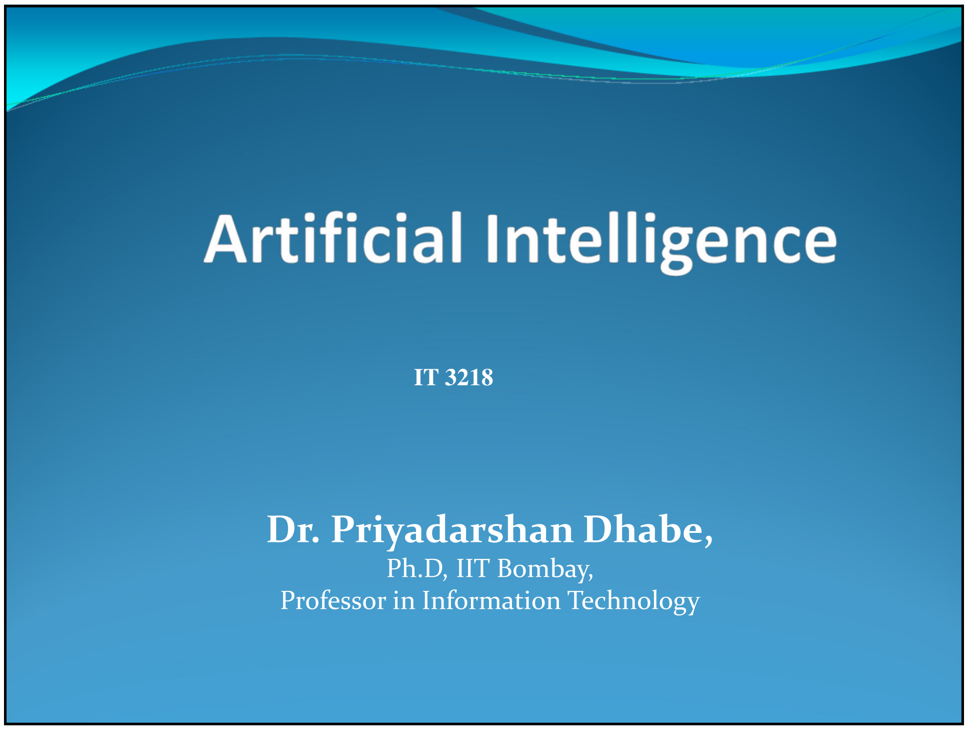
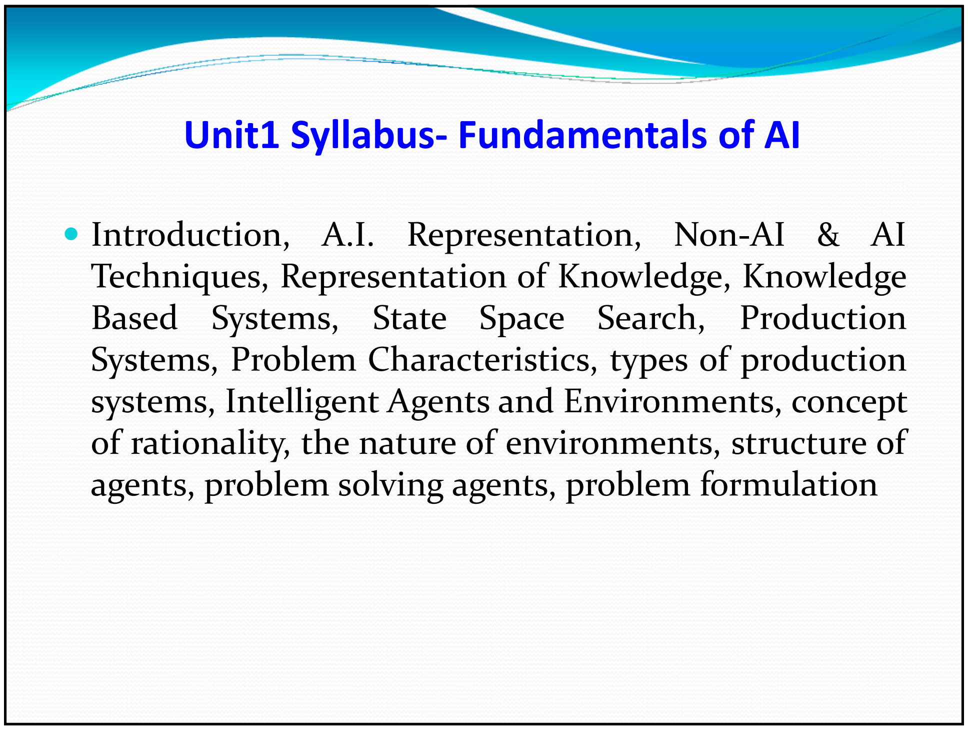


Artificial Intelligence

IT 3218

Dr. Priyadarshan Dhabe,
Ph.D, IIT Bombay,
Professor in Information Technology



Unit1 Syllabus- Fundamentals of AI

- Introduction, A.I. Representation, Non-AI & AI Techniques, Representation of Knowledge, Knowledge Based Systems, State Space Search, Production Systems, Problem Characteristics, types of production systems, Intelligent Agents and Environments, concept of rationality, the nature of environments, structure of agents, problem solving agents, problem formulation

What is Intelligence?

Word intelligence is derived from Latin verb "intelligere" (ability to think).

Definition:- Intelligence is ability to acquire and apply the knowledge, in an autonomous fashion.

- every leaving thing is intelligent
- Magnitude of intelligence may differ from one animal to another
- Measured using I.Q= Psychological age/physical age
- It is tightly relevant to LEARNING and KNOWLEDGE

What is Artificial Intelligence (AI)?

AI is study of how to make machines intelligent like humans?
-Machine must think, act, behave, learn and respond to the change in its environment like a human.
-Making machines **adaptive** to their environment

The Term "AI" and father of AI

- **Alan Turing, Minsky, Newell and Simmon- also contributed to AI**
- **John McCarthy –father of modern AI**
- (American Computer Scientist, 1927-2011) coined this term called "AI" (roughly around 1960)
- Developed programming language **LISP**
- Contributed in design of **ALGOL**, time sharing systems

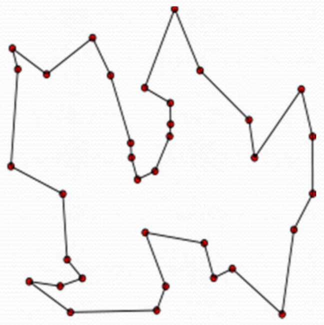
https://en.wikipedia.org/wiki/John_McCarthy_(computer_scientist)

AI is collection of diverse theories

- Learning:- Machine learning, Neural networks, deep learning, reinforcement learning
- Dealing with knowledge based systems
  - Fuzzy logic, Propositional logic, Predicate logic
- Sensing an environment
  - Use of sensors to detect change in environment
- Searching- (informed and blind search)
  - Many problems are of type state space searches e.g chess
- Natural language understanding/processing (alexa, google assistant)

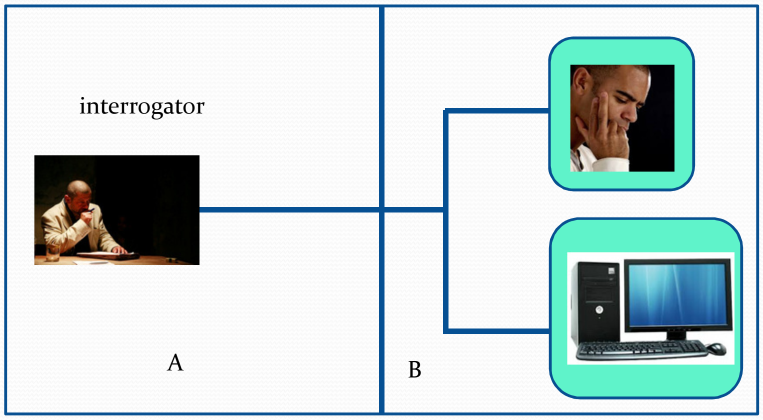
AI and NP-hard problems

- AI provides the only way to solve NP-hard problems in sub-optimal manner and in finite time.
  - Travelling salesman problem 47! Tours



Criterion of success-Turing Test

-one possible form of Turing test
- Do it using ChatGPT (Generative pre-training Transformer)



Gary Kasparov VSS IBMs Deep Blue

Deep Blue defeated Kasparov in 1997 , some times he felt that his opponent is a human and not the machine

https://en.wikipedia.org/wiki/Deep_Blue_versus_Garry_Kasparov

Definitions of AI in 4 categories

Ref:- "AI- a modern approach", Stuart Russell and Peter V Norvig (*Director Research, at Google*)

Logically correct thinking is rationality

AI is study of how to make

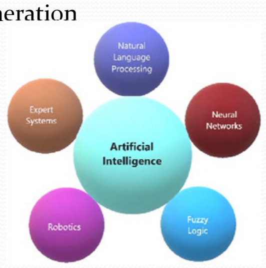
- Systems that think like humans
- Systems that act like humans
- Systems that think rationally
- Systems that act rationally

