Marvi Sultana Badhon

Q1. What is intelligence (Given scientific defination). Do you think following defination of intelligence constitutes a scientific defination. If not, justify your answer.

"Intelligence is the acquiring and applying knowledge".

A1: Intelligence: Intelligence is a measures of success/ performance of an entity in achieving its objectives or goals by interacting with its environment.

No. I don't think "Intelligence is the acquiring and applying knowledge" is not the scientific defination of artificial intelligence.

Intelligence is a measures of success or performance of an entity in acheeving its objectives or goals by interacting with its environment. The level of intelligence should be judge by considereering the difficulty of the goals and the success in acheiving them. It is measured by test, involving a varify of skill and knowledge. The main characteristics of intelligence are,

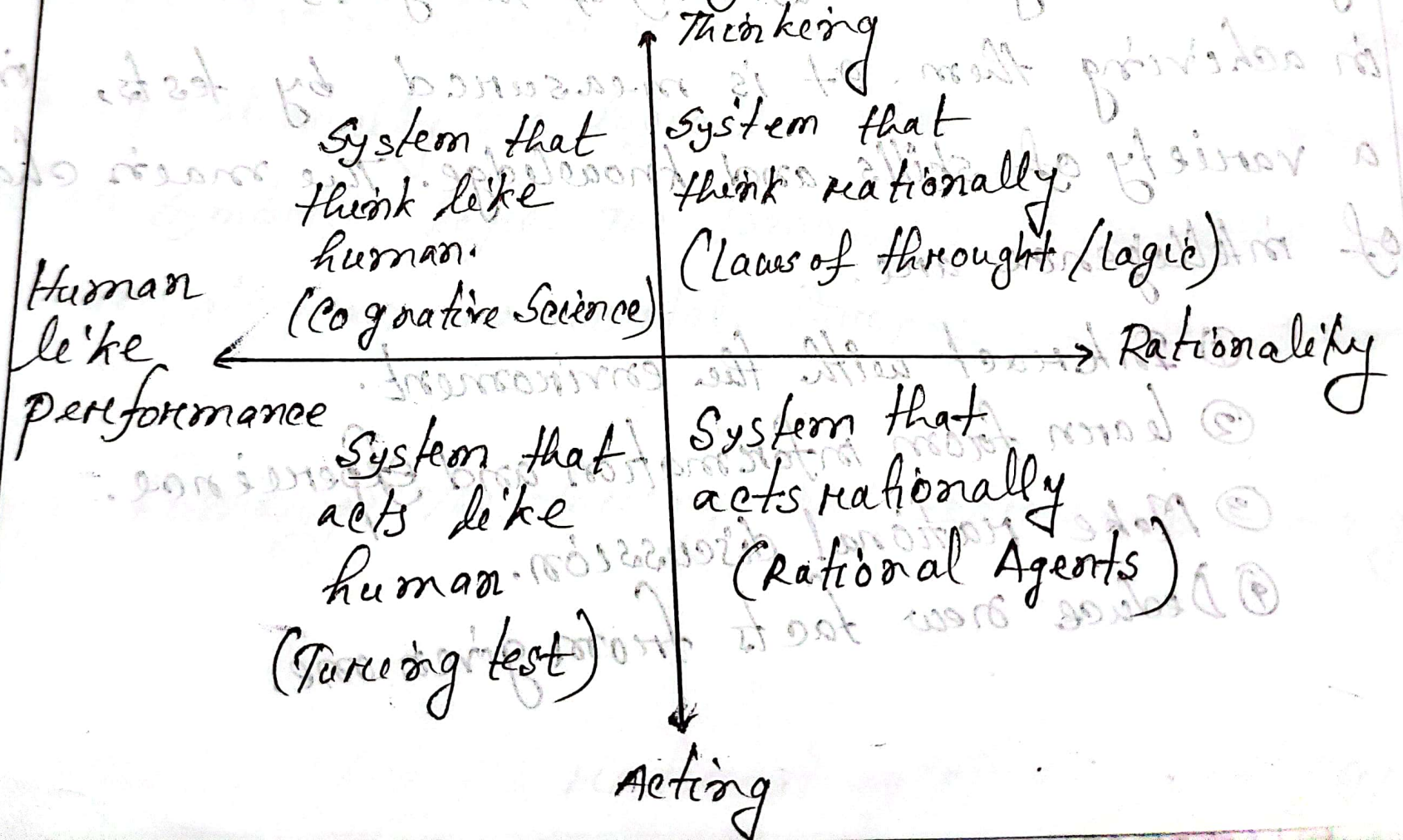
1. Interact with the environment.
2. learn from information and experience.
3. Make rational decision.
4. Deduce new facts from given one.

In the above defination it is said that "Intelligence is the acquiring and applying knowledge." But it is not fullfill the scientific defination. Intelligence is not only acquiring and applying knowledge, but also acheiving its objectives or goals by interacting with it's environment.

So, it is not the scientific defination of intelligence.

1(b) There are four approach in AI.

1. Thinking humanly
2. Thinking Rationally
3. Acting humanly
4. Acting Rationally



Thinking humanly: The cognative modeling approach, thinking like a human.

Thinking Rationally: It is the laws of through approach. Thinking rationally is a logical process. and its concludes based on symbolic logic. The famous syllogisms, "Socrates is a man; all man are mortal, therefore socretes is mortal."

Acting humanly: The turing test approach, acting like human.

Acting Rationally: The rational agent approach. It acts to achive high value and bring best possible outcome for any given task.

① It is more general than "laws of through" approach.
② It is more amenable (नम्र → flexible) to scientific development than human based approach.

that's why 4th is best.

Bade Achhe lagte hain

I've found new emotions
I don't know what this effect is!!
As someone has finally accepted me ♡

1(a).

① Percepts: Number of input in AI system.
② Actions: Output of AI system.
③ Goals: What the agent expect to achive
④ Environment: Agent is interacting with one.

(i)

Agent types	Percepts (Sensors)	Actions (Actuators)	Goals	Environment
Automated taxi Driver	camera, engine sensors, speedmeters, GPS, microphone.	steer, accelerate, brake, talk to passenger.	Safe, fast, legal, comfortable trip, maximum profits.	Roads, other traffic, weather, customers.

(ii) Characterize its environment:

① Accessible or inaccessible: The agents environment is inaccessible. If an agent sensor give it access to the complete state of the environment, then we say that the environment is accessible. Here, the environment is not getting the complete state of it. So, it is not accessible. The environment is inaccessible.

② Deterministic or nondeterministic: If the next state of the environment is completely determined by the current state and the action of the agent. Then the environment is determinis. So, the agent environment is non-deterministic.

sequencial.
→ memory require.

Episodic or non-episodic: In episodic only the current percept is required for the action. Every episode is independent of each other. In non-episodic an agent requires memory of past actions to determine the next best actions. The current decision could affect all future decisions. So, the agent environment is non-episodic.

Static or dynamic: In static, the environment don't change while an agent is acting. The environment may change over time in dynamic case. So, the agent environment is dynamic.

Discrete and Continuous: If there are a limited number of distinct, clearly defined percepts and actions we can say that the environment is ~~continuous~~ discrete. Here the number of percepts and actions are not limited. So the agent's environment is continuous.

finite number of action choices

↳ The action perform can't be number.

