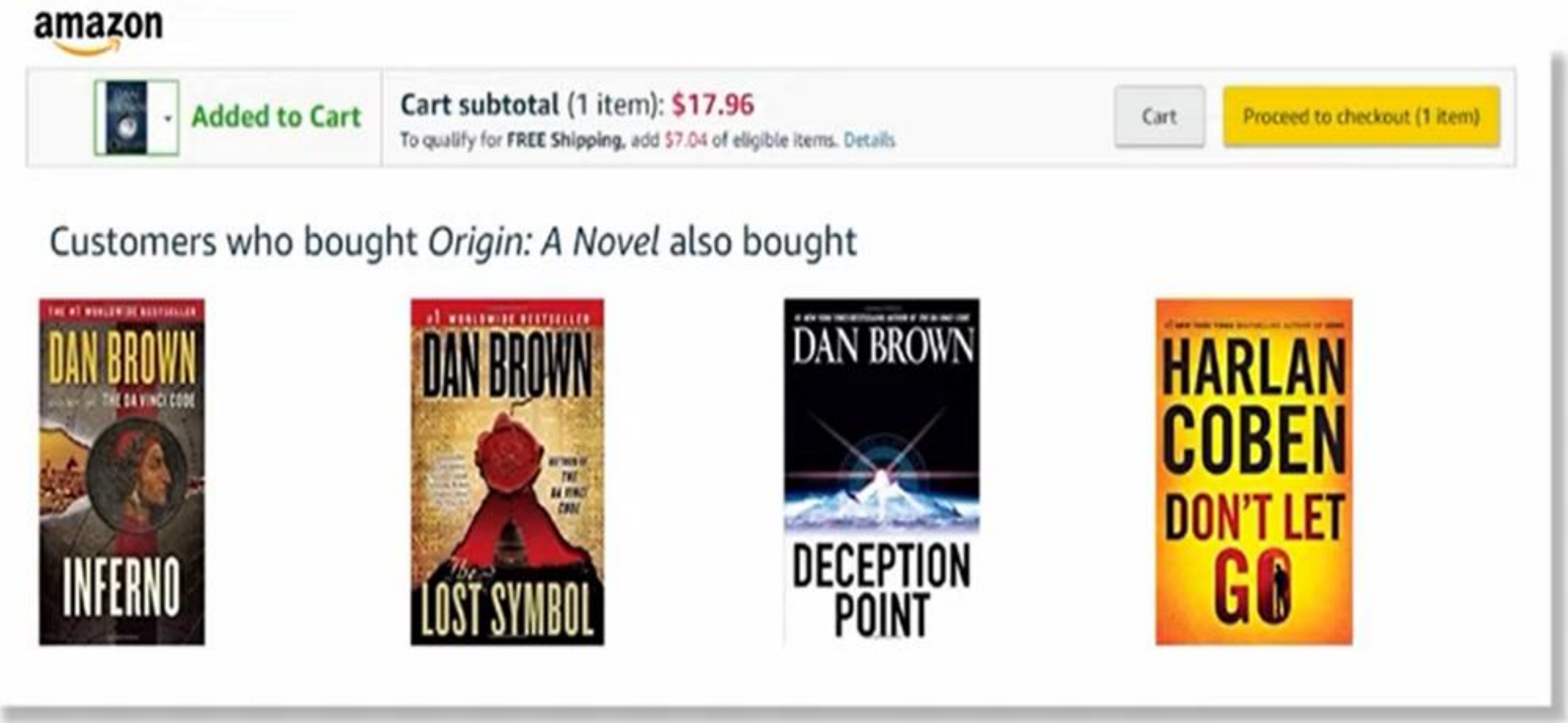
Artificial Intelligence in Practice

Amazon Echo

A home controlled AI-based device that responds to humans according to what they are saying.

Data Facilitates Recommendations

Amazon collects data from users and recommends the best products according to the user's buying/shopping patterns.



Introduction to AI

What is Artificial Intelligence?

According to the father of Artificial Intelligence,

- It is the **science and engineering** of **making intelligent machines**, especially intelligent computer programs.

- *John McCarthy*

Introduction to AI

VIT-AP UNIVERSITY

- Artificial Intelligence is a way of **making a computer, a computer-controlled robot, or a software think intelligently**, in the similar manner the intelligent humans think.

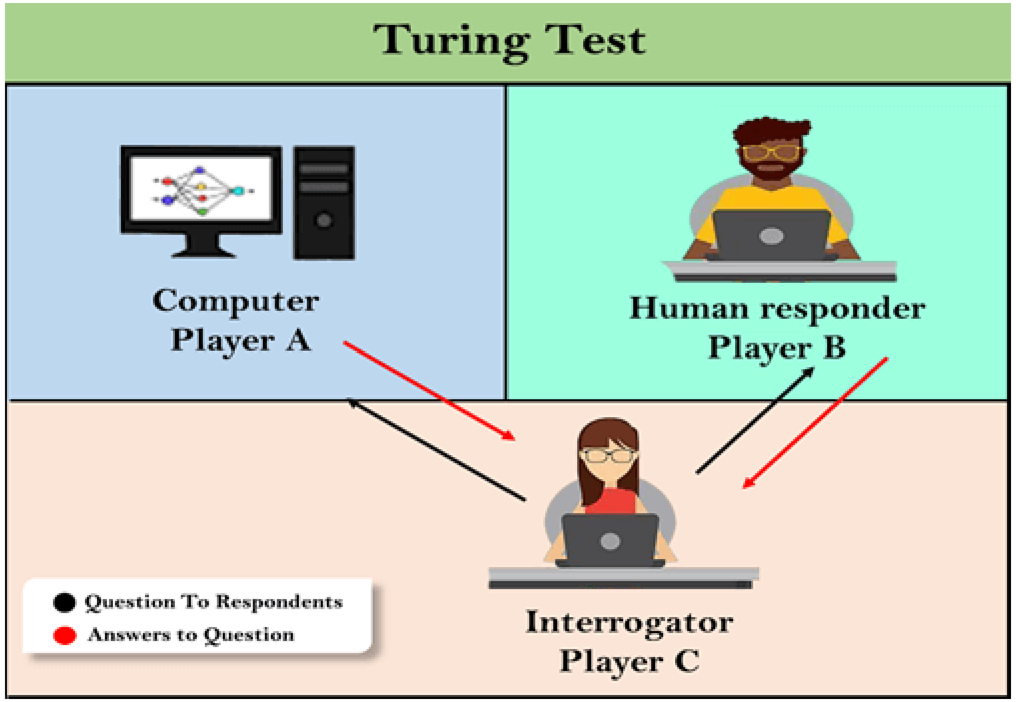
- Artificial Intelligence is the branch of computer science concerned with **making computers behave like humans**.

- ***<u>“The study and design of intelligent agents,"</u>***

Definitions / Approaches of AI

Systems that think like humans “The exciting new effort to make computers think … machines with minds, in the full and literal sense.”	**Systems that think rationally** “The study of mental faculties through the use of computer models.”
Systems that act like humans The art of creating machines that perform functions that require intelligence when performed by people.”	**Systems that act rationally** “Computational intelligence is the study of the design of intelligent agents.”

Acting humanly: Turing Test



- Alan Turing's 1950 article *Computing Machinery and Intelligence* discussed conditions for considering a machine to be intelligent

What would a computer need to pass the Turing test?

- **Natural language processing:** to communicate with examiner.
- **Knowledge representation:** to store and retrieve information provided before or during interrogation.
- **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.
- **Vision (for Total Turing test):** to recognize the examiner's actions and various objects presented by the examiner.
- **Motor control (total test):** to act upon objects as requested.
- **Other senses (total test):** such as audition, smell, touch, etc.

Thinking humanly: cognitive modeling

- Requires scientific theories of **internal activities of the brain**
- The interdisciplinary field of cognitive science brings together computer models from AI and experimental techniques from psychology to construct precise and testable theories of the human mind.

Thinking rationally: "laws of thought"

- **notation** and **rules** of **derivation for thoughts**; may or may not have proceeded to the idea of mechanization.
- **Direct line** through mathematics and philosophy to modern AI.
- These laws of thought were supposed to govern the operation of the mind; their study initiated the field called **logic**.

Acting rationally: rational agent

- Rational behavior: **doing the right thing**
- The right thing: that which is expected to **maximize goal achievement**, given the available information.
- A **rational agent** is one that acts so as to achieve the best outcome or, when there is uncertainty, the best expected outcome.