Peter V Norvig

Task domains of AI

Broad areas where AI can be applied

- **Mundane tasks (every day , ordinary tasks)**
  - Perception-Vision, speech
  - Natural language- Understanding, translation, generation
  - Common sense reasoning
  - Robot control- while navigation
- **Formal Tasks**
  - Games-chess, tic-tac-toe
  - Mathematics-Geometry, logic, theorem proving
- **Expert Tasks**
  - Engineering-design, fault detection and diagnosis, manufacturing planning
  - Scientific and Financial analysis, Medical diagnosis

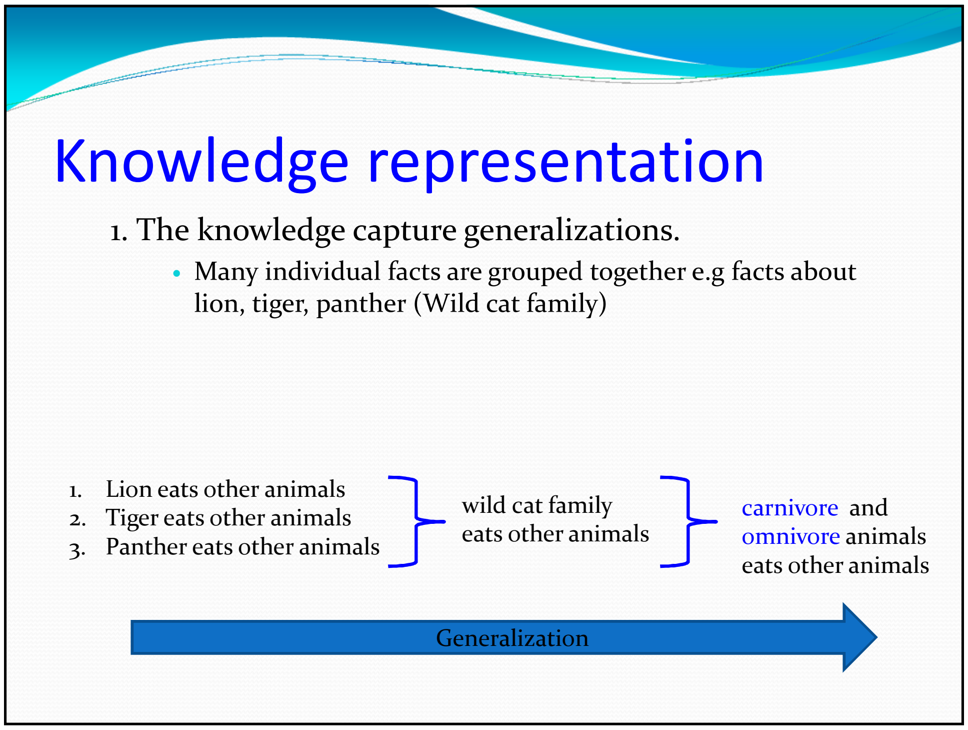


Intelligence requires knowledge

- **Undesirable properties of Knowledge**
  - It is voluminous
  - It is hard to characterize (identify, specify, indicate) accurately
  - It is constantly changing
  - It is organized in the way corresponding to its use (totally different than organization of data)

What is AI technique?

- AI technique is the method that **exploits knowledge** that should be **represented** in such a way that
  - The knowledge capture **generalizations.**
  - It can be **understood** by people who must provide it.
  - It can be easily **modified** to correct errors and to reflect changes in the world
  - It can be used in great many situations even if it is not **totally accurate** or complete
  - It must help to **narrow down** the range of possibilities.



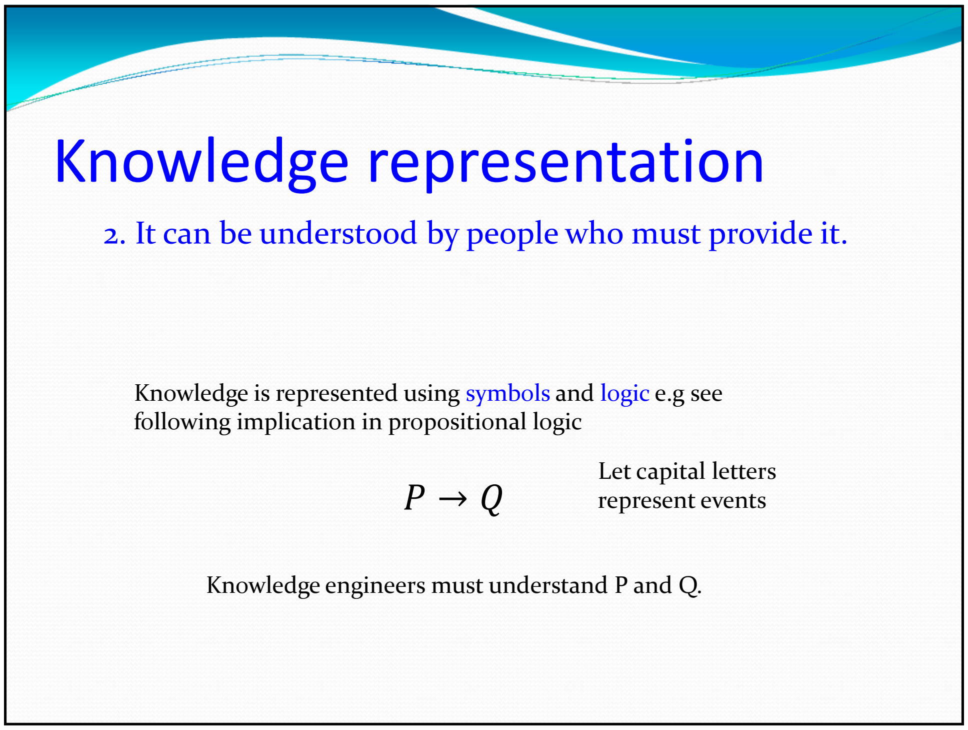
Knowledge representation

1. The knowledge capture generalizations.
   - Many individual facts are grouped together e.g facts about lion, tiger, panther (Wild cat family)

1. Lion eats other animals
2. Tiger eats other animals
3. Panther eats other animals

$\}$ wild cat family eats other animals $\}$ carnivore and omnivore animals eats other animals

Generalization



Knowledge representation

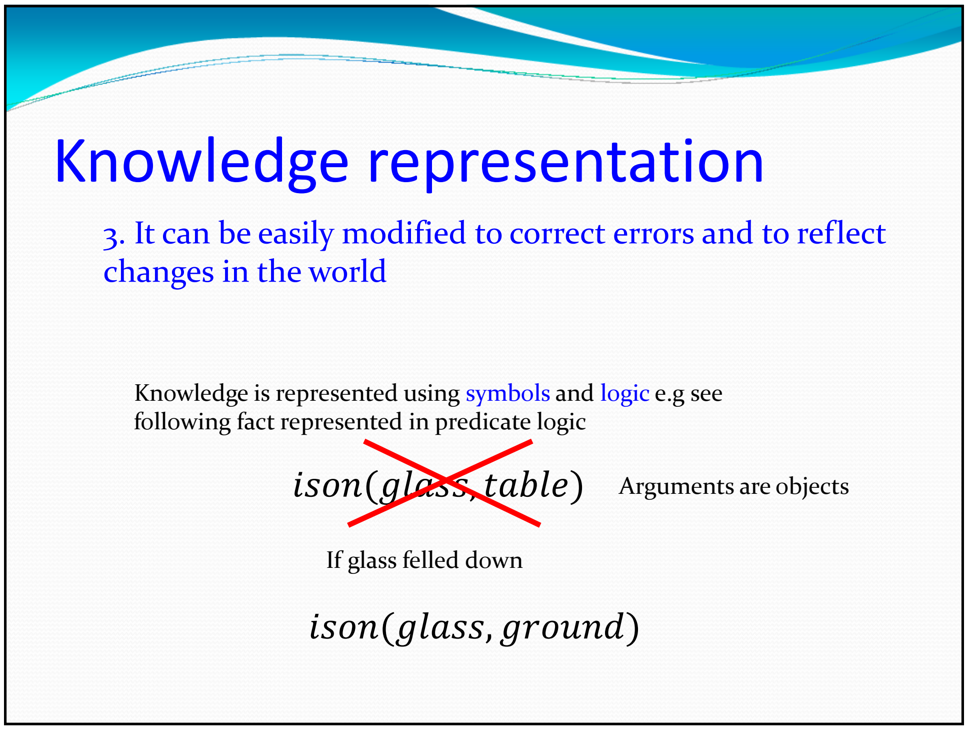
2. It can be understood by people who must provide it.

Knowledge is represented using symbols and logic e.g see following implication in propositional logic

$$P \rightarrow Q$$

Let capital letters represent events

Knowledge engineers must understand P and Q.