2019 1(a) previous (1)(a).

1(b) (or understand)

① learn from experience: Sophia and Jarvis learn from their experience. Their system or brain is made of an advanced AI technology, that enable them to learn understand and evolve through for every experience.

② Make sense out of ambiguous or contradictory message: Sophia and Jarvis dialoge is generated via a decision tree but it is integrated with these output uniquely. The response of sophia and Jarvis at any situation is very flexible. They can make sense ~~of~~ of ambiguous or contradictory message.

② Response quickly and successfully to new situation: Sophia's ~~us~~ responses are predefined-programmed. some response ~~to~~ seem to be generated by conversational algorithm. The output of the algorithm can sometimes be good and other time hilariously awkward. Which helps sophia and Jarvis to response quickly and successfully to new situation. For example, when

Badhon

Sophia first meet with Prime Minister Sheikh Hasina of Bangladesh Sophia by talking about her (Sheikh Hasina). Sophia side "You are the daughter of great leader the father of the nation Bangabandhu Sheikh Mujibur Rahman. Your grand daughter has the same name of mine Sofia". On the other hand Jarvis also respond quickly at any situation when it is necessary and concern and remind about future event according to situation.

④ Use reasoning to solved problem: Reasoning paradigm that exploits the specific knowledge collected from the previous experience to solve new problem in future. For using conventional algorithm both Sophia and Jarvis can do it.

⑤ Apply knowledge to manipulate the environment: Sophia and Jarvis both can predict extream weather environment, the situation what can be happening now, what should Do now, which can help the users to detect it early. But they can not apply knowledge to manipulate the environmet.

So, it can be said that, booth are most likely sign of intelligence.

⑥ Recognize the relative importance of different elements in a situation.

Q: Percepts, action, goal and environment (PAGE) of Mars,

Agent types	Percepts	Action	Goal	Environment
Mars Rover	Spect Sensor, visualize, hear, thermosensors, pixel of varying intensity, color.	Move, record, image capture, print catagorized scene.	Analize the environment of Mars, collecting sample, finding life, minimize cost	Mars, Robort.

(ii) Types of environment of Mars Rover.

(i) Inaccessible.
(ii) Non-deterministic.
(iii) Non-episodic / Sequencial.
(iv) Dynamic
(v) Continuous

* Partially observed
* Single agent.

(iii) What agent architecture is best for automated taxi driver agent. Explain.

Qu: What is Intelligence behaviour:

Intelligence behaviour: Intelligence behaviour is the capability of using one's knowledge about the world to make decisions in novel situations. Infact intelligence behaviour is—

① learn and understand from experience.
② Making sence of ambiguous and contradictory messages.
③ Response quickly and successfully to new situation.
④ Use reasoning to solve problem.
⑤ Apply knowledge to manipulate the environment.

Q: Do you think only sensing and acting will make a machine intelligence.

No, only sensing and acting will not make a machine intelligence.

An intelligence machine is a machine that can perform and complete its given task in the present of an unreliable and dynamic working environment. We will be called a machine intelligence when it has the following sign.

① Learn or understand from experience
② Making sense of ambiguous and contradictory message.

③ Response quickly and successfully to new situations.
④ Use reasoning to solve problems.
⑤ Apply knowledge to manipulate the environment.
⑥ Acquiring and apply knowledge to achieve goals.

Q: Internet shopping agent (PAGE).

Agent types	Percepts	Goals	Actions	Environment
	HTMP page/ text, graphics, scripts.	price, quality minimizing price and cost, appropriateness efficiency.	Display to user, follow URL, fill in form.	current and future www site, vendors and shippers.

Q Environment types:

① partially observable.
② partly deterministic
③ Sequential.
④ partly static // the world changes partly when the agent thinking.
⑤ Discrete.
⑥ Single Agent.

2017 5(b)

Q: AI is the study and construction of rational agent.

A: Yes, AI is the study and construction of rational agent. A rational agent could be anything that makes decisions, as a person, firm, machine or software. It carries out the actions with best possible outcome after considering past and current percepts.

Rational agent always does the right thing that makes the agent most successfull. For each possible percepts sequences, a rational agent should select an action that provides best possible outcomes, maximize its performance.

Infact rational agent maximize its expected performance measure.

For an AI agent, the rational agent is most important because in AI reinforcement learning algorithm, for each best possible action, agents gives the positive reward and for each negative action, an agent gives negative reward. It is more general than laws of through approach and more amenable in scientific development than approach based on human behaviour Because it is crearly defined and more general.

5(c) Features required for passing the turing test

① Natural language processing: NLP is required to communicate with interrogator in general human language like English.

② Knowledge representation: To store and retrieve information during the test.

③ Automated reasoning: To use the previously stored information for answering the questions.

④ Machine learning: To adapt new changes and can detect generalized patterns.

⑤ Vision (For total turing test): To recognize the interrogator actions and other object during test.

⑥ Motor control (for total turing test): To act upon objects if required.

5(d) previous.

NKAM

Example of a data structure that can be used in represent a state space both in explicit and implicit way.

① Node

② Open list

③ Close lest

Search problem is a problem of searching an specific goal with repect to a sequence of action. The search problem can be solved by some algorithm. such as, BFS, DFS, A*, Greedy search etc.

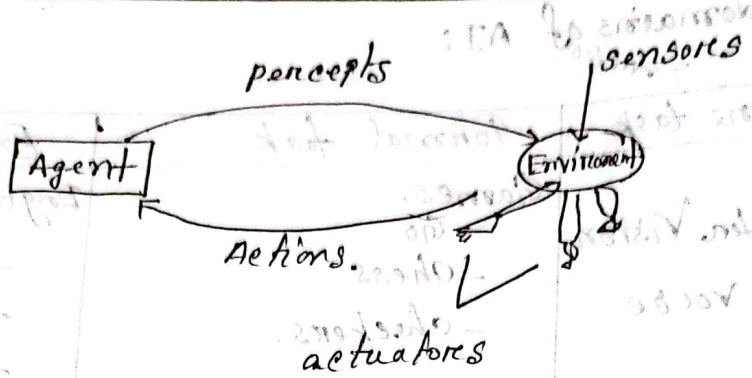
2016

1(a) Task domains of AI:

Mundane task	Formal task	Expert task
Perception - Computer Vision - Speech, voice	Games, - Go - Chess - checkers.	Engineering - Design - Fault Finding - Manufacturing - Monitoring
Natural language processing - Understanding - language Generation - Language translation	Mathematics - Geometry - logic - Integration - Differentiation	Scientific analysis.
Common Sense Reasoning Planning	Theorem Proving.	Financial Analysis
Robot Control.		Medical Diagnosis

1(b) Intelligent agent: An intelligent agent is anything that can viewed as preciving is environment through sensores, and acting upon the environment through actuators.

① Must have the ability to perceive the environment.
② The observation must be used to make decision.
③ Decision should result in an action.
④ The action must be rational agent.



1(b) There are five basic kinds of agent.