History of AI

- The birth of AI (1943 – 1956)
- Early enthusiasm (1952 – 1969)
- Emphasis on knowledge (1966 – 1974)
- Knowledge-based systems (1969 – 1999)
- AI became an industry (1980 – 1989)
- Current trends (1990 – present)

Foundation of AI

- Philosophy
- Computer Engineering
- Linguistics
- Biology
- Psychology
- Economics
- Mathematics

Sub-fields of AI

- Neural Networks
- Evolutionary Computation
- Vision
- Robotics
- Expert Systems
- Speech Processing
- Natural Language Processing
- Planning
- Machine Learning

Future Scope

- Cyber Security
- Face Recognition
- Data Analysis
- Transport
- Various Jobs
- Marketing & Advertising
- Artificial Intelligence Career Domains

Simple Definitions

- **Data**
  Data are the facts and figures about a particular activity.
- **Data Processing**
  The process of collecting all the required data together to produce meaningful information.
- **Information**
  Information is obtained by processing the data into meaningful form
- **Knowledge**
  It is structured representation of all the facts of an AI problem
- **Knowledge base**
  The facts or assertions about the problem domain.
- **Data base**
  The storage medium for the state variables.

Advantages of Artificial Intelligence

- **High Accuracy with less errors:** AI machines or systems are prone to less errors and high accuracy as it takes decisions as per pre-experience or information.
- **High-Speed:** AI systems can be of very high-speed and fast-decision making, because of that AI systems can beat a chess champion in the Chess game.
- **High reliability:** AI machines are highly reliable and can perform the same action multiple times with high accuracy.
- **Useful for risky areas:** AI machines can be helpful in situations such as defusing a bomb, exploring the ocean floor, where to employ a human can be risky.
- **Digital Assistant:** AI can be very useful to provide digital assistant to the users such as AI technology is currently used by various E-commerce websites to show the products as per customer requirement.
- **Useful as a public utility:** AI can be very useful for public utilities such as a self-driving car which can make our journey safer and hassle-free, facial recognition for security purpose, Natural language processing to communicate with the human in human-language, etc.

Disadvantages of Artificial Intelligence

- **High Cost:** The hardware and software requirement of AI is very costly as it requires lots of maintenance to meet current world requirements.
- **Can't think out of the box:** Even we are making smarter machines with AI, but still they cannot work out of the box, as the robot will only do that work for which they are trained, or programmed.
- **No feelings and emotions:** AI machines can be an outstanding performer, but still it does not have the feeling so it cannot make any kind of emotional attachment with human, and may sometime be harmful for users if the proper care is not taken.
- **Increase dependency on machines:** With the increment of technology, people are getting more dependent on devices and hence they are losing their mental capabilities.
- **No Original Creativity:** As humans are so creative and can imagine some new ideas but still AI machines cannot beat this power of human intelligence and cannot be creative and imaginative.

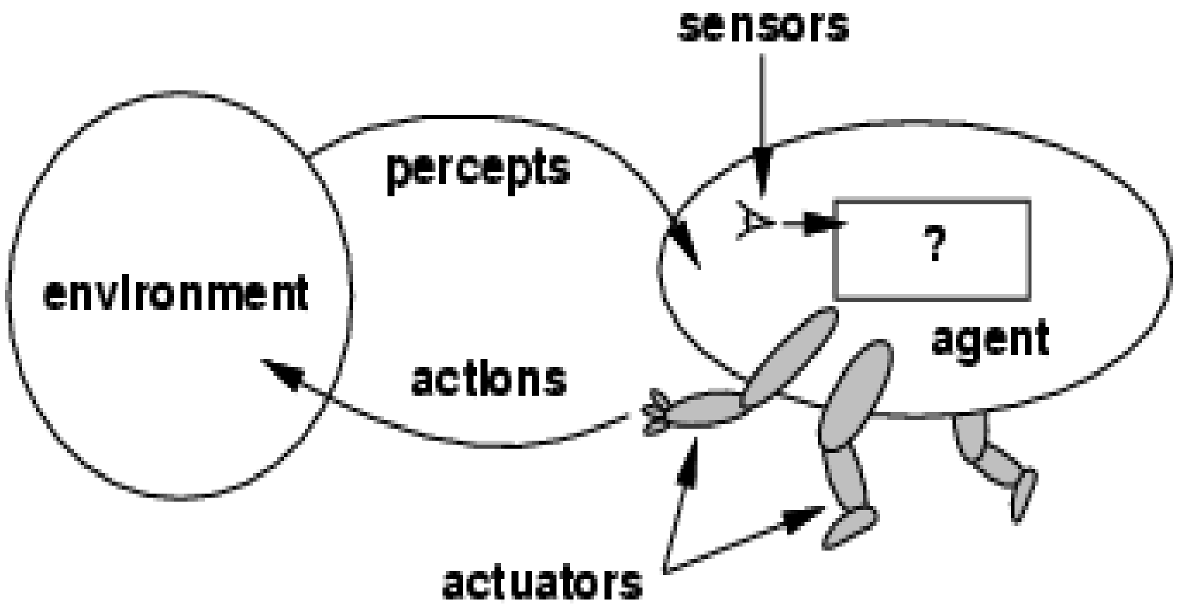
Prerequisite

- Any computer language such as C, C++, Java, Python, etc.(knowledge of Python will be an advantage)
- Knowledge of essential Mathematics such as derivatives, probability theory, etc.

Problem Solving Agents
&
Problem Formulation

Intelligent agent

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



- A <u>human agent </u>has eyes, ears, and other organs for sensors and hands, legs, mouth, and other body parts for actuators.

- A <u>robotic agent </u>might have cameras and infrared range finders for sensors and various motors for actuators

Percept

- We use the term **percept** to refer to the agent's perceptual inputs at any given instant.

Percept Sequence

- An agent's **percept sequence** is the complete history of everything the agent has ever perceived.

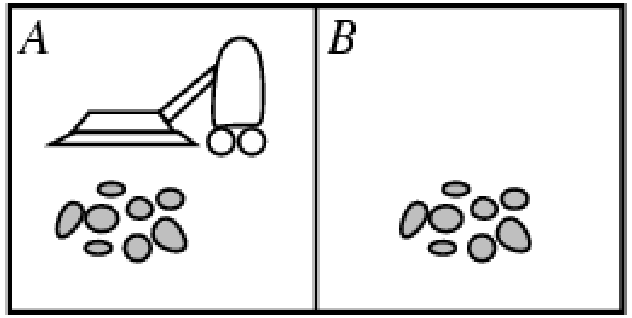
Agent function

- Mathematically speaking, we say that an agent's behavior is described by the **agent function** that maps any given percept sequence to an action.

$$f : \mathcal{P}^* \to \mathcal{A}$$

Example

- A vacuum-cleaner world with just two locations



- **Agent function**

Percept Sequence	**Action**
[A, Clean]	Right
[A, Dirty]	Work
[B, Clean]	Left
[B, Dirty]	Work
[A, Clean], [A, Clean]	Right
[A, Clean], [A, Dirty]	Work
…	
Figure: Partial tabulation of a simple agent functions for the vacuum-cleaner world	

Rational Agent

- A **rational agent** is one that does the right thing-conceptually speaking; every entry in the table for the agent function is filled out correctly.

Task environments

- **To design a rational agent we need to specify a *task environment***
    - a problem specification for which the agent is a solution

Specifying the task environment

The rationality of the simple vacuum-cleaner agent, needs specification of

- ***PEAS:*** *to* specify a task environment
    - ***P***erformance measure
    - ***E***nvironment
    - ***A***ctuators
    - ***S***ensors

PEAS: Specifying an automated taxi driver

***P*erformance measure:**

- ?

***E*nvironment:**

- ?

***A*ctuators:**

- ?

***S*ensors:**

- ?

PEAS: Specifying an automated taxi driver

Performance measure:

- safe, fast, legal, comfortable, maximize profits

Environment:

- roads, other traffic, pedestrians, customers

Actuators:

- steering, accelerator, brake, signal, horn

Sensors:

- cameras, sonar, speedometer, GPS

PEAS: Medical diagnosis system

Performance measure**:**

- ?

Environment**:**

- ?

Actuators**:**

- ?

Sensors**:**

- ?

PEAS: Medical diagnosis system

- ***P****erformance measure*: Healthy patient, minimize costs, lawsuits
- ***E****nvironment*: Patient, hospital, staff
- ***A****ctuators*: Screen display (form including: questions, tests, diagnoses, treatments, referrals)
- ***S****ensors*: Keyboard (entry of symptoms, findings, patient's answers)

Agent Type	Performance Measure	Environments	Actuators	Sensors
Taxi driver	Safe: fast, legal, comfortable trip, maximize profits	Roads,other traffic,pedestrians, customers	Steering,accelerator, brake, Signal,horn,display	Cameras,sonar, Speedometer,GPS, Odometer,engine sensors,keyboards, accelerometer

Figure: PEAS description of the task environment for an automated taxi.

Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, minimize costs, lawsuits	Patient, hospital, staff	Display 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 categorization of scene	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	Maximize purity, yield, safety	Refinery, operators	Valves, pumps, heaters, displays	Temperature, pressure, chemical sensors
Interactive English tutor	Maximize student's score on test	Set of students, testing agency	Display exercises, suggestions, corrections	Keyboard entry

Problem Solving Agents
&
Problem Formulation

PROBLEMS, PROBLEM SPACES AND SEARCH

To solve the problem of building a system you should take the following steps:

1. Define the problem accurately including detailed specifications and what constitutes a suitable solution.
2. Scrutinize the problem carefully, for some features may have a central affect on the chosen method of solution.
3. Segregate and represent the background knowledge needed in the solution of the problem.
4. Choose the best solving techniques for the problem to solve a solution.