Knowledge representation

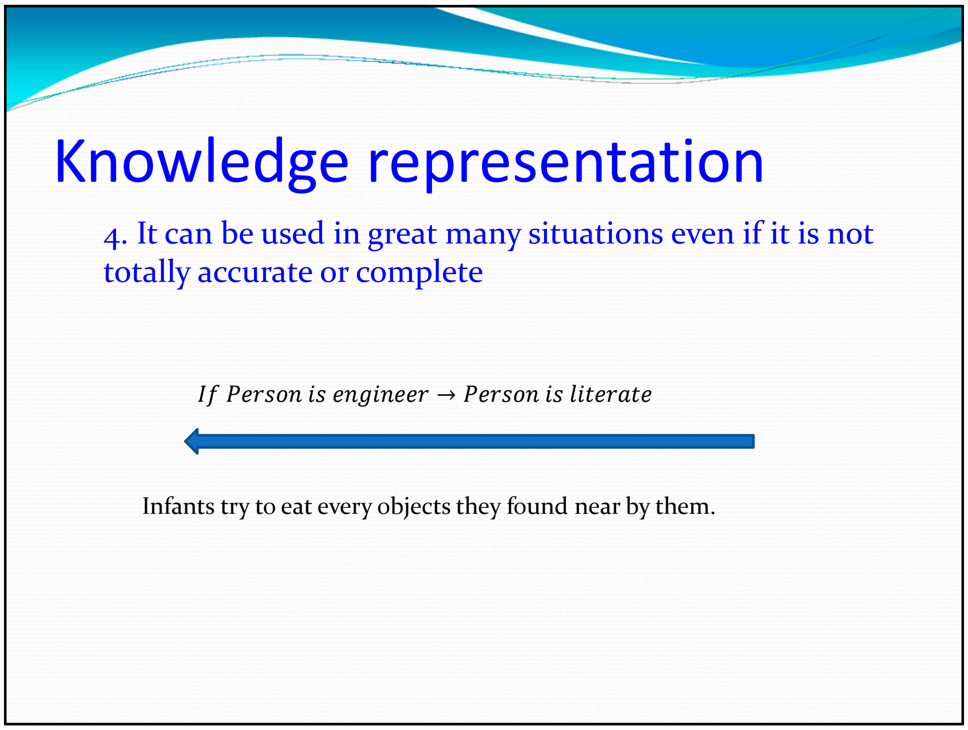
3. It can be easily modified to correct errors and to reflect changes in the world

Knowledge is represented using symbols and logic e.g see following fact represented in predicate logic

$\cancel{ison(glass, table)}$ Arguments are objects

If glass felled down

$ison(glass, ground)$



Knowledge representation

4. It can be used in great many situations even if it is not totally accurate or complete

$If\ Person\ is\ engineer \rightarrow Person\ is\ literate$

Infants try to eat every objects they found near by them.

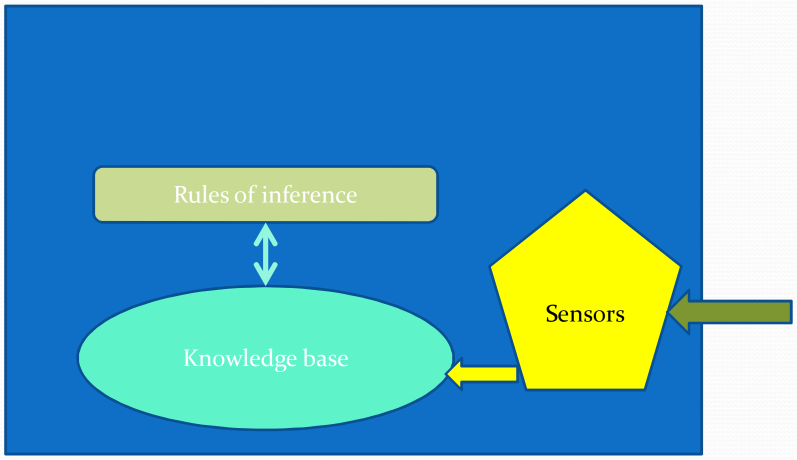
Knowledge representation

5. It must help to narrow down the range of possibilities.

BANK

Bank of the river

Knowledge based systems-definition



Playing Tic-Tac-Toe

-it is a pen-paper game & need 2 players

- It uses a 3 x 3 cell board
- Players P1 and P2 choose symbols among "X" and "O"
- A player wins if succeeded to put 3 symbols in any row, column or diagonal.

Decide

1. Representation of board position in computer memory
2. How to generate possible moves?
3. Computing a score associated with each possible move

Playing Tic-Tac-Toe

O	X	X
O		

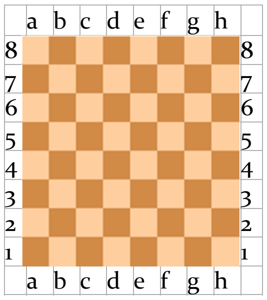
Assume the current position of the board as shown. Let the next turn belongs to a player with symbol "O". Find possible moves and suggest best one by measuring the scroes.

Problem as a "*state space search*"

- Many problems can be solved as searching through a space of collection of states.
- Playing chess can be considered as a problem of moving through
  - 10^120 (shannon number) possible board positions
  - with 35 as branching factor
  - average game length 80 moves.

https://en.wikipedia.org/wiki/Shannon_number

Playing Chess

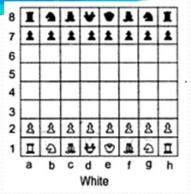


-It has an 8 x 8=64 square board. Played by 2 players
-Columns are called as "files" and are labeled from left to right as a to h.
-Rows are called as "ranks" and are number from 1 to 8 as shown.
-There are 6 types of pieces in chess each moves in a different manner

rank

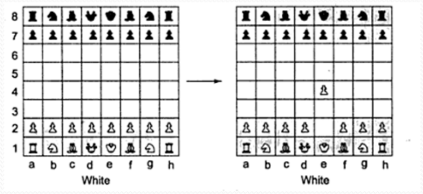
Files

How to play Chess?



- We need description of starting position.
- Description of (multiple) ending positions (win or draw)
- Description of legal moves providing a path from initial state to the goal state.
- A move can be defined using two parts: a left part describing current board position and a right side describing the changes to be made to the board position to reflect the move.

How to play Chess?



A legal chess move- a rule

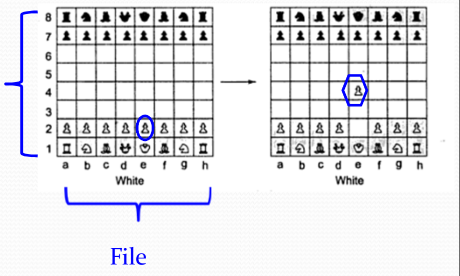
Playing chess is
–Explicitly writing and storing all the possible rules
- Taking input as current board position and finding the match with left side of the rule and apply that rule .

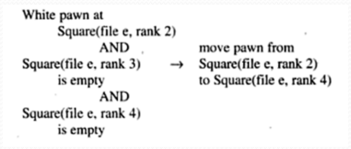
Drawbacks :-

1. No one can supply a complete set of rules or may need long time and can not be done without mistakes
2. No program can easily handle such rules (need excessive space)

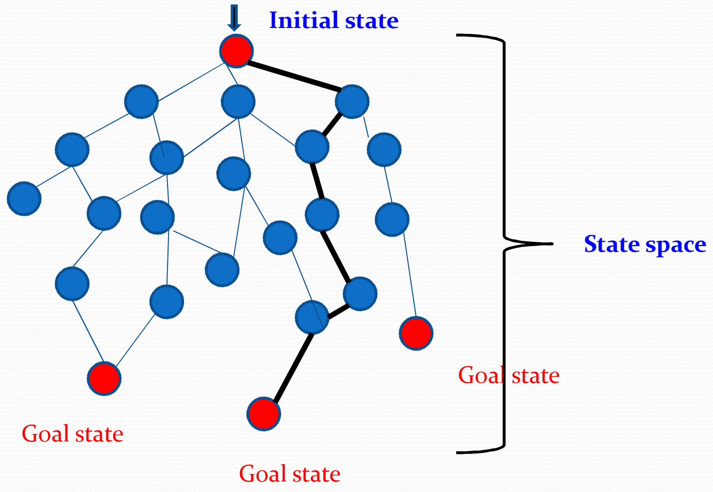
How to play Chess?

A more general way of describing rule




```
White pawn at
      Square(file e, rank 2)
             AND                     move pawn from
Square(file e, rank 3)      →    Square(file e, rank 2)
      is empty                       to Square(file e, rank 4)
             AND
Square(file e, rank 4)
      is empty
```

Playing chess as state space search

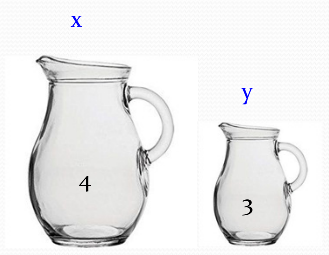


State space search

- Many real world problems can be represented as state space search and forms basis of AI methods.
- This representation is helpful for less structured problems.
- Solving a problem using such representation is finding one of the possible paths from initial state to goal state.

Water jug problem-as state space search

1 gallon=3.78541 lit



Problem:- You are given two jugs, a 4 gallon one and a 3 gallon one. Neither has 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 in 4 gallon jug?