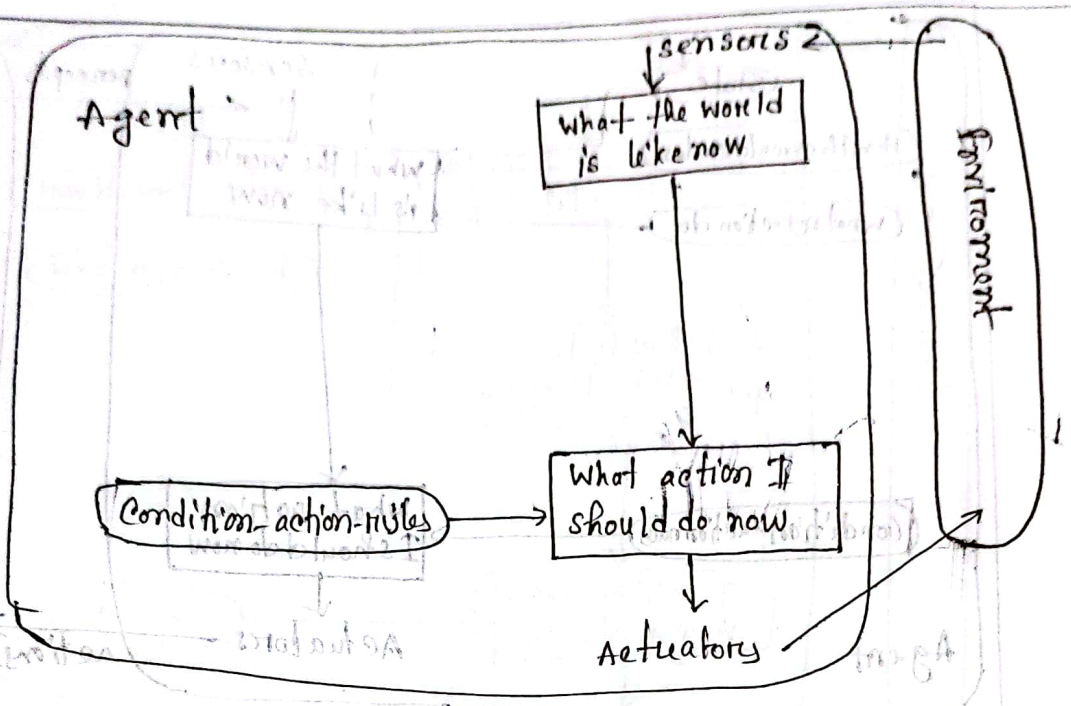
① Simple reflex agents.
② Model-based agents.
③ Goal-based agents.
④ Utility-based agents.

All of which can be generalized into learning agents that can improve their performance and generate better actions.

⑤ Learning-based agents.

① Simple reflex agents: It acts only ~~on~~ based on current state ~~only~~ and ignoring the percept history.

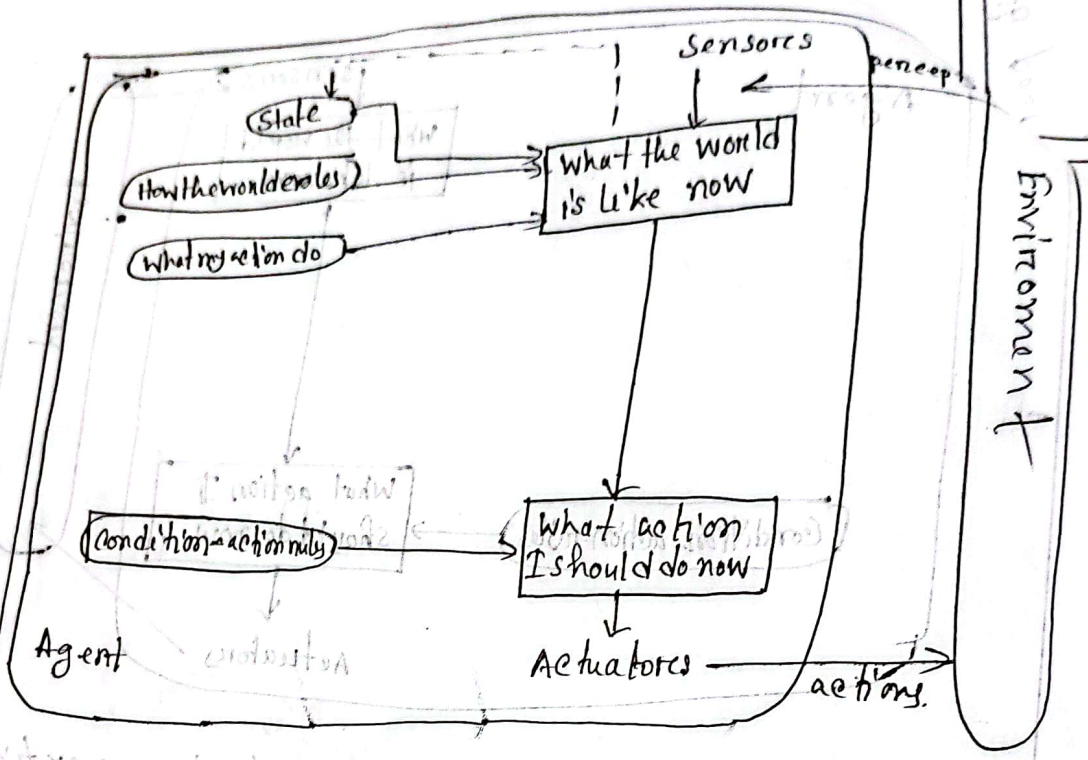
- It is simple but limited.
- Can only work if the environment is fully observable, that is the correct action is based on the current percept only.



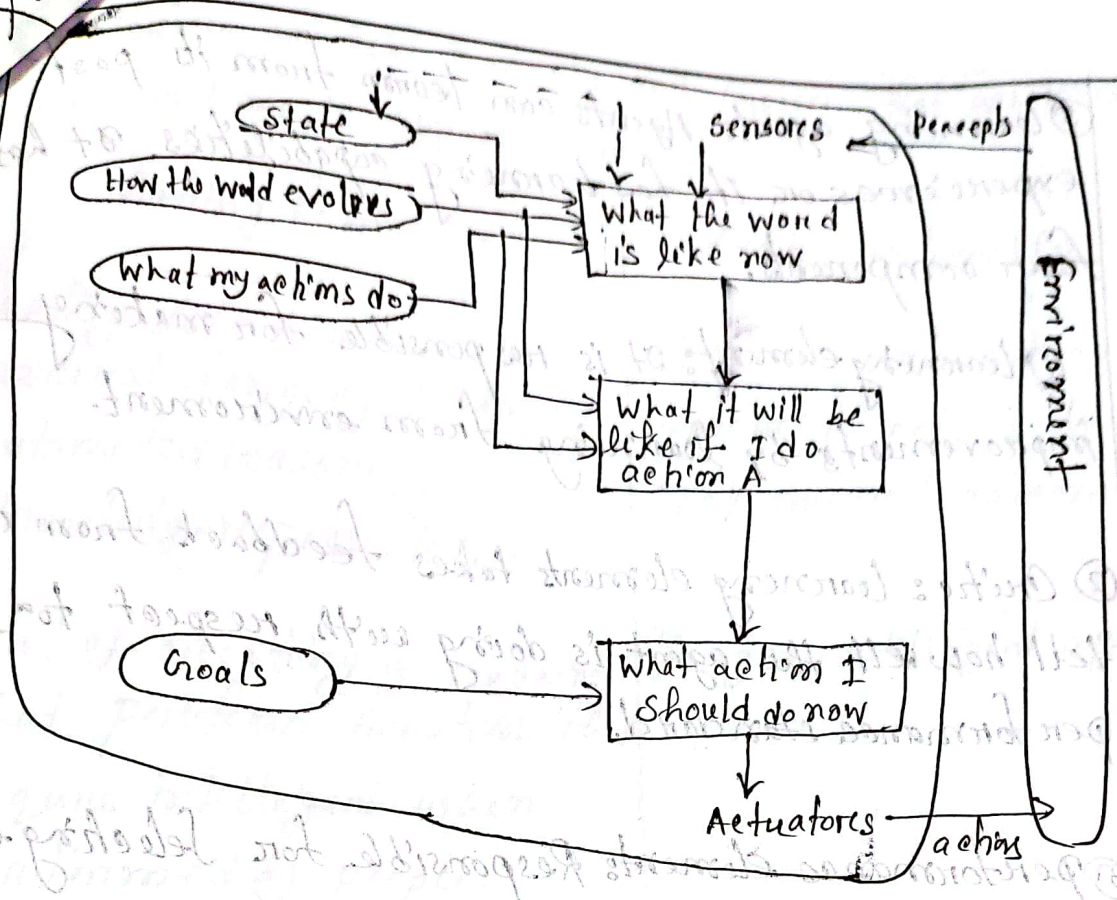
② Model-based reflex agent: It works in a partially observable environment. and track the situation. It has two important factors.