VIT-AP UNIVERSITY

***Problem solving* is a process** of generating solutions from observed data.

- a *'problem'* is characterized by a set of *goals*,
- a set of *objects*, and
- a set of *operations*.

These could be ill-defined and may evolve during problem solving.

- A **'*problem space*'** is an abstract space.
  - ✓ A problem space encompasses all *valid states* that can be generated by the application of any combination of *operators* on any combination of *objects*.
  - ✓ The problem space may contain one or more *solutions*. A solution is a combination of *operations* and *objects* that achieve the *goals*.
- A '***search***' refers to the search for a solution in a problem space.
  - ✓ Search proceeds with different types of '*search control strategies*'.
  - ✓ The *depth-first search and breadth-first search* are the two common *search strategies.*

Problem Solving - Definition

- Problem solving is a process of generating solutions from observed data.
- Problem is characterized by a set of goals, set of objects, and set of operations.
- The method of solving problem through AI involves the process of
  - defining the search space,
  - deciding start and goal states
  - finding the path from start state to goal state through search space.

These are the following steps which require to solve a problem :

Goal Formulation:

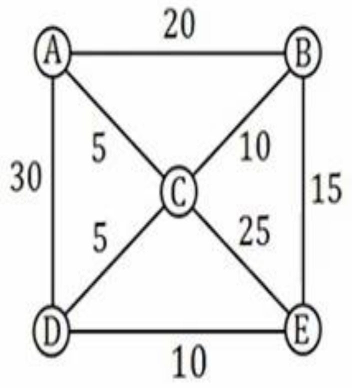
This one is the first and simple step in problem-solving. It organizes finite steps to formulate a target/goals which require some action to achieve the goal. Today the formulation of the goal is based on AI agents.

Problem formulation:

It is one of the core steps of problem-solving which decides **what action should be taken to achieve the formulated goal**. In AI this core part is dependent upon software agent which consisted of the following components to formulate the associated problem.

Goal Formulation and Problem Formulation

- Goal Formulation:
  - Problem solving is about having a goal we want to reach, (i.e. I want to travel from 'A' to 'E').
  - Goal is a set of environment states in which the goal is satisfied.
- Problem Formulation:
  - A problem formulation is about deciding what actions and states to consider,
  - describe states as "in (CITYNAME)" where CITYNAME is the name of the city in which we are currently in.



- The process of finding such sequence is called search, a search algorithm is like a black box which takes problem as input returns a solution, and once the solution is found the sequence of actions carried out, and this is called the execution phase.

Components to formulate the associated problem:

Initial State: This state requires an initial state for the problem which starts the AI agent towards a specified goal. In this state new methods also initialize problem domain solving by a specific class.

Action: This stage of problem formulation works with function with a specific class taken from the initial state and all possible actions done in this stage.

Transition: This stage of problem formulation integrates the actual action done by the previous action stage and collects the final stage to forward it to their next stage.

Goal test: This stage determines that the specified goal achieved by the integrated transition model or not, whenever the goal achieves stop the action and forward into the next stage to determines the cost to achieve the goal.

Path costing: This component of problem-solving numerical assigned what will be the cost to achieve the goal. It requires all hardware software and human working cost.

Important Terminologies

- Transition Model
- State Space
- Graph
- Path
- Path Cost
- Step Cost
- Optimal Solution

Toy and Real world Problems

TOY PROBLEMS

- Vacuum World Example
- The 8-puzzle
- 8-queens problem

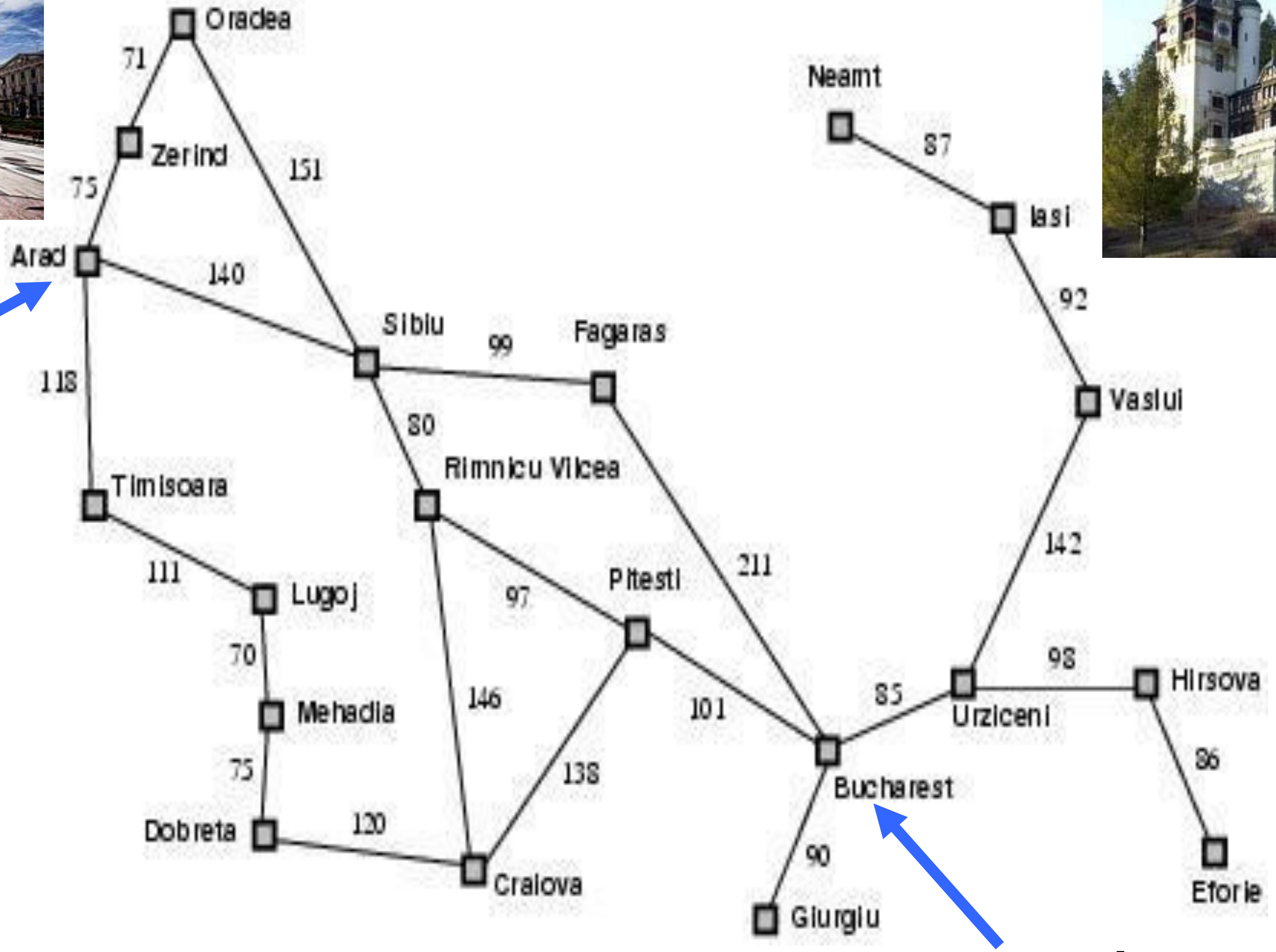
REAL-WORLD PROBLEMS

ROUTE-FINDING PROBLEM - AIRLINE TRAVEL PROBLEM

TOURING PROBLEMS

- THE TRAVELLING SALES PERSON PROBLEM (TSP)
- VLSI layout
- ROBOT navigation
- AUTOMATIC ASSEMBLY SEQUENCING
- INTERNET SEARCHING

Example search problem: holiday in Romania



You are here

You need to be here

Holiday in Romania II

- **On holiday in Romania; currently in Arad**
  - Flight leaves tomorrow from Bucharest
- **Formulate *goal***
- **Formulate *search problem***
- **Find *solution***

  …

Holiday in Romania II

- **On holiday in Romania; currently in Arad**
  - Flight leaves tomorrow from Bucharest
- **Formulate *goal***
  - Be in Bucharest
- **Formulate *search problem***
  - States: various cities
  - Actions: drive between cities
  - Performance measure: minimize distance
- **Find *solution***
  - Sequence of cities; e.g. Arad, Sibiu, Fagaras, Bucharest, …

More formally, a problem is defined by:

1. A set of *states* $\mathbf{S}$
2. *An initial state* $s_i \in S$
3. *A set of actions* $\mathbf{A}$
   - $\forall s$ **Actions(s)** = *the set of actions that can be executed in* **s**, *that are applicable* in **s**.
4. *Transition Model:* $\forall s \forall a \in Actions(s)$ $Result(s, a) \rightarrow s_r$
   - s_r *is called a successor of* **s**
   - $\{s_i\} \cup Successors(s_i)^*$ = *state space*
5. *Goal test* ***Goal(s)***
   - Can be implicit, e.g. ***checkmate(x)***
   - ***s*** is a *goal state* if ***Goal(s)*** is *true*
6. *Path cost* (additive)
   - e.g. sum of distances, number of actions executed, …
   - ***c(x,a,y)*** is the step cost, assumed ≥ 0
     - (where action ***a*** goes from state ***x*** to state ***y***)

Solution

A *solution* is a sequence of actions from the *initial state* to a *goal state.*

Optional Solution:

A solution is *optimal* if no solution has a lower path cost.