Representation of current situation and the state space for this problem is described using ordered pair (x,y)

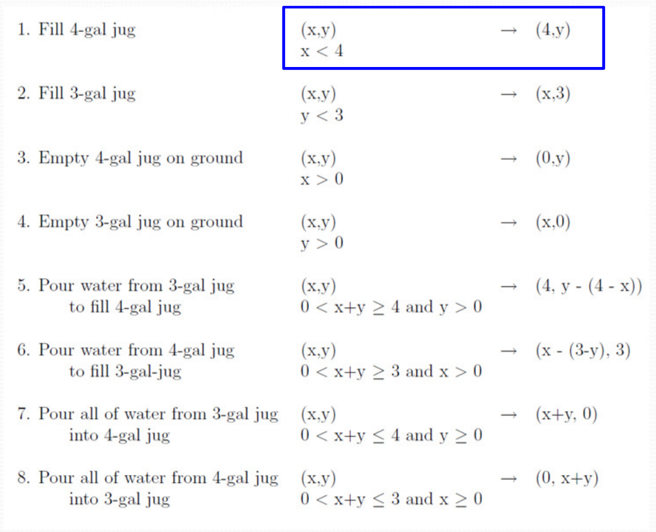
x- number of gallons of water in a 4 gallon jug x=0,1,2,3,4

y- number of gallons of water in a 3 gallon jug y=0,1,2,3

Start state is (0,0) and end state is (2,n) for n<=3

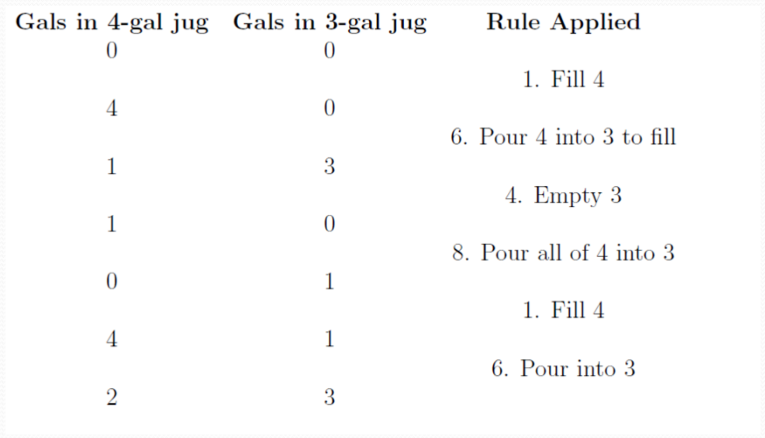
Production rules/operators for Water jug problem

Rules/operator



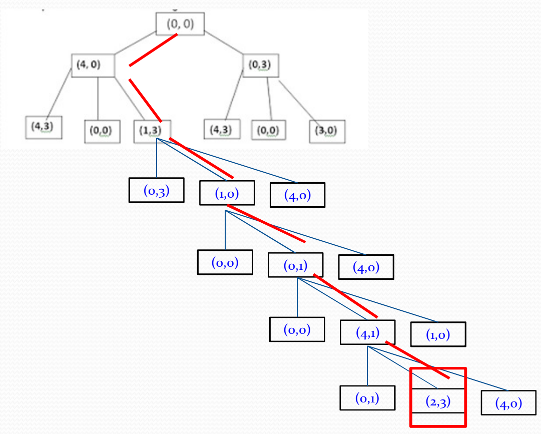
1. Fill 4-gal jug	(x,y) $x < 4$	$\rightarrow$	(4,y)
2. Fill 3-gal jug	(x,y) $y < 3$	$\rightarrow$	(x,3)
3. Empty 4-gal jug on ground	(x,y) $x > 0$	$\rightarrow$	(0,y)
4. Empty 3-gal jug on ground	(x,y) $y > 0$	$\rightarrow$	(x,0)
5. Pour water from 3-gal jug to fill 4-gal jug	(x,y) $0 < x+y \geq 4$ and $y > 0$	$\rightarrow$	(4, y - (4 - x))
6. Pour water from 4-gal jug to fill 3-gal-jug	(x,y) $0 < x+y \geq 3$ and $x > 0$	$\rightarrow$	(x - (3-y), 3)
7. Pour all of water from 3-gal jug into 4-gal jug	(x,y) $0 < x+y \leq 4$ and $y \geq 0$	$\rightarrow$	(x+y, 0)
8. Pour all of water from 4-gal jug into 3-gal jug	(x,y) $0 < x+y \leq 3$ and $x \geq 0$	$\rightarrow$	(0, x+y)

Water jug problem: solution



Gals in 4-gal jug	Gals in 3-gal jug	Rule Applied
0	0	
		1. Fill 4
4	0	
		6. Pour 4 into 3 to fill
1	3	
		4. Empty 3
1	0	
		8. Pour all of 4 into 3
0	1	
		1. Fill 4
4	1	
		6. Pour into 3
2	3	

Water jug : solution using graph



Production (rule) system

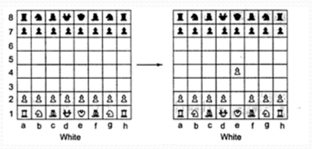
- **Production systems** provides structure to AI programs to facilitate describing and performing search.
- Wikipedia: "*A **production system** (or **production rule system**) is a computer program typically used to provide some form of artificial intelligence, which consists primarily of a set of rules about behavior. These rules, termed **productions**, are a basic representation found useful in automated planning, expert systems and action selection. A production system provides the mechanism necessary to execute productions in order to achieve some goal for the system*".

Production system-Components

- **Production systems consists of**
  - A set of rules (production rules)
  - One or more knowledge/Database (working memory)
  - Control strategy (Search technique)
  - Rule applier

Components of Production system

- 1. A set of rules : each consisting of a left hand side (a pattern) that determines the applicability of the rule and a right hand side that describes the operation to be performed if the rule is applied (forward rules)



Antecedent Consequence

Condition Action Rule

Components of Production system

- 2: **One or more knowledge/ database** : that contain info appropriate for a given task. Some part of the database may be permanent and info stored in them is structured in appropriate way. (e.g pieces and their moves in chess)
- 3. **A control strategy**:- that specify the order in which the rules will be compared and a way of resolving conflicts that arises when several rules matches at once. (search techniques) (e.g multiple moves in chess)
- 4. **A rule applier:-** a component that actually apply the rule selected by control strategy to change the current position.

Control strategies- search Techniques

- Which rule to apply? What to do if more than 1 rule is applicable?- will have considerable impact on how quickly the problem is solved?
- Properties of Control strategies:-

Properties of Control strategies

- **1. A good control strategy must cause a movement-** if we decide to use control strategy for water jug problem to "always start with top of the list of rules and choosing the first applicable rule", we never solve the problem.
- **2.It must be systematic:-** in water jug problem using cycles we may arrive at same state and will waste considerable time, if "*systematic*" strategy is not used. A systematic strategy means it causes global motion and local motion as well, that produces goal state.

Features of production systems

- **Simplicity-** It codes knowledge by using simple IF-THEN format, which is simple knowledge representation.
- **Modularity:-** Rules are added or deleted without any harmful effect.
- **Modifiability:-** It allows system to modify rules. Hence, writing few skeleton rules and then fine tuning for accuracy for a specific application is possible.
- **Knowledge intensive:-** KB of production system contains pure knowledge and not the control.

Disadvantages of production systems

- **Opacity**- (slowlyness) This is due to less prioritization of rules
- **Inefficiency**- due to large number of rules
- **Absence of learning-** how a problem is solved earlier is not stored
- **Conflict resolution-** many rules can be applied at a condition

Control strategies

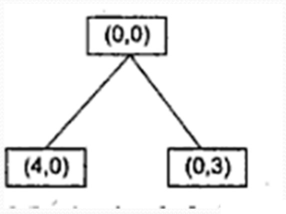
- Breadth first search
- Depth first search

Algorithm-Breadth first search

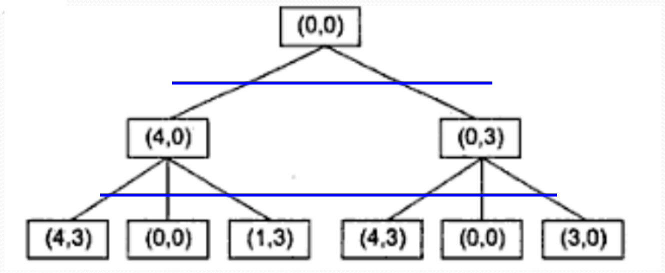
1. Create a variable called NODE-LIST and set it to the initial node
2. Until a goal state is found or NODE-LIST is empty do
   a. Remove the first element from NODE-LIST and call it E. If NODE-LIST was empty, quit.
   b. For each way that each rule can match the state described in E do
      1. Apply rule to generate a new state
      2. If the new state is a goal state, quit and return this state
      3. Otherwise, add the new state to the end of NODE-LIST

Breadth first search to water jug problem

One level of Breadth first search



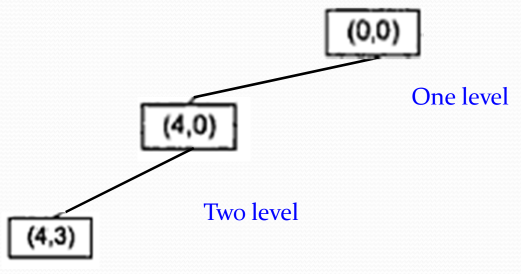
Two levels of Breadth first search



Depth first search

1. If initial state is a goal state, quit and return success.
2. Otherwise, do the following until success or failure is signaled
    1. Generate a successor, E, of the initial state. If there are no more successors, signal failure. (blind alley)
    2. Call depth first search with E as the initial state.
    3. If success is returned signal success. Otherwise continue in this loop.