① Model: It is knowledge about "how things happen in the world", so it is called model based.

② Internal state: It is a representation of the current state based on the percept history.



③ Goal-based agents: The knowledge of the current state environment is not always sufficient to decide for an agent to what to do. The agent needs to know its goal which describes desirable situations. They choose an action, so that they can achieve the goal.



④ Utility-based agents: These agents are similar to the goal-based agent but provide an extra component of utility measurement which makes them different by providing a measure of success at a given state. It act based not only goals but also the best way to achieve the goal. It is useful when there are multiple possible alternatives, ~~to perf~~ and an agent has to choose in order to perform the best action.

2(a)

⑤ learning agent: Agents can learn from its past experiences or it has learning capabilities. It has four components.

① learning element: It is responsible for making improvements by learning from environment.

② Critic: Learning elements takes feedback from critic. tell how well the agent is doing with respect to fixed performance standard.

③ performance element: Responsible for selecting external action.

④ Proble generator: Responsible for suggesting actions that will lead to new and informative experience.

④ learn
& analyze performance
& Look new way to improve performance.

2015

system act like humanly and System that act rationally, here 2nd one is appropriate.

Acting humanly	Acting Rationally
Acting like human- natural language processing, Automated reasoning Knowledge representation.	Study of rational agents expected to act as to achieve best expected outcome.
art of creating a machine that perform functions that require intelligent when performed by people.	Design of intelligent agent.
The study of how to make computers do things at which, at the moment, people are better.	Intelligent behaviour.

1(c) previous.

2020

2(a) Formulate a search problem: Search problem is the process of deciding what action should be taken to achieve the formulated goal. Components involve in problem formulation,

① Initial state.
② Actions.
③ Transition model.
④ Goal test
⑤ path cost.

S: the full of states
S_0: Initial states
A: S-S set of operators
G: the set of final states.

① Initial state: Initial state that the agent starts in.

② Actions: It gives the description of all the possible actions to the agent.

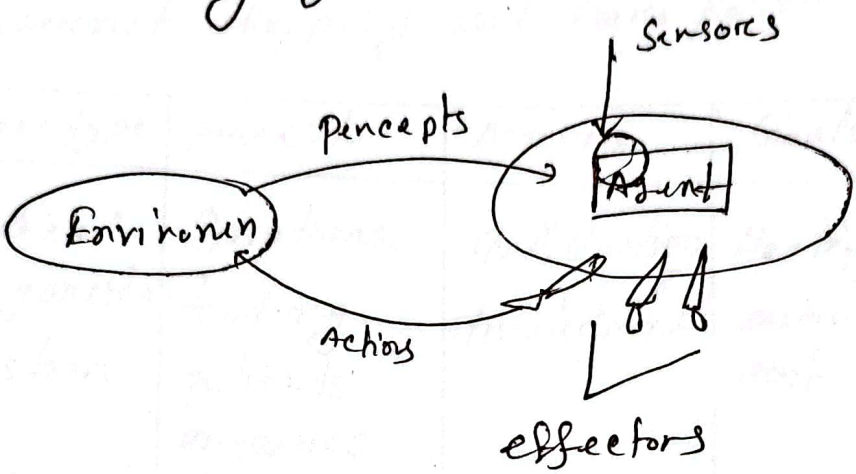
③ Transition model: A description of what each action can do, can be represented as a transition model.

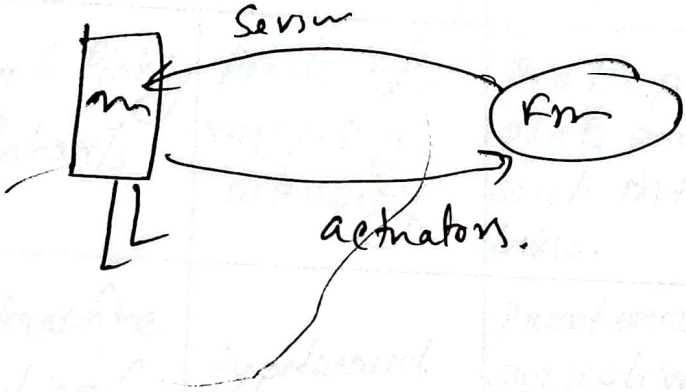
④ Goal test: It determines whether a give state is a goal state.

⑤ path cost: Assign numerical cost at each path.

Example: Note.

An intelligent agent is an autonomous entity which act upon an environment using sensors and actuator for achieving goals.





2018
3(d). Medical diagonestic System, Park Picking Robort, Interactive english tutor, Vacuum cleaner, Internet shopping and Mars Rover.

Agent type	Percepts	Actions	Goals	Environment
Medical Diagonestic System	Symptoms, finding, patients answers	Test, Question, treatedment	Heathy patient, meinimize cost.	Hospital patient.
Park Pinking Robots	Pixels of varying intensity	Pick up parts and sort into bins.	place parts in correct bins.	Conveyonbelt with parts.
Interactive English tutor	typedwored	Preint exercises, suggestions, corrections.	Maximized student's score on test	Sets of students.
Vacuum cleanner	Locations and contents	Left, light, Suck, Noop.	amoat of dirty clean, meinize time, electricity consumed, etc.	Room

	Accessible	Deterministic	Episodic	Static	Discrete
Medical Diagnosis system	No	No	No	No	No
Part picking Robot	No	No	Yes	No	No
Interactive english tutor	No	No	No	No	Yes.
Vacuum cleaner					

Badhon —

I'will go wherever you go
I'll be on your side,
I'm am the season you're
the seasonal wind,
beloved ♥

* Which agent architecture is best for the Mars Rovere.

For low level navigation model based ~~on~~ reflex agent is suitable. For routing planning and experimentation etc, combination of goal-based and utility base architecture is best for the Mars Rover.

In the single reflex agent architecture, the agent takes decisions on the basis of the current percepts and ignore the rest of the percept history. These agents only succed in the fully observable environment. But for the best possible outcome agent need to kept the percept history.