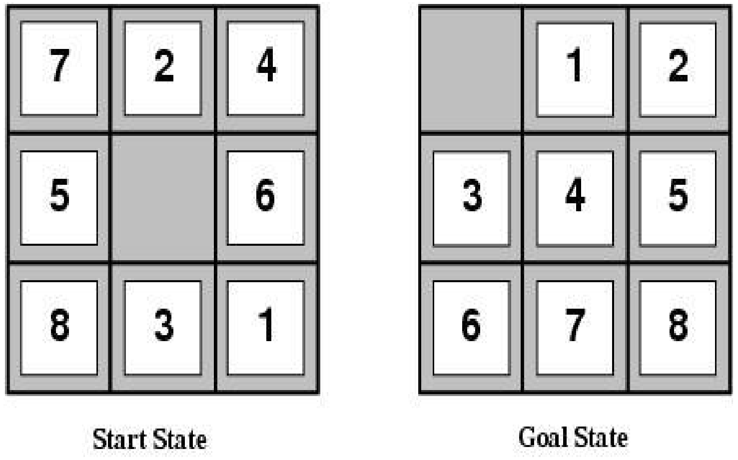
Art: Formulating a Search Problem

Decide:

- **Which properties matter & how to represent**
  - *Initial State, Goal State, Possible Intermediate States*
- **Which actions are possible & how to represent**
  - *Operator Set: Actions and Transition Model*
- **Which action is next**
  - *Path Cost Function*

Formulation greatly affects combinatorics of search space and therefore speed of search

Example: 8-puzzle



- States??
- Initial state??
- Actions??
- Transition Model??
- Goal test??
- Path cost??

Example: 8-puzzle

7	2	4
5		6
8	3	1

Start State

	1	2
3	4	5
6	7	8

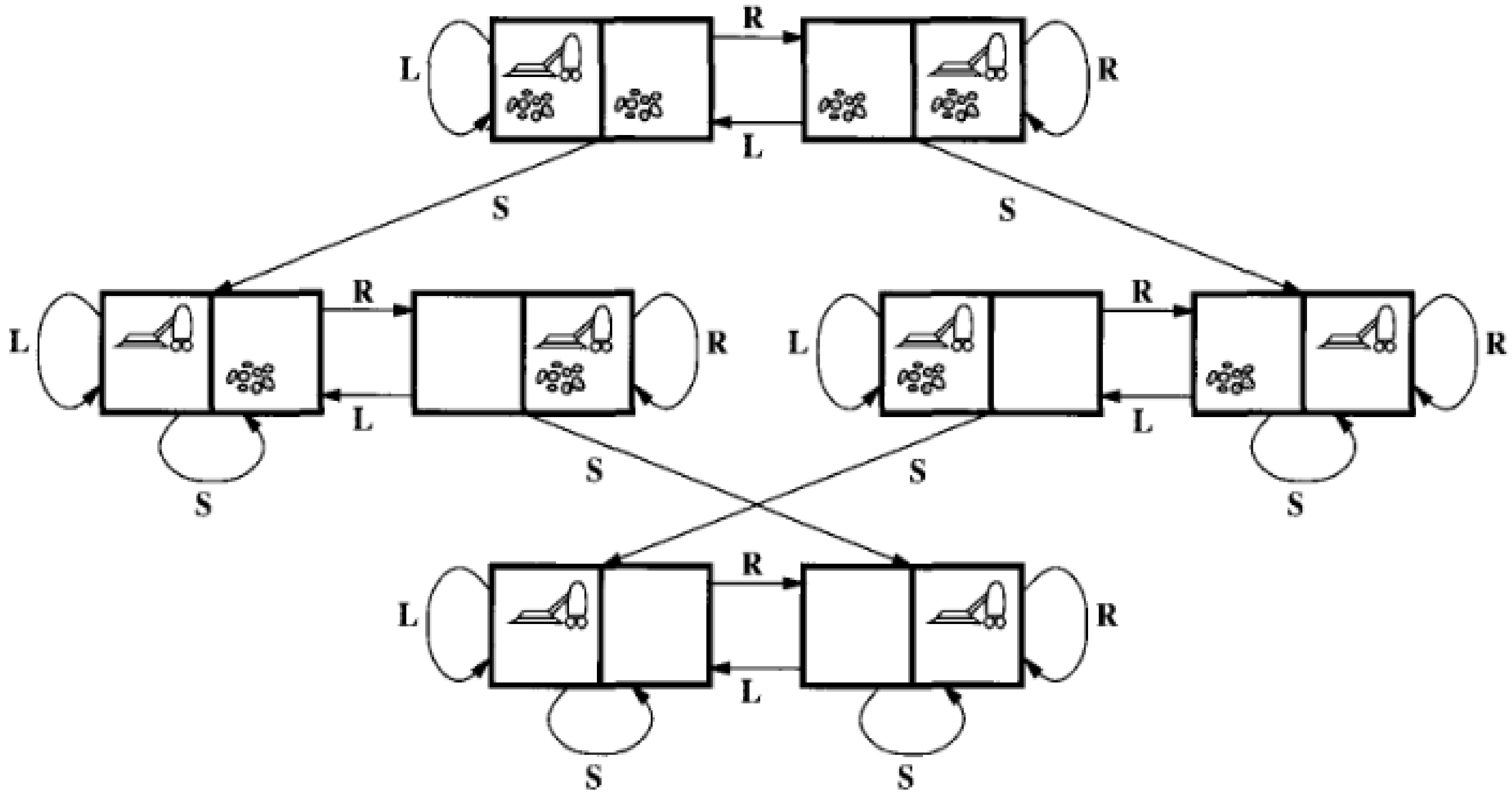
Goal State

- **States??** **List of 9 locations- e.g., [7,2,4,5,-,6,8,3,1]**
- **Initial state??** **[7,2,4,5,-,6,8,3,1]**
- **Actions??** ***{Left, Right, Up, Down}***
- **Transition Model??** **...**
- **Goal test??** **Check if goal configuration is reached Number**
- **Path cost??** **of actions to reach goal (1 per move)**

Vacuum World Example

- **States**: The agent is in one of two locations.,each of which might or might not contain dirt. Thus there are 2 x 2^2 = 8 possible world states.
- **Initial state**: Any state can be designated as initial state.
- **Successor function** : This generates the legal states that results from trying the three actions (left, right, suck). The complete state space is shown in figure 2.3
- **Goal Test** : This tests whether all the squares are clean.
- **Path test** : Each step costs one ,so that the path cost is the number of steps in the path.