Depth first search to water jug problem



Depth first search

- **Advantages:-**
- Requires less space, since only nodes on the current path are stored, where in BFS all the tree that has been generated so far need to be stored.
- DFS may find the solution without examining much of the search space. On the other hand in BFS all the tree to level n need to be examined

Breadth First Search

- **Advantages:-**
- It will not be trapped in exploring a blind alley.
- If there is a solution BFS is guaranteed to find it. Further more if multiple solutions exists, BFS find the minimal (**optimal**) solution. (i.e solution with minimal cost or path length)

Types of Search Techniques

- **Blind search (un-informed search):-**

  - No extra information is available about the states like **goodness** of a node that helps in guiding the search.

  - DFS and BFS are blind search techniques.

- **Informed search (heuristic search):-**

- Heuristic means to discover
- Some (little) information is available to guide the search
- Like cost of reaching from current state to successor or measure of "goodness" (rough estimate) of a successor.
- A* and AO* algorithms are heuristic search techniques

Heuristic Technique or Heuristics

- Term **heuristic** is derived from Greek word "**Eureka**", means to self-discover.
- *Wikipedia* defines (https://en.wikipedia.org/wiki/Heuristic)
- "It refers to experience-based techniques for problem-solving, learning, and discovery, which is not guaranteed to be optimal, perfect, or completely rational. But can solve problems quickly .
- Where finding an optimal solution is impossible or impractical, heuristic methods can be used to speed up the process of finding a **satisfactory solution**. Heuristics can be mental shortcuts that ease the cognitive load of making decision.

Heuristics Example

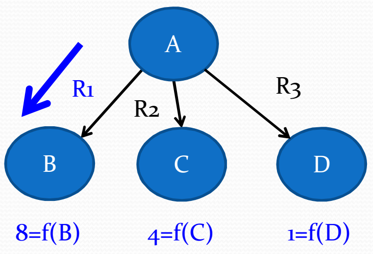
- A lady from England, *Muriel Bristol* (1888-1950) could predict with high success rate, that either a milk or tea is added, first, in the cup of tea.
- She was famous subject of "*Lady Tasting Tea*" experiment.
- She was also able to detect the taste difference due to different orders of pouring milk and tea.

https://en.wikipedia.org/wiki/Muriel_Bristol

Heuristic functions

- Heuristic (informed) search techniques are using "heuristic functions", that takes input as a node (state) and computes and returns a measure of "desirability", usually a number.

Example- "nearest neighbor heuristic" for travelling salesman problem. Efforts are reduced from N! to N^2



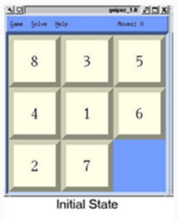
Heuristic functions properties

Properties:-

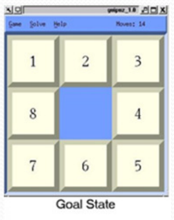
1. They provide an **approximate** values
2. They are **simple** (no complex logic and execute in small time)
3. Need to be designed by keeping an **eye on goal state**

8-Puzzle game

- Single player game, with 3 x 3 board, 8- tiles are numbered from 1 to 8 and there is one gap. With multiple initial state a single goal state.



Initial State



Goal State

- The max. number of states possible can be the permutation of 9 values from 0,1,2,….8 (0-indicate gap)

Total states $= 9! = 9x8x7x6.....x1$

https://www.d.umn.edu/~jrichar4/8puz.html

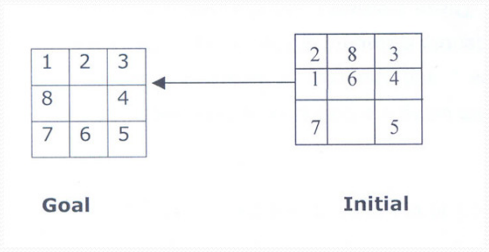
8-Puzzle game- some moves

$$\begin{bmatrix} & 1 & 3 \\ 4 & 2 & 5 \\ 7 & 8 & 6 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & & 3 \\ 4 & 2 & 5 \\ 7 & 8 & 6 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 2 & 3 \\ 4 & & 5 \\ 7 & 8 & 6 \end{bmatrix}$$

https://www.cs.princeton.edu/courses/archive/spr10/cos226/assignments/8puzzle.html

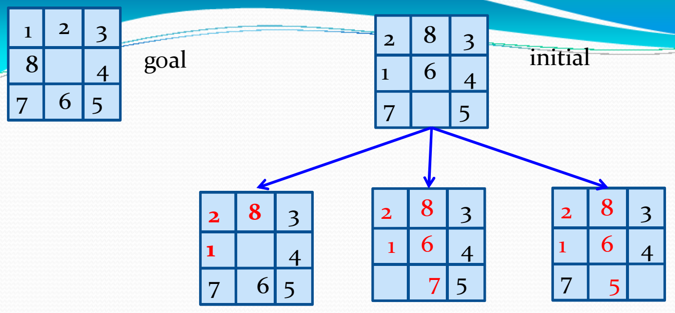
Heuristic function design for 8-puzzle

1. **h1(s)**=Number of misplaced tiles w.r.t goal state.



Generate all possible states that are legal under this game and apply heuristic function h1 for all of them

2. **h2(s)**=Euclidean distance between current state and goal state.



h1(A)=3 h1(B)=6 h1(C)=6

h1(s)=Number of misplaced tiles w.r.t goal state.

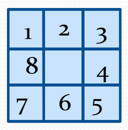
h2(A)= **9.27** h2(B)= 13.11 h2(c)= 12.17

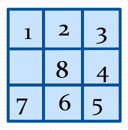
h2(s)=Euclidean distance between current state and goal state

$$x=(x_1,x_2,\dots,x_n)\quad \text{and } y=(y_1,y_2,\dots,y_n)$$

then

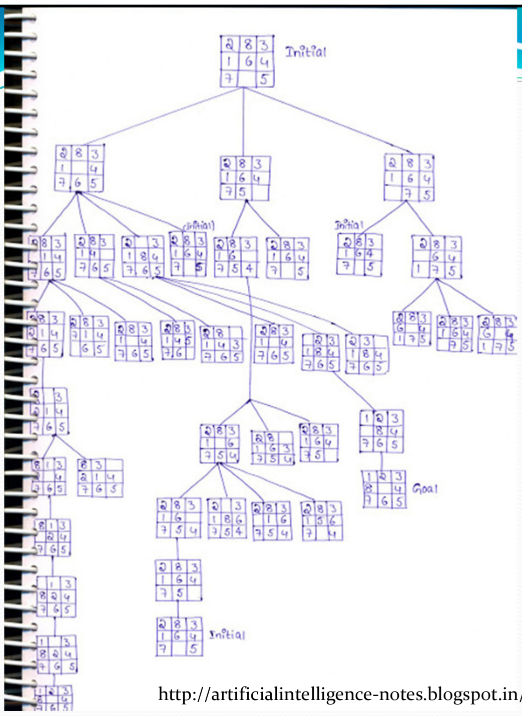
$$d=\sqrt{(x_1-y_1)^2+(x_2-y_2)^2+\dots(x_n-y_n)^2}$$





$$d=SQRT((1-1)^2+(2-2)^2+(3-3)^2+(8-0)^2+(0-8)^2$$
$$+(4-4)^2+(7-7)^2+(6-6)^2+(5-5)^2)$$

$$d=SQRT(64+64)=SQRT(128)=11.31$$



8-puzzle game tree

http://artificialintelligence-notes.blogspot.in/2010/07/8-puzzle-problem.html

Characteristics of the problems encountered in AI

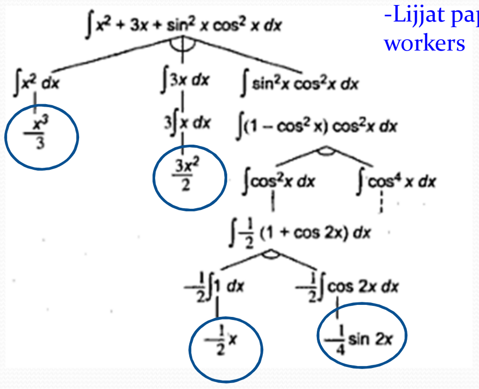
- For deciding a more appropriate method for solving a problem, we need to know its characteristics.

1. Is the problem decomposable into set of (nearly) independent smaller or easier sub-problems?

- If **yes** then we can solve very large problems easily. Solve sub-problems and integrate their solutions into a global solution.