Model based ~~Re~~flexl agent: It can work in the partially observable environment and track the situation. It has two important facts.

① Model: It is know about "how things happen in the world"

② It is representation of current state based on percept history.

So, for the low level navigation model this agent architecture can be used.

Goal based agent: The knowledge of the current state environment is not always sufficient to decide for an agent to what to do. Goal based agent expand the capabilities of the model based agent by having the "goal information".

In utility based architecture, these agent are similar to the goal based architecture but provide and extra component of utility measurement. It is not based only the goals but also the best way to achieve them. It is useful when there are multiple possible alternatives and an agent has to choose in order to perform the best action.

So combination of goals and utility is best for the more more.

2020

2(a) Search problem: Search problem is a problem of searching a specific goal state with respect to a sequence of actions. The search problem can be solved by some algorith such as, DFS, BFS, A*, Greedy etc.

Formulate a search problem: Search problem is the process of formulating what action should be taken to achive the desired goal. Components involved in the search problem formulation—

① Initial state: Initial state that the agent start in.

② Actions: It gives the descreption of all the possible action to the agent

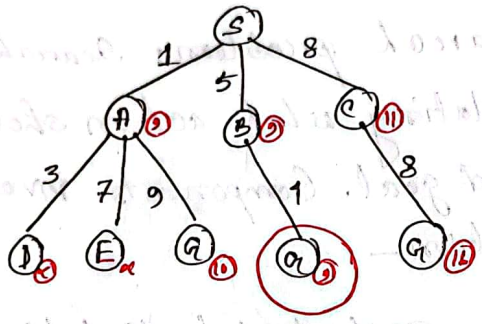
③ Transition function: A descreption of what each action can do.

④ Goal test: It determine whether a given state is in a goal state.

⑤ Path cost: Assign numerical cost at each path.

Below Give an example of data structures that can be used represent a state-space both in explicit and implicit way.

Using A* search algorithm,



S — 8
A — 8
B — 4
C — 3
D — ∞
E — ∞
G — 0

Goal state.

cost of path + heuristic

node exp	open list	close list
	{S(8)}	{ }
S	{A(9), B(9), C(11)}	{S(8)}
A	{B(9), G(10), C(11), D(inf), E(inf)}	{SA}
B	{G(9), G(10), C(11), D(inf), E(inf)}	{SAB}
G	{G(10), C(11), D(inf), E(inf)}	{SABG}

Solution path → SBG

cost → 9

Expended node → 4

Using the expended node, open list and close list. we can represent a state space in explicit and implicit way.

2(b)

Find initial state, goal test, successor function and cost function.

State: locations/cities

(i) For travelling salesman problem:

i) Initial state: starting point (No cities is visited)

ii) Successor function: move from one location to another.

iii) Goal test: All the location visited. agent at initial state.

iv) Path cost: Distance between location.

(ii) Missionaries and cannibals problem:

i) Initial state: 3 missionaries, 3 cannibals and a boat are on the near river. <3,3,1>

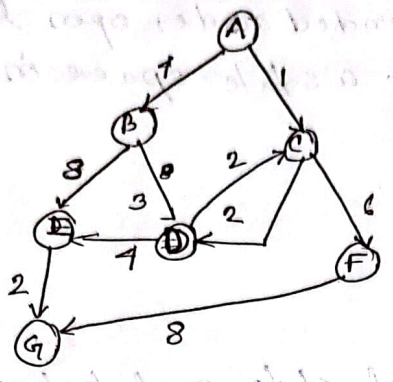
ii) Successor function: Move boat containing some set of occupants across the river to the other side.

iii) Goal test: Move all missionaries and cannibals across the river. <0,0,0>

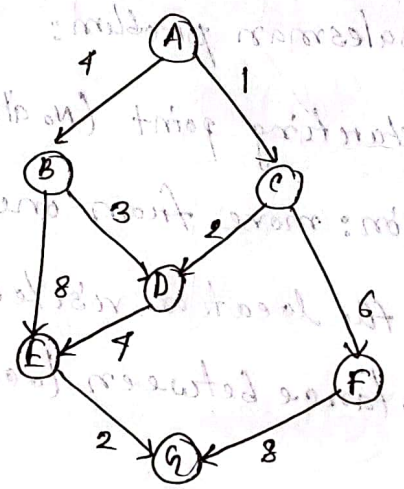
(iv) Path cost: Number of crossing.

2020

2(c)

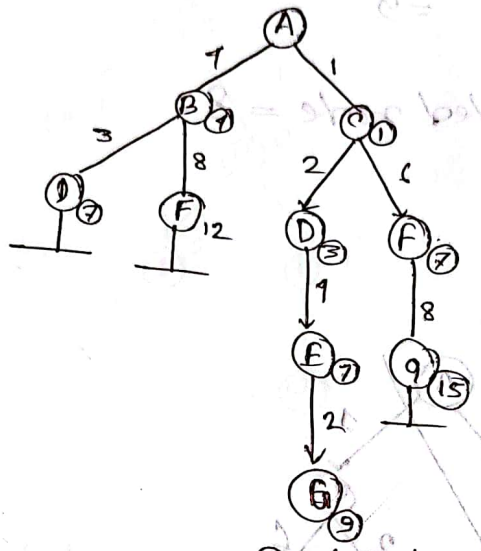


(i) State space of the problem,



(ii)

Uniform cost search:



Goal state.

(iii)

Expended node	open list	close list.
	{A(0)}	{ }
A	{C(1), B(4)}	{A}
C	{D(3), B(4), F(7)}	{AC}
D	{B(4), F(7), E(7)}	{ACD}
B	{F(7), E(7), D(7), F(12)}	{ACDB}
F	{E(7), D(7), F(12), G(15)}	{ACDBF}
E	{D(7) G(9), F(12), G(15)}	{ACDBFE}
D	{D G(9), F(12), G(15)}	{ACDBFED}
G	{F(12), G(15)}	{ACDBFEDG}

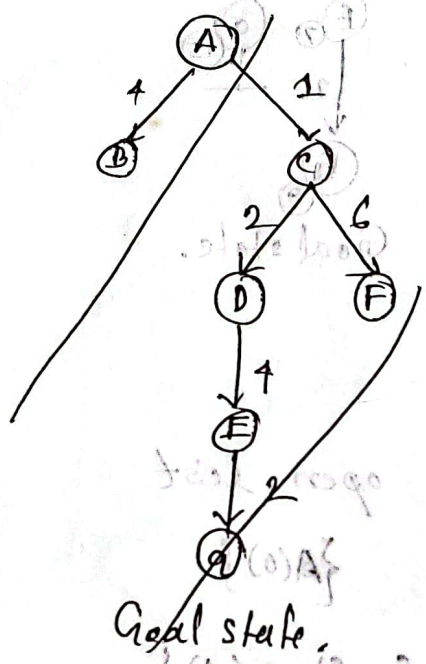
Solution path found ACDEG

cost = 9

Number of expended node = 8

o Greedy Search

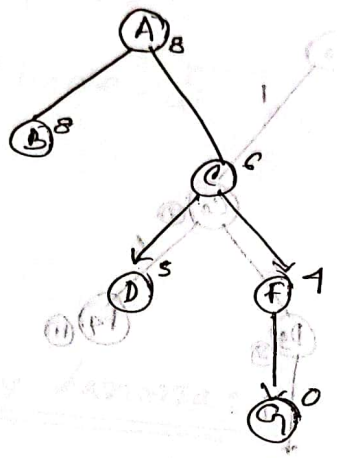
(ii)