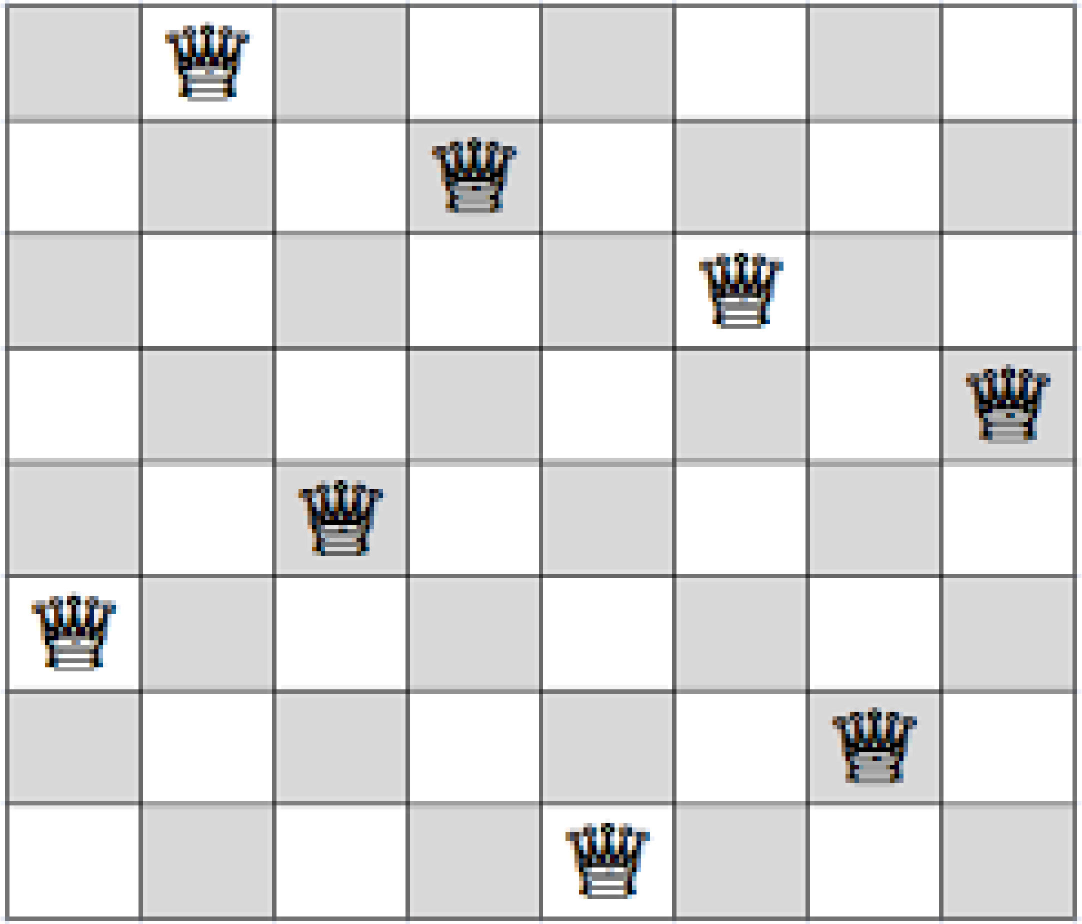
Vacuum World Example



8-queens problem

- **States** : Any arrangement of 0 to 8 queens on board is a state.
- **Initial state** : No queen on the board.
- **Successor function** : Add a queen to any empty square.
- **Goal Test** : 8 queens are on the board, none attacked.

8-queens problem



Problem Definition

Problem Definition

A problem is defined by its '*elements*' and their '*relations*'. To provide a formal description of a problem, we need to do the following:

a. Define a *state space* that contains all the possible configurations of the relevant objects, including some impossible ones.
b. Specify one or more states that describe possible situations, from which the problem-solving process may start. These states are called *initial states*.
c. Specify one or more states that would be acceptable solution to the problem.

These states are called *goal states*.

Problem Definition

Specify a set of *rules* that describe the actions (*operators*) available.

The problem can then be solved by using the *rules*, in combination with an appropriate *control strategy*, to move through the *problem space* until a *path* from an *initial state* to a *goal state* is found. This process is known as ***'search'***. Thus:

- *Search* is fundamental to the problem-solving process.
- *Search* is a general mechanism that can be used when a more direct method is not known.
- *Search* provides the framework into which more direct methods for solving subparts of a problem can be embedded. A very large number of AI problems are formulated as search problems.
- Problem space

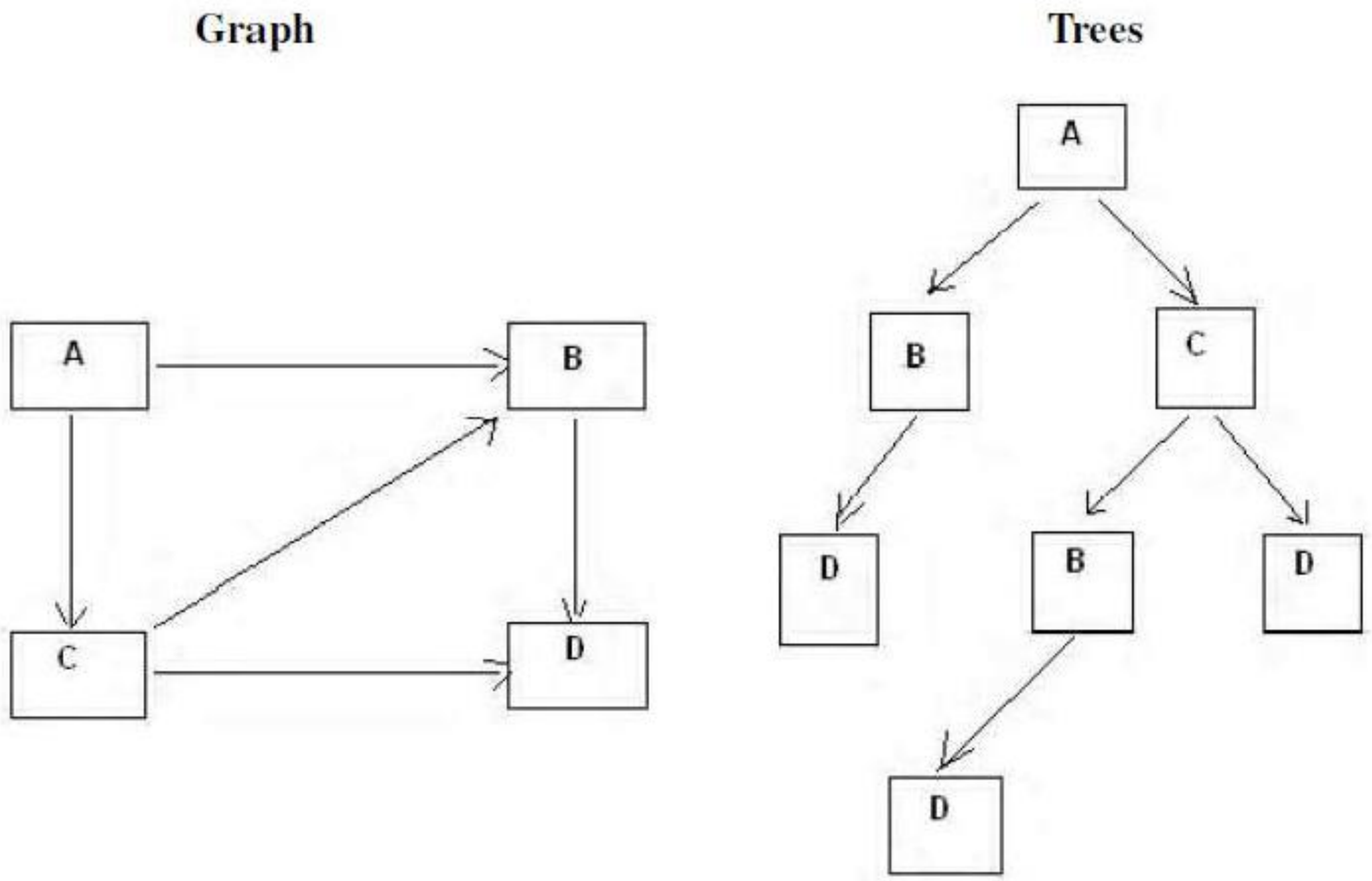
Problem Definition

A *problem space* is represented by a directed graph, where *nodes* represent search state and *paths* represent the operators applied to change the *state*.

To simplify search algorithms, it is often convenient to logically and programmatically represent a problem space as a **tree**. A *tree* usually decreases the complexity of a search at a cost. Here, the cost is due to duplicating some nodes on the tree that were linked numerous times in the graph, e.g. node ***B*** and node ***D***.

A *tree is a graph* in which any two vertices are connected by exactly one path. Alternatively, any connected *graph with no cycles is a tree*.

Problem Definition



DEFINING PROBLEM AS A STATE SPACE SEARCH

VIT-AP UNIVERSITY

- To solve the problem of playing a game, we require the rules of the game and targets for winning as well as representing positions in the game.
- The opening position can be defined as the initial state and a winning position as a goal state.
- Moves from initial state to other states leading to the goal state follow legally.

State Space Search

A *state space* represents a problem in terms of *states* and *operators* that change states.
A state space consists of:

- A representation of the *states* the system can be in. For example, in a board game, the board represents the current state of the game.
- A set of *operators* that can change one state into another state. In a board game, the operators are the legal moves from any given state. Often the operators are represented as programs that change a state representation to represent the new state.
- An *initial state*.
- A set of *final states*; some of these may be desirable, others undesirable. This set is often represented implicitly by a program that detects terminal states.

PRODUCTION SYSTEMS

What is Production System?

Production system or production rule system is a computer program typically used to provide some form of artificial intelligence, which consists primarily of a set of rules about behavior but it also includes the mechanism necessary to follow those rules as the system responds to states of the world.


SMILE! OUR AI WILL TAKE A PICTURE OF YOUR FACE AND WILL KNOW EXACTLY WHAT YOU WOULD LIKE FOR DINNER TONIGHT.
KDNUGGETS.COM

Production systems provide appropriate structures for performing and describing search processes. A production system has four basic components as enumerated below.

- set of rules

  A set of rules each consisting of a left side that determines the applicability of the rule and a right side that describes the operation to be performed if the rule is applied.

  Ex. C→A (LHS-Condition and RHS-Action)

- Knowledge base / Global Database

  The global database is the central data structure used by the production system in Artificial Intelligence.