Characteristics of the problems encountered in AI

-Level of decomposition?
-Lijjat papad example-43k workers


$\int x^2 + 3x + \sin^2 x \cos^2 x \, dx$
$\int x^2 \, dx$
$\frac{x^3}{3}$
$\int 3x \, dx$
$3\int x \, dx$
$\frac{3x^2}{2}$
$\int \sin^2 x \cos^2 x \, dx$
$\int (1 - \cos^2 x) \cos^2 x \, dx$
$\int \cos^2 x \, dx$
$\int \cos^4 x \, dx$
$\int \frac{1}{2}(1 + \cos 2x) \, dx$
$\frac{1}{2}\int 1 \, dx$
$\frac{1}{2}\int \cos 2x \, dx$
$\frac{1}{2}x$
$\frac{1}{4}\sin 2x$

A decomposable problem

Characteristics of the problems encountered in AI

2. Can solution steps be ignored or at least undone if they prove unwise?

A. Theorem proving- we may start with proving a lemma assuming it will be helpful and then we realize that it is of no use. (ignore solution step)

B. 8-puzzle:- is a square tray in which 8-square tiles with numbers 1 to 8 are placed and there is a gap at ninth square. (solution steps undone by backtracking)

C. Chess:-

solution step are irrecoverable

Start

2	8	3
1	6	4
7		5

Goal

1	2	3
8		4
7	6	5

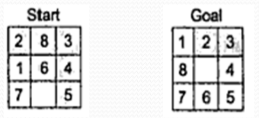
Characteristics of the problems encountered in AI and Control strategy

- **Ignorable problems:-** need simple control structures that never backtracks and easy to implement.
- **Recoverable Problems:-** Need more complicated control strategy that makes mistakes and thus, backtracking is necessary.
- **Irrecoverable problems:-** need control strategy that put more efforts on taking decision since it is final.

Characteristics of the problems encountered in AI

3. Is the universe predictable?

- In *8 puzzle*, we know what will happen after a move? And thus planning process can be used with backtracking. (certain outcome)



Start

2	8	3
1	6	4
7		5

Goal

1	2	3
8		4
7	6	5

- But in *playing cards* (uncertain outcome).

Characteristics of the problems encountered in AI

4. Is a Good solution Absolute or Relative?

- In traveling sales man problem, there can be multiple paths from source to destination. But, if we want the optimal path, then problem is Best–path problem (absolute solution/independent solution) otherwise, it will be Any-path problem (relative solution).

Characteristics of the problems encountered in AI

5. Is the solution a State or Path?

- *Water jug* problem is finding a path problem to state (2,x). How can we obtain this state is necessary?
- Playing *tic-tac-toe* is a finding a state problem.

6. What is the role of knowledge?

- Problems for which a lot of knowledge is important only to constrain the search for a solution. (**Chess**)
- Problems for which a lot of knowledge is required even to be able to recognize a solution. (**Rummy**)

Characteristics of the problems encountered in AI

7. Does the task require interaction with a Person?

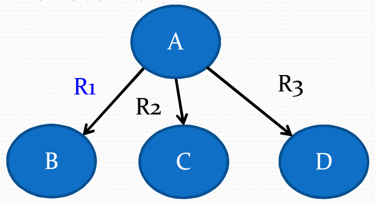
- Leads to two kind of problems
- **Solitary Problems:-** in which a computer is given problem description and produces an answer *with no intermediate communication* and with no demand of reasoning.
- **Conversational Problems:-** *Need intermediate communication* to assist computer or to provide info to the user or both.

Types (characteristics) of Production systems

- Based upon characteristics, **Production Systems** are of following types.

1. Monotonic (predictable/unchaning) production system:-

is one in which application of a rule **never prevents** the later application of **another rule** that could also have been applied at the time the first rule was selected. Otherwise, it is called **non-monotonic.**


$$\begin{tikzcd} & A \arrow[dl, "R1"'] \arrow[d, "R2"] \arrow[dr, "R3"] & \\ B & C & D \end{tikzcd}$$

Types (characteristics) of Production systems

2. Partially commutative (exchange) production system:- is one in which, if the application of particular sequence of rules transforms the state x into state y, then any permutation of those rules, allowable, also transforms state x into state y. (theorem proving)

3. Commutative production system:- is both monotonic and partially commutative.

Types (characteristics) of Production systems and their uses

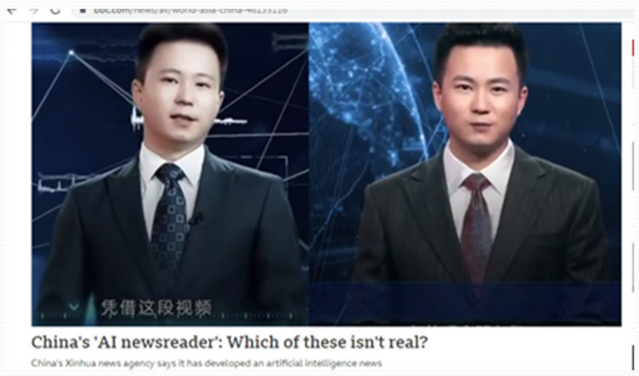
- **Partially commutative, monotonic production systems** are used for- ignorable problems like theorem proving
- **Partially commutative, non-monotonic production** systems are used for- recoverable problems like Robot navigation, 8-puzzle
- **Not partially commutative:-** are useful for solving irrecoverable problems like playing chess.

AI and Intelligent Agents

- One of the prime goal of AI is to develop intelligent agents (also called as *AI assistants*)
- Example- **Alexa** deveoped by Amazon
- (small version 5k INR)
  - Interact with voice
  - Music playback
  - Making To-Do list
  - Setting alarms
  - Playing audio books
  - Can provide info on weather, sports, news and etc
- In **2018** around **10k** amazon employees were working on Alexa

https://en.wikipedia.org/wiki/Amazon_Alexa

AI 3D news readers/Anchors



China's 'AI newsreader': Which of these isn't real?

China's Xinhua news agency says it has developed an artificial intelligence news

https://www.bbc.com/news/av/world-asia-china-46135116

What is an agent?

- Oxford dictionary "Agent is –An entity (person/program/machine (H/W +S/W)) who acts on behalf of another person or group"

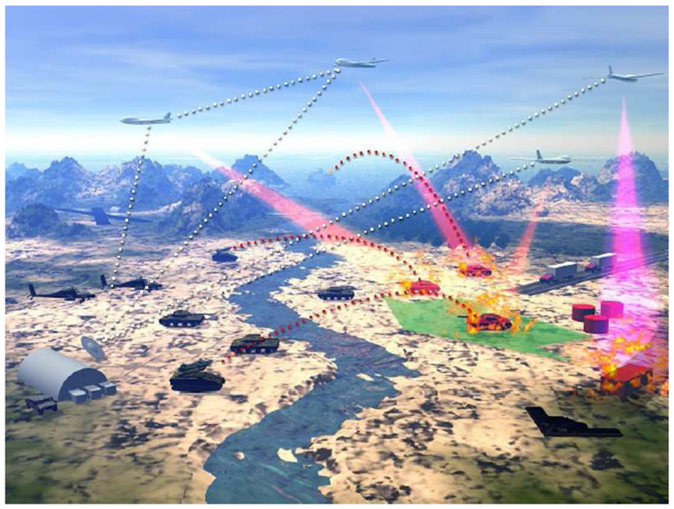
What is an intelligent Agent?

- **General definition**
- **An agent can be anything that can Perceive its environment through Sensors and Acting upon that environment through actuators/effectors.**
- **Behaves intelligently.**
- **Agent do the task and report to some one or communicate to others.**

What can be the Agent?

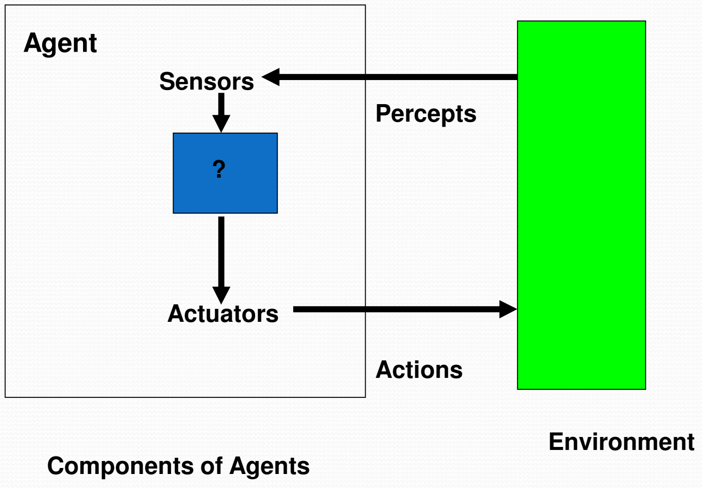
- A sensor with the capability to communicate can be an agent in the sensor network.
- A Satellite, A vehicle can be agent.
- A human can also be an Agent.
- A software program can be an agent

A Multiple Agent system.

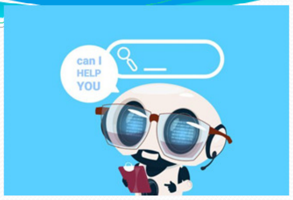


- A cooperative war-fare system

Basic components of an Agent



Agent Terms



- **Percept Sequence**:-
  - is “**history of everything**” the agent has learned.
  - Actions of agents are depend on percept sequence.
- **Agent Function**:-
  - Describes “**behavior**” of an agent
  - Internally it is implemented by a “**agent program**”.
- **External Characteristics**:-
  - of an agent is Tabulation of “Percept sequence” and “Action”.
- **Agent- architecture + Program**

https://www.prdaily.com/try-these-10-ai-assistants-to-make-your-productivity-soar/

What is a Rational Agent?

- Agents that work "Rationally". (logically)
- Rationality :- can be defined on the basis of Performance measure, prior knowledge, Actions and percept sequence.
- Definition:- For each possible percept Sequence a rational agent should select an action that can maximize its performance.