(iii)

Expended node: Open list Close List

Greedy algorithm → heuristic cost according ordered



A – 8
B – 8
C – 6
D – 5
E – 1
F – 4
G – 0

$f(n) = h(n)$

node expended	open list	close list
	{A(8)}	{ }
A	{C(6), B(8)}	{A}
C	{F(4), D(5), B(8)}	{AC}
F	{G(0), F(4), D(5) B(8)}	{ACF}
G	{F(4), D(5), B(8)}	{ACFG}

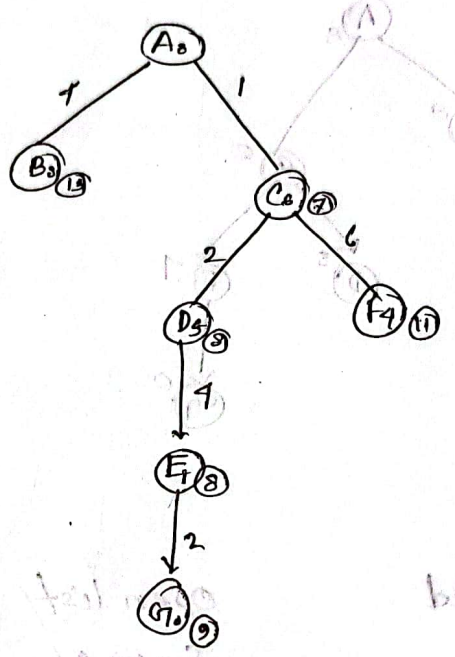
Solution path found ACFG.

cost =

4 node is expended

A* Search

(ii)



Goal state

(iii)

node expended	Open list	close list
	{A(8)}	{ }
A	{C(7), B(13)}	{A}
C	{D(8), F(11), B(13)}	{AC}
D	{E(8), F(11), B(13)}	{ACB}
E	{G(9), F(11), B(13)}	{ACBE}
G	{F(11), B(13)}	{ACBEG}

Solution path → ACDE G

cost → 9

Expended node = 5

2019

2(b) Monkey banana:

i) Initial state: As describe in the text.

ii) Successor function: Hop off crate, Hop on crate, Push crate from one spot to ~~non~~ another, stack one to another, Walk from one spot to another. Grab banana (if standing on the crate).

iii) Goal test: Monkey has banana.

(iv) Cost function: Number of actions.

Solution path → ACDE G

cost → 9

Expended node = 5

2019

2(b) Monkey banana:

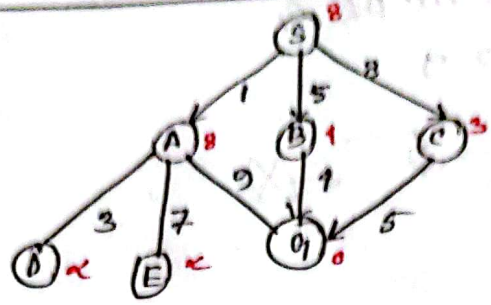
i) Initial state: As describe in the text.

ii) Successor function: Hop off crate, Hop on crate, Push crate from one spot to ~~non~~ another, stack one to another, Walk from one spot to another. Grab banana (if standing on the crate).

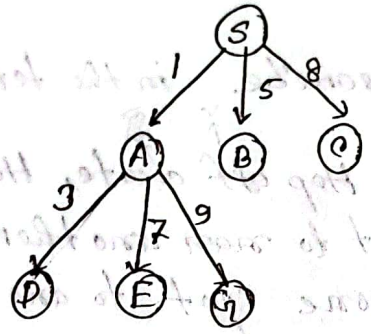
iii) Goal test: Monkey has banana.

(iv) Cost function: Number of actions.

2019
2 (c)



Depth first Search:



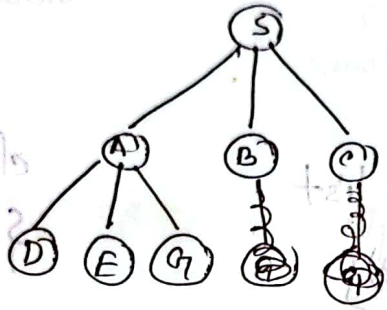
expended node	openlist	close list
	{S}	{}
S	{A,B,C}	{S}
A	{DEGBC}	{SA}
D	{EGBC}	{SAD}
E	{GBC}	{SABE}
G	{BC}	{SABEG}

Solution path → SAG

path cost → 10

No. of expanded node → 5.

Iterative deeping:



— L_0 (d=0)

— L_1 (d=1)

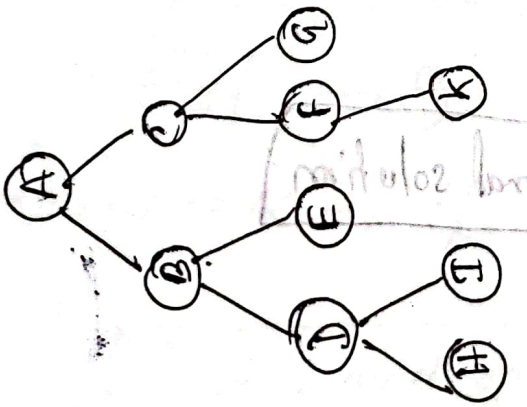
— L_2 (d=2)

Iteration 1 : S

Iteration 2 : SABC

Iteration 3 : SABCDEG.

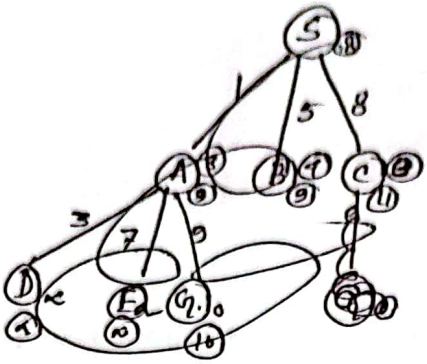
A

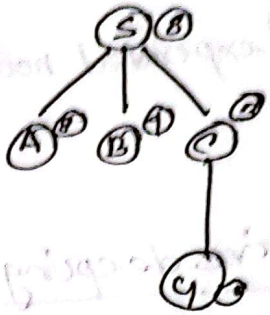


I_3 : ABDECFG

I_4 : ABDHIECFKG.

Greedy Search





Goal state.

expended node	Open list	close list.
	{S(8)}	{ }
S	{C(3), B(4), A(8)}	{S}
C	{G(0), B(4), A(8)}	{SC}
G	{B(4), A(8)}	{SCG}

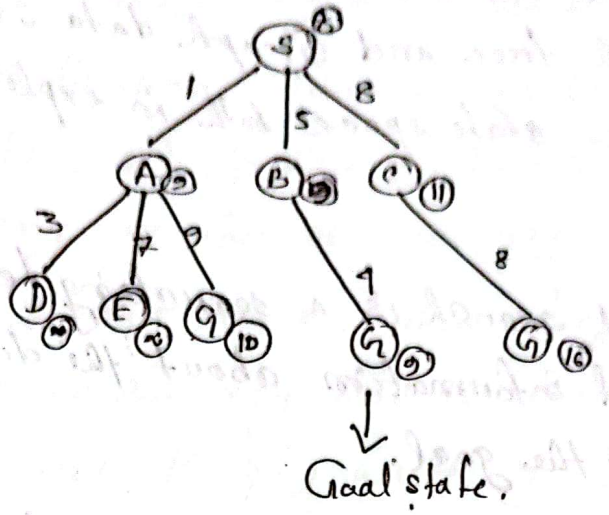
Solution path : SCG.