- control strategy

  A control strategy then chooses which applicable rule should be applied and ceases computation when a termination condition on the database is satisfied. If multiple rules are to fire at the same time, the control system resolves the conflicts.

- Rule Applier
  - A rule firing module.

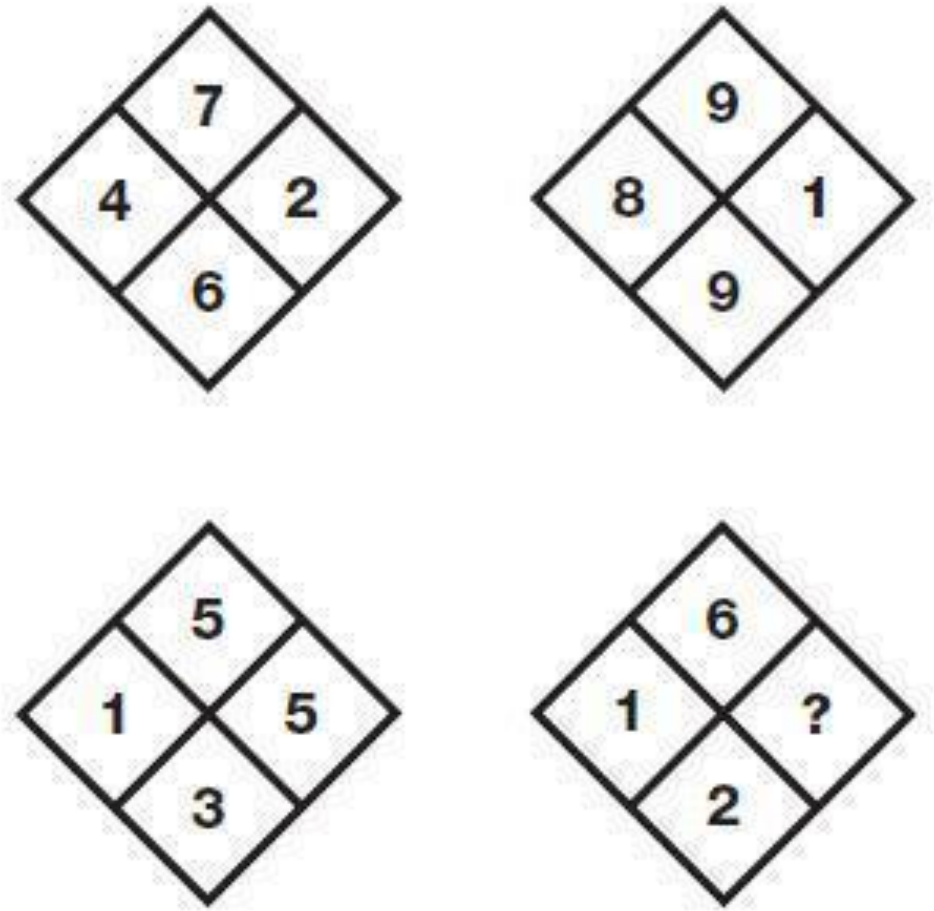
Water jug problem

- Consider a water jug Problem:

  You are given two jugs, a 4-gallon one and a 3-gallon one. Neither have any measuring markers on it. There is a pump that can be used to fill the jugs with water. How can you get exactly 2 gallons of water into the 4-gallon jug?

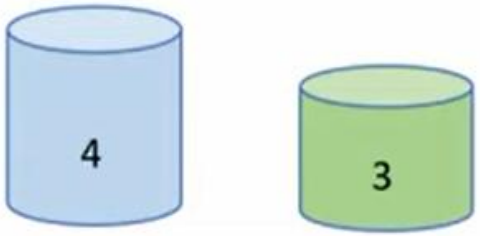
- Explicit Assumptions:

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


7
4
2
6
9
8
1
9
5
1
5
3
6
1
?
2

Water Jug Problem

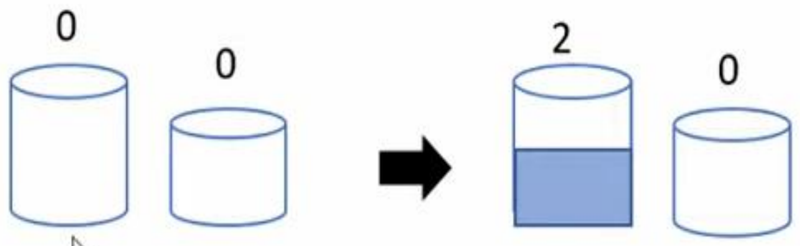
- 4-gallon one and a 3-gallon Jug



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

State Representation

- X represents the quantity of water in the 4-gallon jug **X= 0,1,2,3,4**
- Y represents the quantity of water in 3-gallon jug **Y=0,1,2,3**
- **Start State: (0,0)**
- **Goal State: (2,0)**



- **Operations**: 1.Empty a Jug

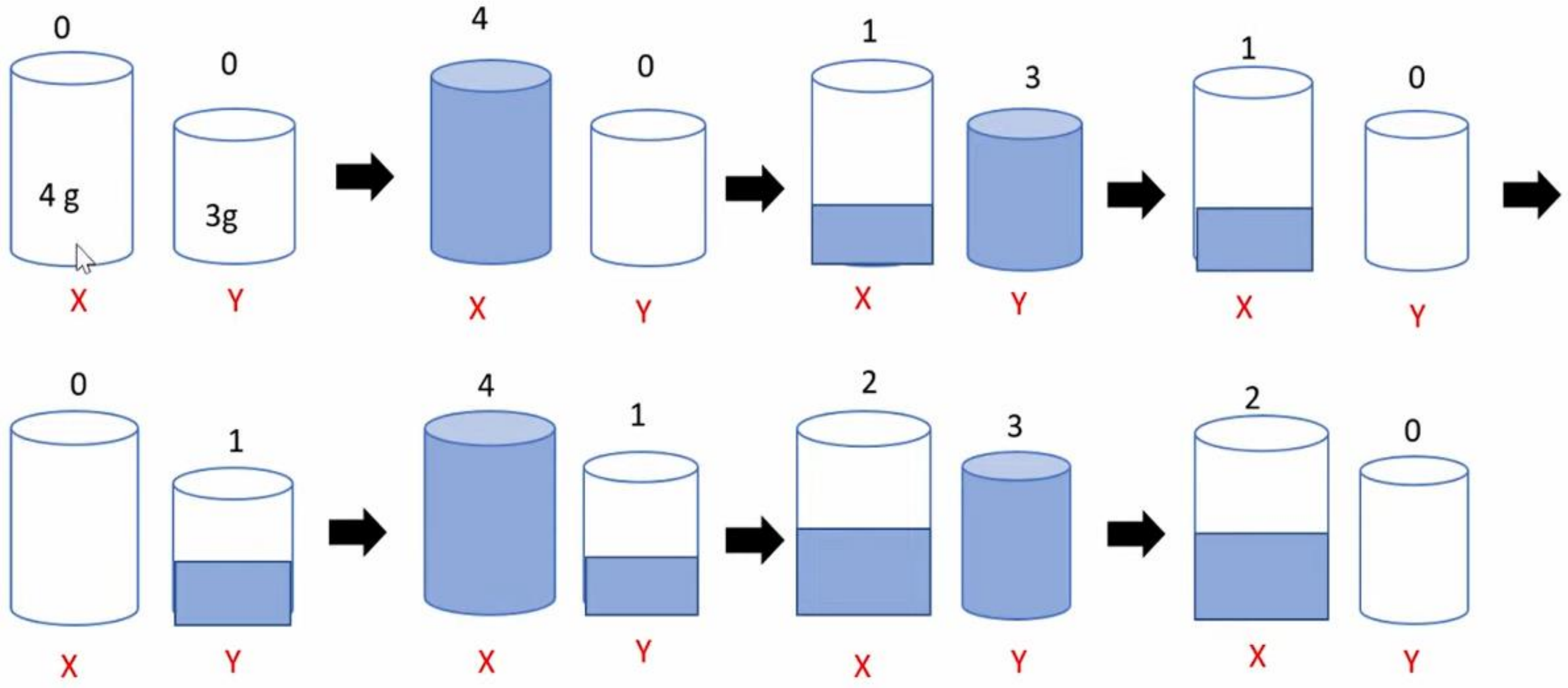
  2.Fill a Jug

  3.Pour water from one jug to the other

Production Rules

Rule	Initial State	Condition	Final state	Description of the action taken
1	(x,y)	If x<4	(4,y)	Fill the 4 gallon jug completely
2	(x,y)	if y<3	(x,3)	Fill the 3 gallon jug completely
3	(x,y)	If x>0	(x-d,y)	Pour some water from the 4 gallon jug
4	(x,y)	If y>0	(x,y-d)	Pour some water from the 3 gallon jug
5	(x,y)	If x>0	(0,y)	Empty the 4 gallon jug
6	(x,y)	If y>0	(x,0)	Empty the 3 gallon jug
7	(x,y)	If (x+y)<7	(4, y-[4-x])	Pour some water from the 3 gallon jug to fill the four gallon jug
8	(x,y)	If (x+y)<7	(x-[3-y],3)	Pour some water from the 4 gallon jug to fill the 3 gallon jug.
9	(x,y)	If (x+y)<4	(x+y,0)	Pour all water from 3 gallon jug to the 4 gallon jug
10	(x,y)	if (x+y)<3	(0, x+y)	Pour all water from the 4 gallon jug to the 3 gallon jug

Solution



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	1. Fill 4
3.	1	3	6.Pour 4 into 3 to fill
4.	1	0	4. Empty 3
5.	0	1	8. Pour all of 4 into 3
6.	4	1	1. Fill 4
7.	2	3	6. Pour 4 into 3

Types of Production Systems

- **Monotonic Production System**:
  - It's a production system in which the application of a rule never prevents the later application of another rule, that could have also been applied at the time the first rule was selected.