Doing right things is rationality

Properties of Rational (logical) agents

1. Omniscience:-
   - is property that allow agents to know the actual consequences of its actions and can act accordingly.
2. Learning:-
   - Must have capability to learn from environment.
   - Actions depends on knowledge,& knowledge on Learning.
   - It keeps the Knowledge up to date.
3. Autonomy:-
   - Must be autonomous for Learning and Decisions.
   - When agents have less or no knowledge they act randomly.
   - They can assume initial knowledge.

Task Environment of an Agent

- It is collection of Performance Measure, Environment, Actuators and Sensors.
- It is called PEAS description of agent.
- For design of an agent we must know PEAS description.

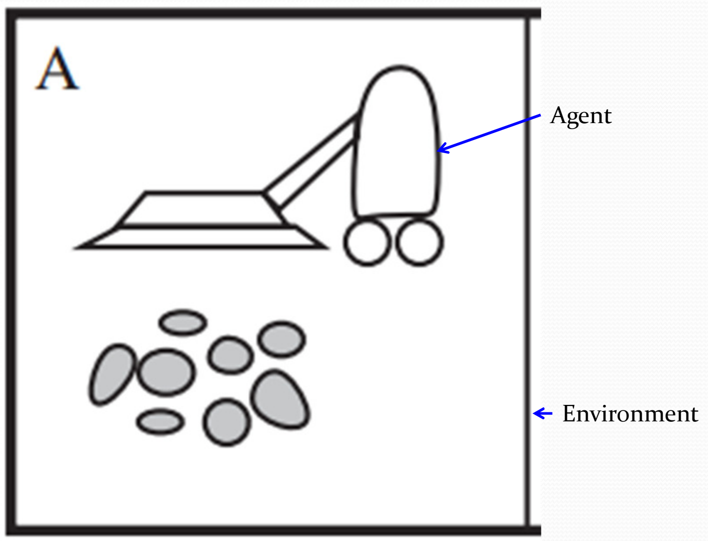
PEAS Description of Agent "Taxi Driver"

Tesla Auto Pilot

Agent Type	Performance Measure	Environment	Actions	Sensors
Taxi Driver	Safety, Speed, Comfort	Roads, Other traffic, Signs, dividers	Breaking, Acceleration, steering, clutching and etc	Cameras, Speedometer, GPS, Accelerometer

https://en.wikipedia.org/wiki/Tesla_Autopilot

AI Task Environment- Vacuum Cleaner Agent



Properties of AI Task Environment

1. Fully (8-puzzle) VSS Partially observable (taxi driving)
2. Deterministic VSS Stochastic
   - Fully observable is deterministic partially observable is stochastic
3. Episodic VSS Sequential. (chess vss driving)
4. Static Vss Dynamic
   - knowledge don't change (speech processing) vss change in knowledge (vision system)
5. Discrete VSS Continuous (chess vss self driving cars)
   - Playing chess is discrete and taxi driving is continuous
6. Single Agent Vss Multi Agent- Cooperate or compete

Agent Architectures

Architecture –
wikipedia- design and building study of anything

- Following are 4 basic types of Intelligent agents architectures
  1. Simple Reflex Agent.
  2. Model based Reflex Agent.
  3. Goal based Agents
  4. Utility based agents

$\downarrow$ Complexity, Intelligence

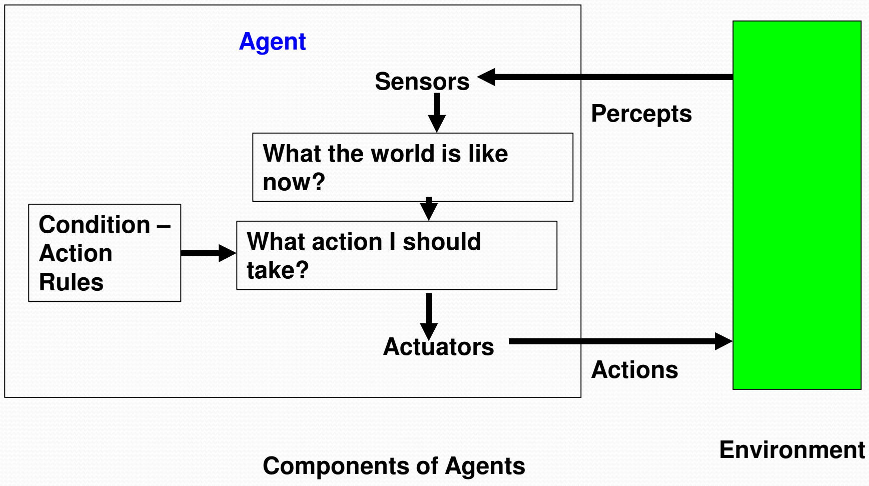
Simple Reflex agent

- Reflex- means reaction
- Takes actions based on current percept and ignore rest percept history.
- It consists of condition action rule like

  if (obstacle ahead)→ Apply the break

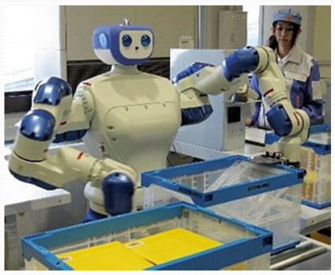
```
Function SimpleReflexAgent (percept) return Action
Static: rules (condition-action)
State← Interprete-Input (percept)
Rule←Rule-Match(State, Rules)
Action←Rule-Action (Rule)
```

Diagram of Simple Reflex Agent



Example of Simple reflex Agent

- Mail sorting Robot.
- Environment:- Conveyor belt of letters
- Rules
  - *If (PIN=411037)Then put letter in Pune Bin*

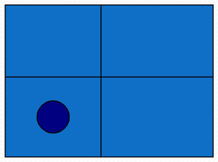


Properties of Simple Reflex Agent

1. They can caught in an infinite loop if environment is partially observable

- Simple reflex Vaccum Cleaner Agent-
  - -if (tile is clean)→move to left
  - - Escape from infinite loop is possible by taking a random action
  - - Thus randomized SRA can outperform than deterministic SRA

2. Suitable for Fixed environment



Model Based Agents

What is a model-?- Wikipedia- In general, a **model** is a representation of a person or a system which provides some information about it.

- **Representation** can be in
  - Mathematical form- mathematical model
  - Set of concepts
  - Physical form

Model of a Road for driving cars

Model Based Agents

$$bill = 13.5x + 18$$

Mathematical model of bill payment system

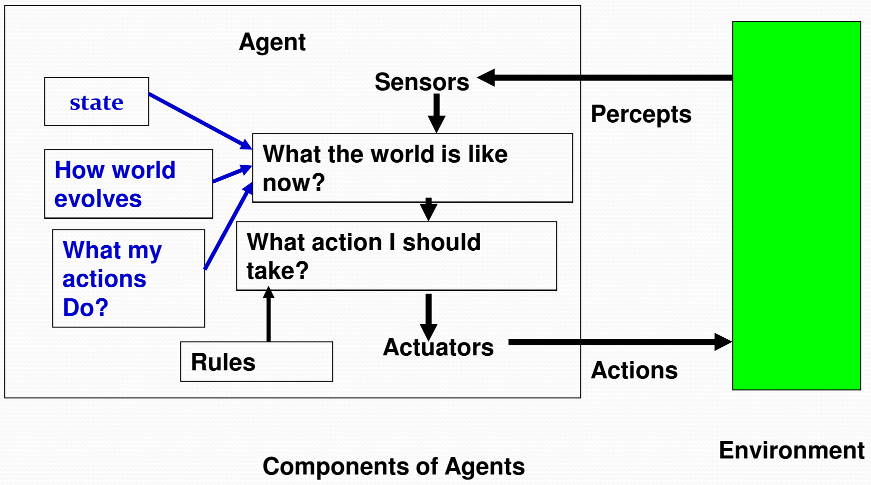
Model Based Agents

- An effective way of to handle partial observability is to maintain an internal state that depend on percept history.
- This internal state is called a model.
- To model the environment we need
  - knowledge of world
  - Effect of action on this world

Function of Model Based Agents

```
Function  Model-Based-ReflexAgent (percept) return Action
Static: state (description of current world), rules, Actions
State ← Update-State (state recognized in past, Past action,
          Current percept)
Rule ← Rule-Match(State, Rules)
Action ← Rule-Action (Rule)
Return action
```

Model Based Agent Architecture



Components of Agents

Example of Model based Vacuum Cleaner

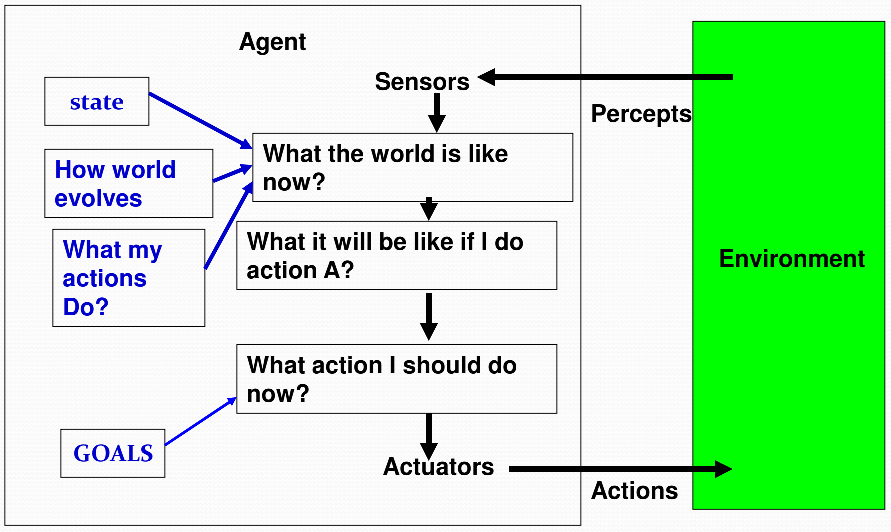
- Environment:- Room (20 X 20 tiles)
- Action :- based on model of the room

Goal Based Agents

Goal – aim or desired results

- Condition-Action rules and only model is <u>not sufficient</u> to decide properly what to do? Unless GOAL of the agent is not clear to it.
- Labor – Mason – Architect- Example
- Goal based agents takes actions for <u>their own GOALS</u> and <u>not for only changes in the environment</u>.
- They model the environment according to their GOAL.