Cost : 13

3 node is expended.

First but not optimal solution

A* search:



S – 8
A – 8
B – 1
C – 3
D – ∞
E – ∞
G – 0

Expended node	Open list	Close list
	{S(8)}	{ }
S	{A(9), B(9), C(11)}	{S}
A	{B(9), G(10), C(11)}	{SA}
B	{G(9), G(10), C(11)}	{SAB}
G	{G(10), C(11)}	{SABG}

Solution path → SBG.

Cost → 9

Expended node → 4.

$\boxed{\text{First, and optimal solution}}$

2018
2(a) Compare the basic search algorithm which are used in both tree and Graph data structure, to represent a state space both in explicit and implicit way.

An uninformed search is a searching technique that has no additional information about the distance from the current state to the goal.

An informed search is a searching algorithm that has additional information about the estimate from the current state to the goal state.

Basis of comparison	Informed Search	Uninformed Search
Basic knowledge	Use knowledge to find the steps to the solution.	No use of knowledge
Efficiency	Highly efficient, consumes less time and cost.	Efficiency is mediatory
Cost	low	Comparatively high
Performance	Find solution more quickly.	Speed is slower than informed search
Algorithm	Heuristic. depth search, breath search, A* Search etc.	Depth First Search, breadth First Search, Uniform cost Search

2018

2(b)

(ii) Initial state: Jugs have values [0,0,0].

Successor function: Given values [x,y,z], generate [12,y,z], [x,8,y], [x,y,3] (by filling); [0,y,z], [x,0,z], [x,y,z] by emptying. or for our two jugs with the current value x, y and y pour into x; this changes the jug with x to the minimum of x+y and the capacity of the jug. and decrement the jug with y by the amount gained by the first jug.

Goal test: Measure out exactly one gallon.

path cost: number of actions.

2018

2(b)

(ii) Initial state: Jugs have values [0,0,0].

Successor function: Given values [x,y,z], generate [12,y,z], [x,8,y], [x,y,3] (by filling); [0,y,z], [x,0,z], [x,y,z] by emptying. or fore one two jugs with the current value x,y and y pour into x; this changes the jug with x to the minimum of x+y and the capacity of the jug. and decrement the jug with y by the amount gaind by the first jug.

Goal test: Measure out exactly one gallon.

path cost: number of actions.

1(a)
(ii) Initial state: No regions colored.
Actions : Assign a colore to an uncolored region.
Transition model: The previously uncolored region has the assigned colore.

Goal test: All regions colored, and no two adjacent regions have the same colore.
Cost function: Number of assignments.

2020

3(a) Note.

3(b) The iterative deepening algorithm is a combination of DFS and BFS algorithm. The search angorithm find out the best depth limit and does it gradually increasing the limit until a goal is found.

This algorithm ~~combines the~~ performs DFS search up to a certain "depth limit" and it keeps increasing the depth limit after each iteration until the goal node is found.

1(a) Search problem: A search problem consist of a search space, start state and goal state. and succe function.

2017

Why ~~Problem~~ problem formulation must follow goal form

In goal formulation, we decide which aspects we are intere in and which aspects can be ignored. In the goal formula process, the goal is to be set and we sould assess the states in which the goal is satisfied. In problem formul we decide how to manipulate the important aspects a ignore the others, So, without doing goal formulation if we do the problem formulation, we would not know what to include in our problem and what to leave and what should be acheived. So, problem formulation must follow goal formulation. That means Problem Formulation must be done only after the goal formulat is done.

2017

1 (b) (i) Four color, need to color a planar map. We have to color so that no two adjacent region ~~states~~ have the same color.

(i) Initial state: No ~~color~~ regions is colored. $\{w, x, y, z\}$

(ii) Action (3rd ed.) / Successor (2nd ed.): ~~The previous uncolored region has the assigned color~~. Assign a color to an uncolored region.

(iii) Transition model (3rd ed): The previous uncolored region has the assigned color.

(iv) Goal Test: All region colored, and no two adjacent regions has the assigned color.

(v) Cost function: Number of assignment

$\{w \neq x, w \neq y, w \neq z, x \neq y, x \neq z, y \neq z\}$

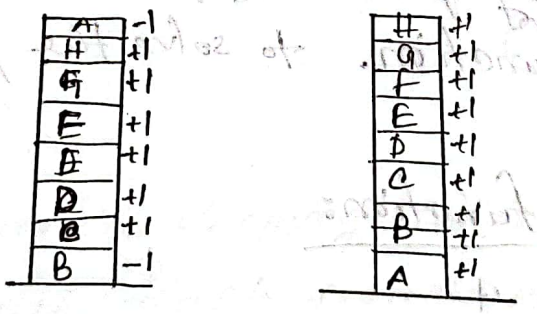
α

2020
3(a)

Heuristic function: Heuristic function is a function which is used in informed search. and it finds the most promising path. It estimates how close a state is to the goal state. It is represent by h(n). And it calculates the cost of an optimal path between the pair of states. The heuristic method, might not always give the best solution, but it guaranteed to find a good solution in resonable time. This technique always use to find solution quickly.

Below show that hill climbing procedure is failed with the local heuristic function but works perfectly with the global heuristic function:

local heuristic function:

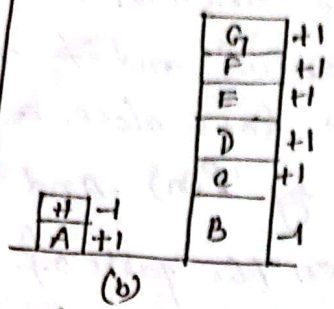


Initial state score = (8-2)
= 6

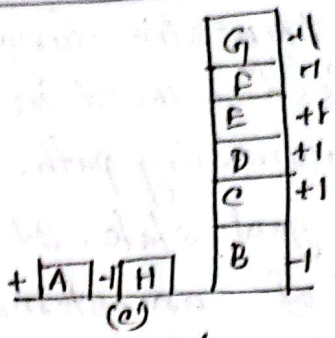
Goal state score = 8
A, B, C, D, E, G, F, G, H all are correct.

C, D, E, F, G, H correct = 6

B, A wrong = 2



Next state score = (6-2)
= 4



Next state score = (6-2)
= 4.

case a : Initial state score = (6-2) = 4

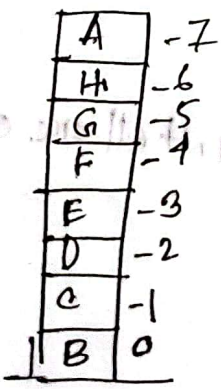
case b : Next state score = (6-2) = 4

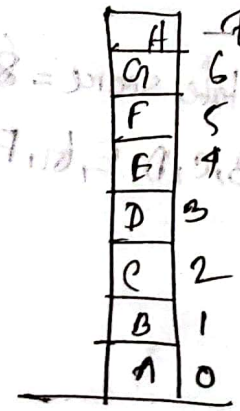
case c : Next state score = (6-2) = 4

All three cases a, b, c have no better score than the current state = 6

So, the hill climbing procedure is failed with the local heuristic function. to solve the proble.