- **Non-Monotonic Production Systems**:
  - These are useful for solving ignorable problems. These systems are important from an implementation standpoint because they can be implemented without the ability to backtrack to previous states when it is discovered that an incorrect path was followed. This production system increases efficiency since it is not necessary to keep track of the changes made in the search process.

Types of Production Systems

- **Partially Commutative Production System**:

  It's a type of production system in which the application of a sequence of rules transforms state X into state Y, then any permutation of those rules that is allowable also transforms state x into state Y. Theorem proving falls under the monotonic partially communicative system.

- **Commutative Systems**:

  These are usually useful for problems in which changes occur but can be reversed and in which the order of operation is not critical. Production systems that are not usually not partially commutative are useful for many problems in which irreversible changes occur, such as chemical analysis. When dealing with such systems, the order in which operations are performed is very important and hence correct decisions must be made at the first attempt itself.

Ques:- Describe a Production System in AI.

↳ Helps in Structuring AI Programs in a way that facilitates describing and performing the Search Process.

Production System Consists of:-
- (i) Set of Rules
- (ii) Knowledge Base
- (iii) Control Strategy
- (iv) Rule Applier.

Steps to Solve the Problem:-

↳ i) first reduce Problem so that it can be shown in a precise statement.
ii) Problem can be solved by Searching a path through Space. [Start → Goal]
iii) Solving process can be modelled.

Advantages:-

(i) Structuring AI Problems.] Excellent tool.
(ii) Highly modular.] rules – add, remove, change.
(iii) Rules are expressed in natural form] Easy to understand.

Characteristics:-

↳ i) Monotonic PS: Appⁿ of a rule never prevents later appⁿ of another rule.] Rules are independent.

ii) Non-Monotonic PS: which this is not true.

iii) Partially Commutative PS: $(r1, r2)$ $x \rightarrow y$
Permutation ↳ $(x \rightarrow y)$

iv) Commutative PS: Both monotonic and Partially Commutative.

I speak without a mouth and hear without ears. I have no body, but I come alive with wind. What am I?

-**echo**

You see a boat filled with people. It has not sunk, but when you look again you don’t see a single person on the boat. Why?

-**All the people were married.**

the frog climbs up 3ft every day but slips 2ft during night the well is 20ft how long will it take him to reach the top

Control Strategy

Control Strategy in Artificial Intelligence scenario is a technique or strategy, tells us about which rule has to be applied next while searching for the solution of a problem within problem space.

It helps us to decide which rule has to apply next without getting stuck at any point. These rules decide the way we approach the problem and how quickly it is solved and even whether a problem is finally solved.

Control Strategy helps to find the solution when there is more than one rule or fewer rules for finding the solution at each point in problem space.

A good Control strategy has two main characteristics:

Control Strategy

Control Strategy should cause Motion

Each rule or strategy applied should cause the motion because if there will be no motion than such control strategy will never lead to a solution. Motion states about the change of state and if a state will not change then there be no movement from an initial state and we would never solve the problem.

Control strategy should be Systematic

Though the strategy applied should create the motion but if do not follow some systematic strategy than we are likely to reach the same state number of times before reaching the solution which increases the number of steps. Taking care of only first strategy we may go through particular useless sequences of operators several times. Control Strategy should be systematic implies a need for global motion (over the course of several steps) as well as for local motion (over the course of single step).

Search Strategies:

- Every stage in a state space generating algorithm that we need to apply to reach goal stage
- Search is the systematic examination of states to find paths from the start/root state to the goal state.
- Many traditional search algorithms are used in AI applications
- Consequently, many special techniques are developed; using heuristic functions. The algorithms that use heuristic functions are called heuristic algorithms.
- Heuristic algorithms are not really intelligent; they appear to be intelligent because they achieve better performance.
- Heuristic algorithms are more efficient because they take advantage of feedback from the data to direct the search path.

There are mainly two types of search strategies:

1. Uninformed search (blind search).

Also called blind, exhaustive or brute-force search uses no information about the problem to guide the search and therefore may not be very efficient.

2. Informed search (heuristic search)

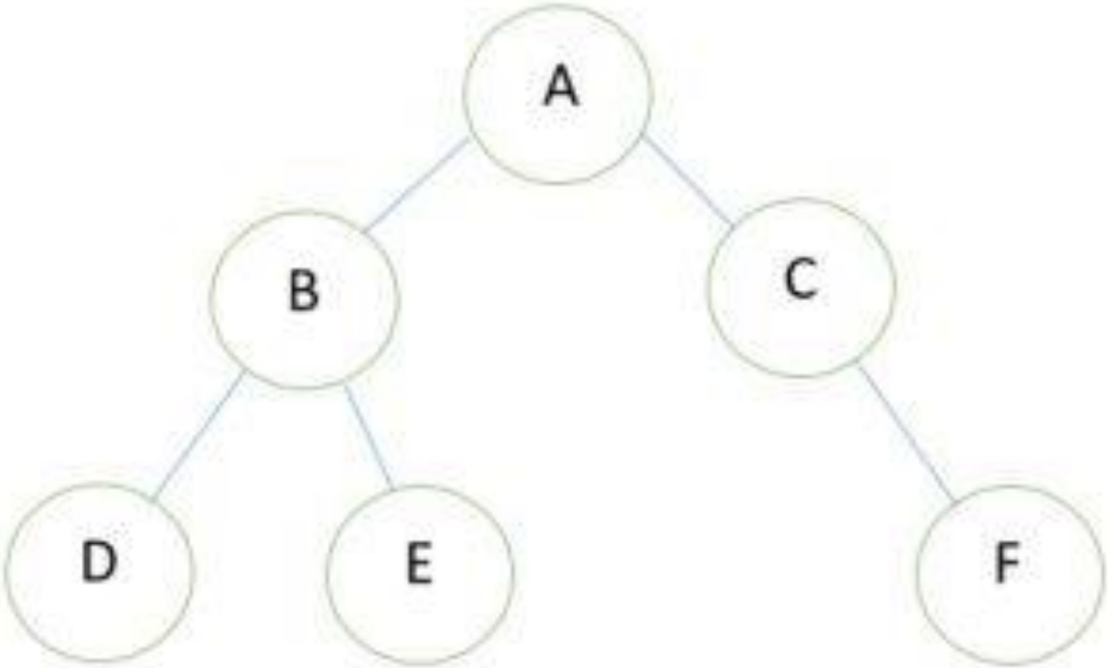
Also called heuristic or intelligent search, uses information about the problem to guide the search, usually guesses the distance to a goal state, and is therefore efficient, but the search may not be always possible.

Examples:

- Breadth-First Search: It searches along the breadth and follows first-in-first-out queue data structure approach. It will start to scan node A first and then B-C-D-E-F.

- Depth-First Search: It searches along the depth and follows the stack approach. The sequence for scanning nodes will be A-B-D-E-C-F, it scans all the sub-nodes of parent nodes and then moves to another node.

- Widely used Control Strategies are Breadth-First Search, Depth-First Search, Generate and Test, Hill-Climbing, Best-first search, Problem Reduction and many more.



Problem Characteristics

- To choose the most appropriate method
- Its necessary to analyse the problem

- Artificial Intelligence is a "way of making a computer, a computer-controlled robot, or software think intelligently, in the similar manner the intelligent humans think".

- Since artificial intelligence (AI) is mainly related to the search process, it is important to have some methodology to choose the best possible solution.

- To choose an appropriate method for a particular problem first we need to categorize the problem based on the following characteristics.

Problem Characteristics

VIT-AP UNIVERSITY

1. Is the problem Decomposable?
2. Can solution steps to be ignored or undone?
3. Is the problem's universe predictable?
4. Is the good solution is absolute or relative?
5. Is the solution a state or a path?
6. What is the role of knowledge?
7. Does the task require interaction with a person?

Is the problem Decomposable?

- Decomposable problem:

Is the problem Decomposable?