Goal based Agent architecture


Agent
Sensors
Percepts
state
How world evolves
What my actions Do?
What the world is like now?
What it will be like if I do action A?
What action I should do now?
GOALS
Actuators
Actions
Environment

Goal based Agent-features

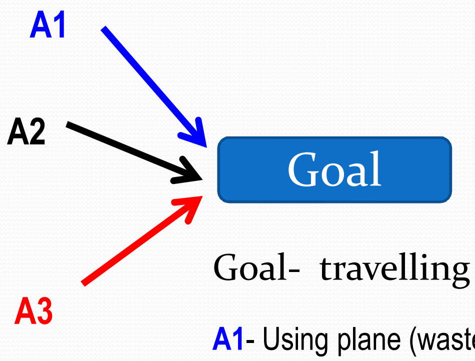
- **Less efficient-** Since they think more about which action to take
- **More flexible-** since their goals can be changed

Utility Based agents

- Utility- The quality of being useful
- Achieving Goals are not enough for high-quality behavior .
- _How we achieved Goal? is important.
  (less time/effort, more reliable, more safer)
- Sometimes goals are contradicting to Utility.
  - E.g of a researcher innovating Locks.

Utility Based agents

A **goal** can be achieved by **multiple actions** but some actions are **more useful** than others



Goal- travelling from VIT to IITB

A1- Using plane (wasteful of time and money)
A2- Using train (more useful)
A3- Using Bus (traffic jam is common)

Utility Based agents

- Uses "Utility Functions" that maps a state, or sequence of steps onto a real number, which describes a degree of perferability or degree of utility/usefulness.
- They are also having the goals.

Example- **Route recommendation system**

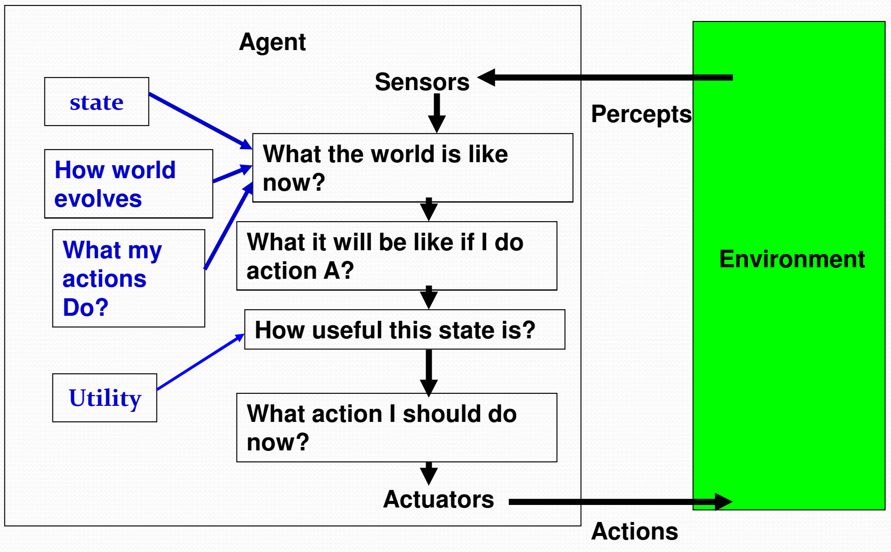
Suggests the "*best*" possible route currently available to keep their customers happy
-Best- short, comfort, safe, happy

Figure 1 Ensuring alternative routes (map data provided by Yahoo Maps)

Uses of Utility Functions

- **It helps in two way**
  - **1. Selecting Goals when there are conflicting goals (Speed, Safety)**
  - **2. Prioritizing Goals when there are more than one goals**

Structure of Utility Based Agent

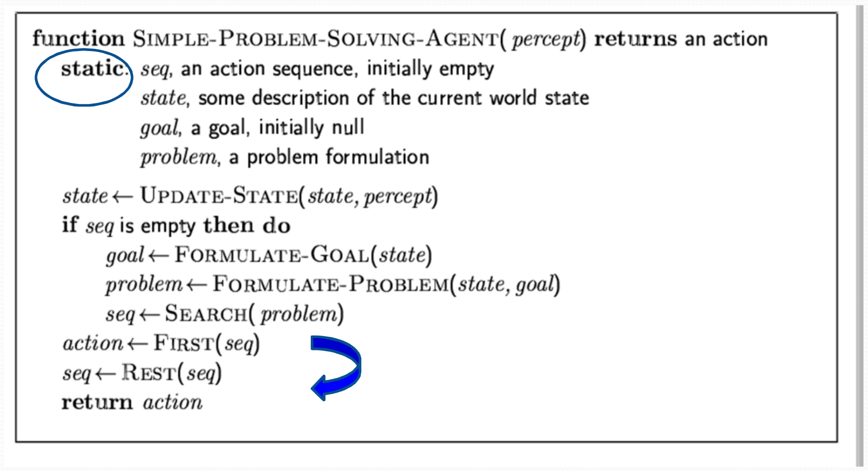

Agent
Sensors
Percepts
state
How world evolves
What my actions Do?
Utility
What the world is like now?
What it will be like if I do action A?
How useful this state is?
What action I should do now?
Actuators
Actions
Environment

Read about problem solving agents

Problem solving Agents

- A kind of goal based agents
- But also have capability to formulate (adapt to) Goal and problem
- Goal formulation:- Is first step and is based on current situation and the agents performance measures. (Goal: pune to mumbai, measure: safety)
- Problem Formulation:- Given a goal, decide actions and states.
- These agents try to find out sequence of actions that leads to goal state from current state. This process is called “search”.

Problem solving Agents



```
function SIMPLE-PROBLEM-SOLVING-AGENT(percept) returns an action
    static: seq, an action sequence, initially empty
            state, some description of the current world state
            goal, a goal, initially null
            problem, a problem formulation

    state ← UPDATE-STATE(state, percept)
    if seq is empty then do
        goal ← FORMULATE-GOAL(state)
        problem ← FORMULATE-PROBLEM(state, goal)
        seq ← SEARCH(problem)
    action ← FIRST(seq)
    seq ← REST(seq)
    return action
```

Problem solving Agents
-Problem Formulation

- **Problem Formulation:-** can be done with defining 4 components

1. **Initial state:-** what is initial state?
2. **Actions:-** Description of possible actions available to the agents. Generally, a successor-function(x) is used that returns a set of {action, successor} ordered pairs, where

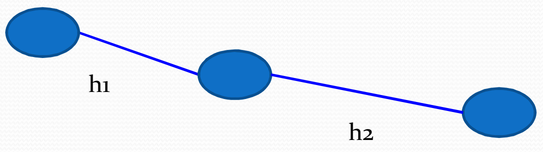
   Action- is legal action in state x

   Successor- is a state that can be reached from x

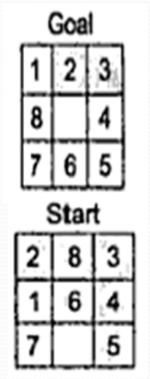
Problem solving Agents
-Problem Formulation

3. **Goal test:-** that determines whether a given state is goal state. Either explicit list of goal states is available or abstract properties.

4. **A Path cost function:-** that assigns a numeric cost to each path. Total cost is sum of costs of step paths from initial to goal state



Problem solving Agents
-8-puzzle



- **Goal Formulation**:-
- **Problem Formulation**:-

1 . Initial State:-

2. Actions:- [UP, LEFT, RIGHT, DOWN}

3. Goal Test:- **Goal state is given**

4. Path cost function:- **Cost 1 for each move**

Agent functions

- Very crucial part to implement
- From simple to very complex logic
- It may use many techniques like ANN, Fuzzy logic, Deep Learning and etc.
- Implementation can be sequential or parallel
- May be from a small embedded H/W to powerful HPC systems

References

1. Stuart Russell and Peter Norvig, " *Artificial Intelligence: A modern approach*", Pearson education
2. Eline Rich and Kevin Knight, "*Artificial Intelligence*", Tata McGraw Hill

Various distances which can be used for 8-puzzle

- Euclidean (L2 Norm) distance
- Manhattan distance
- Mahalanobis distance
- Jaccard distance
- Cosine distance
- Edit distance (https://www.geeksforgeeks.org/edit-distance-dp-5)
- Hamming distance
- P-Norm distance

1	2	3
8		4
7	6	5