- Non - Decomposable problem:

Problem Characteristics

1. Is the problem Decomposable?
2. Can solution steps to be ignored or undone?
3. Is the problem's universe predictable?
4. Is the good solution is absolute or relative?
5. Is the solution a state or a path?
6. What is the role of knowledge?
7. Does the task require interaction with a person?

Can solution steps to be ignored or undone?

- Consider following 3 problems

1: proving a theorem or lemma

  – some steps can be ignored

  – logically derives to a solution

Can solution steps to be ignored or undone?

VIT-AP UNIVERSITY

- Consider following 3 problems

2: 8 puzzle problem

1	3	4
8	6	2
7		5

Initial State

1	3	4
8	6	2
7	5	

1	3	4
8	6	
7	5	2

1	2	3
8		4
7	6	5

Goal State

- Recoverable problem

Can solution steps to be ignored or undone?

- Consider following 3 problems

3: chess problem

- Irrecoverable problem

Can solution steps to be ignored or undone?

- Proving a theorem or lemma
  - Ignorable
- 8 puzzle problem
  - Recoverable
- Chess game
  - Irrecoverable

Recovery of the problem plays an important role in determining the complexity of the control structure

Problem Characteristics

1. Is the problem Decomposable?
2. Can solution steps to be ignored or undone?
3. Is the problem’s universe predictable?
4. Is the good solution is absolute or relative?
5. Is the solution a state or a path?
6. What is the role of knowledge?
7. Does the task require interaction with a person?

Is the problem's universe predictable?

- 8 puzzle problem – next step is always predictable – normal planning - certain outcome

1	3	4
8	6	2
7		5

1	3	4
8	6	2
7	5	

- Card game – next step is hidden – use probability for finding – uncertain problem

Problem Characteristics

VIT-AP UNIVERSITY

1. Is the problem Decomposable?
2. Can solution steps to be ignored or undone?
3. Is the problem's universe predictable?
4. Is the good solution is absolute or relative?
5. Is the solution a state or a path?
6. What is the role of knowledge?
7. Does the task require interaction with a person?

Is the good solution is absolute or relative?

- Consider this example: Given some facts

1. Deena is a man.
2. Deena is a worker in a company.
3. Deena was born in 1905.
4. All men are mortal.
5. All workers in a factory died when there was an accident in 1952.
6. No mortal lives longer than 100 years.

"Is Deena alive"

Is the good solution is absolute or relative?

1. Deena is a man.
2. Deena is a worker in a company.
3. Deena was born in 1905.
4. All men are mortal.
5. All workers in a factory died when there was an accident in 1952.
6. No mortal lives longer than 100 years.

"Is Deena alive"

Solution 1:
1. Deena is a man.
2. Deena was born in 1915.
3. All men are mortal.
4. Now it is 2020, so Siva's age is 105 years.
5. No mortal lives longer than 100 years.

Solution 2:
1. Deena is a worker in the company.
2. All workers in the company died in 1952.

Any path problem – Relative solutions

Is the good solution is absolute or relative?

Consideration 2 : Travelling Salesman problem

Travelling salesman problem: Goal : shortest path from source city to destination city – visiting all cities one by one

Solution : Shortest path – to visit all cities

Is the good solution is absolute or relative?

Solution:

Any path problem – Relative solution

Best path problem – Absolute Solution

Problem Characteristics

1. Is the problem Decomposable?
2. Can solution steps to be ignored or undone?
3. Is the problem's universe predictable?
4. Is the good solution is absolute or relative?
5. Is the solution a state or a path?
6. What is the role of knowledge?
7. Does the task require interaction with a person?

Is the solution a state or a path?

Consideration 1: Inference from the statement

"The bank president ate a dish of pasta salad with the fork"

Inference: pasta salad was a dish , pasta salad was eaten, pasta salad consists of pasta….

Consideration 2 : Path problem

Water jug problem: Goal : (2,0)

Solution : path taken to reach goal state from initial state

Solution: path taken from (0,0) – (2,0)

Problem Characteristics

1. Is the problem Decomposable?
2. Can solution steps to be ignored or undone?
3. Is the problem's universe predictable?
4. Is the good solution is absolute or relative?
5. Is the solution a state or a path?
6. What is the role of knowledge?
7. Does the task require interaction with a person?

What is the role of knowledge?

- **Chess playing**

– rules for determining legal moves + some simple control mechanisms

- **News paper story**

– scan all daily newspapers + how many supports Modi jee + how many supports Soina jee for upcoming election

Problem Characteristics

VIT-AP UNIVERSITY

1. Is the problem Decomposable?
2. Can solution steps to be ignored or undone?
3. Is the problem’s universe predictable?
4. Is the good solution is absolute or relative?
5. Is the solution a state or a path?
6. What is the role of knowledge?
7. Does the task require interaction with a person?

Does the task require interaction with a person?

- **Solitary**:
  - Computer is given with a problem description
  - no intermediate communication
  - no demand for an explanation
- **Conversational**:
  - Intermediate conversations between person to computer
  - User need to provide additional information
  - Robotics

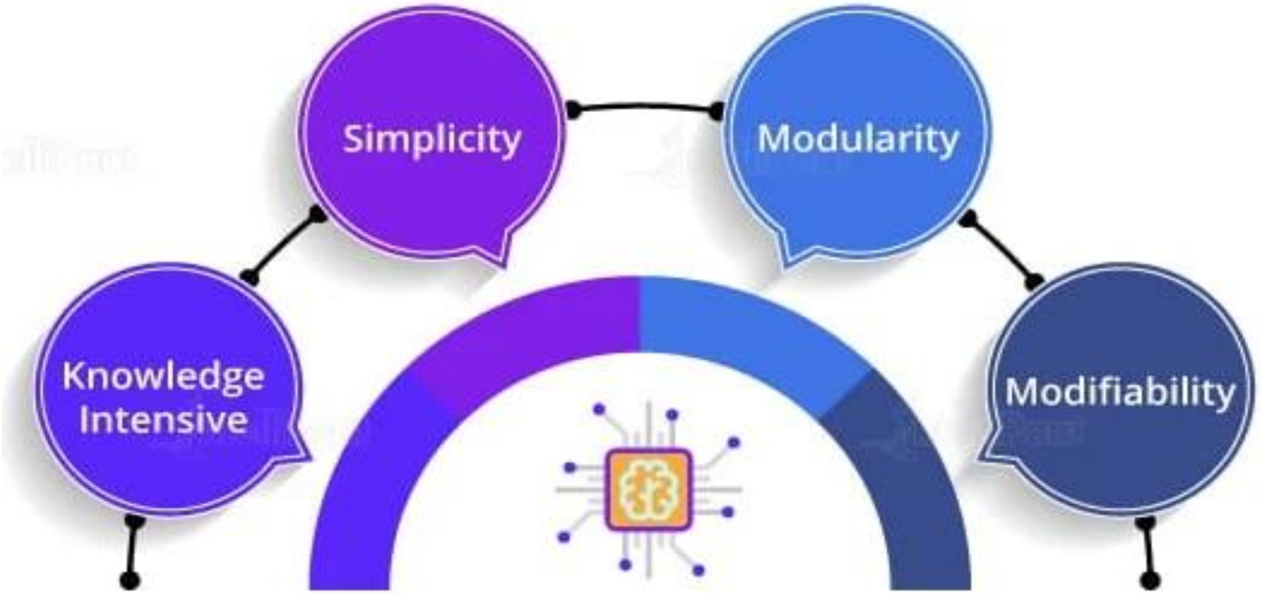
Problem Characteristics

1. Is the problem Decomposable?
2. Can solution steps to be ignored or undone?
3. Is the problem’s universe predictable?
4. Is the good solution is absolute or relative?
5. Is the solution a state or a path?
6. What is the role of knowledge?
7. Does the task require interaction with a person?

Production System Characteristics

There are mainly four characteristics of the production system in AI that is

- simplicity
- modifiability
- modularity
- knowledge-intensive.



Simplicity

The production rule in AI is in the form of an 'IF-THEN' statement. Every rule in the production system has a unique structure. It helps represent knowledge and reasoning in the simplest way possible to solve real-world problems. Also, it helps improve the readability and understanding of the production rules.

Modularity

The modularity of a production rule helps in its incremental improvement as the production rule can be in discrete parts. The production rule is made from a collection of information and facts that may not have dependencies unless there is a rule connecting them together. The addition or deletion of single information will not have a major effect on the output. Modularity helps enhance the performance of the production system by adjusting the parameters of the rules.

Modifiability

The feature of modifiability helps alter the rules as per requirements. Initially, the skeletal form of the production system is created. We then gather the requirements and make changes in the raw structure of the production system. This helps in the iterative improvement of the production system.

Knowledge-intensive

Production systems contain knowledge in the form of a human spoken language, i.e., English. It is not built using any programming languages. The knowledge is represented in plain English sentences. Production rules help make productive conclusions from these sentences.

Thank You