Global heuristic function:





Initial state score = -28

C = -1 - - .

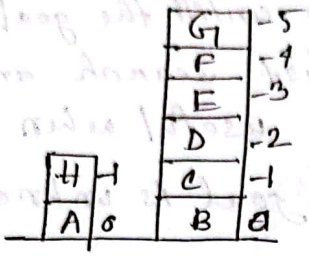
A = -7

Goal state score = 28

B = 1 . - -

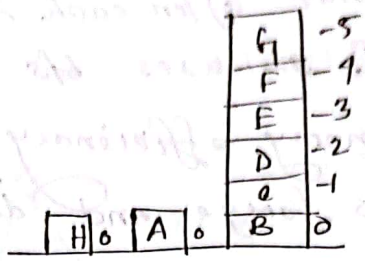
H = 7.

For each block add 1 point for every block in that support structure. and Increment -1



(b)

Score = -16



Score = -15

case a: The initial state score = -28

case b: Next state score = -16

case c: Next state score = -15

No local minimum problem.

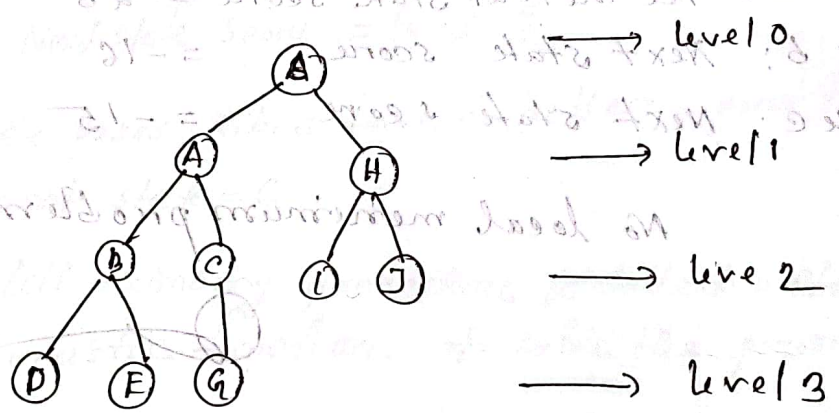
2020

3 (c) Iterative deeping (ID) is the combination the benefits of breadth first search and depth first search.

The iterative deeping is a combination of dfs and bfs algorithm. This search algorithm find out the best depth limit and does it by gradually increasing the limit until a goal is found.

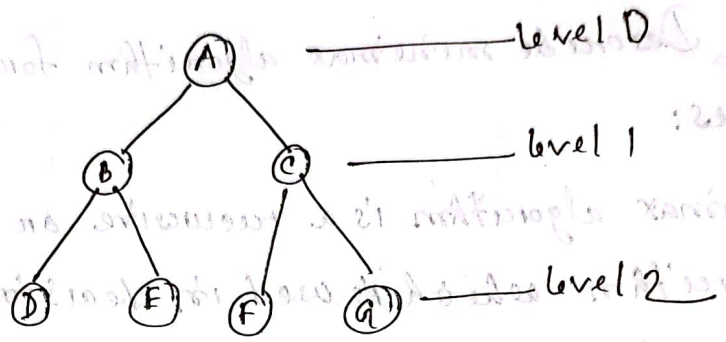
This algorith performs depth first search up to a certain "depth limit". and it keeps increasing the depth limit after each iteration untill the goal node is found. It uses bfs for fast search and dfs for memory efficiency. ID is useful when search space is large and depth of goal is unknown.

Example: following tree structure showing the IDDFS. It performs various iteration (dfs) until ~~the~~ it doesn't find the goal node.



1st iteration: ~~A~~ S
2nd iteration: ~~ABH~~ SAH
3rd " : ~~SABCIJ~~ ~~AB~~ SABCHIJ.
4th iteration : ~~AB~~ SABDECGHIJ.
At the forth iteration the algorithm find the goal node.

BFS . level order Search.
. left to Right.



ABCDEFG

2020

3(c) Breadth first search (where each step has the same cost) is an example of A* algorithm. It is simple written as,

$$f(n) = g(n) + 0$$

A* is admissible if it uses an admissible heuristic and $h(n) = 0$. So, BFS is also admissible when $h(n) = 0$.

Breadth first Search takes no account of distance to the goal because a zero estimate cannot possibly be an overestimate of that distance. it has to be admissible. This means that

This means that BFS can be seen as a basic example of Algorithm A*.

2020
3(d) Describe minimax algorithm for searching game trees:

Minimax algorithm is a recursive or backtracking algorithm which is used in decision making and game theory.

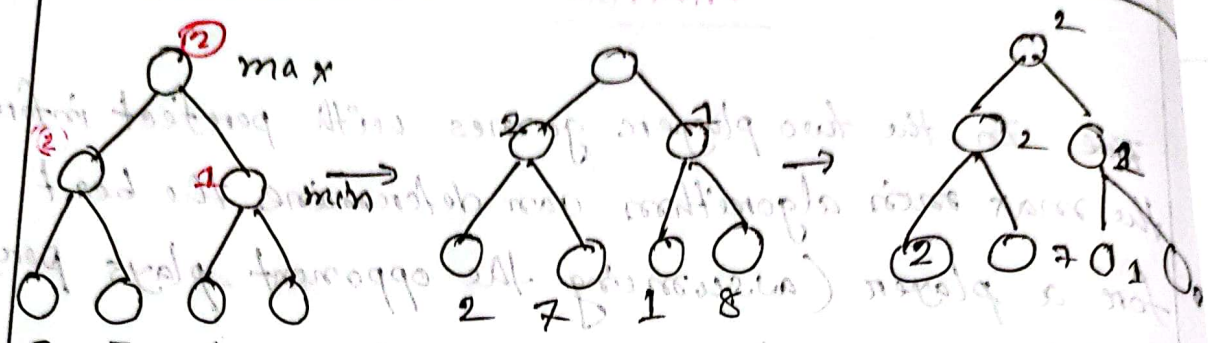
MiniMax

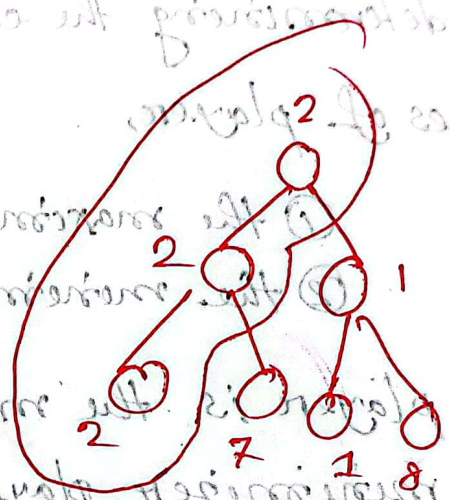
~~The~~ In the two player games with perfect information, n.d. the max min algorithm can determine the best move for a player (assuming the opponent plays perfectly) by determining the entire game tree. There are two types of player,

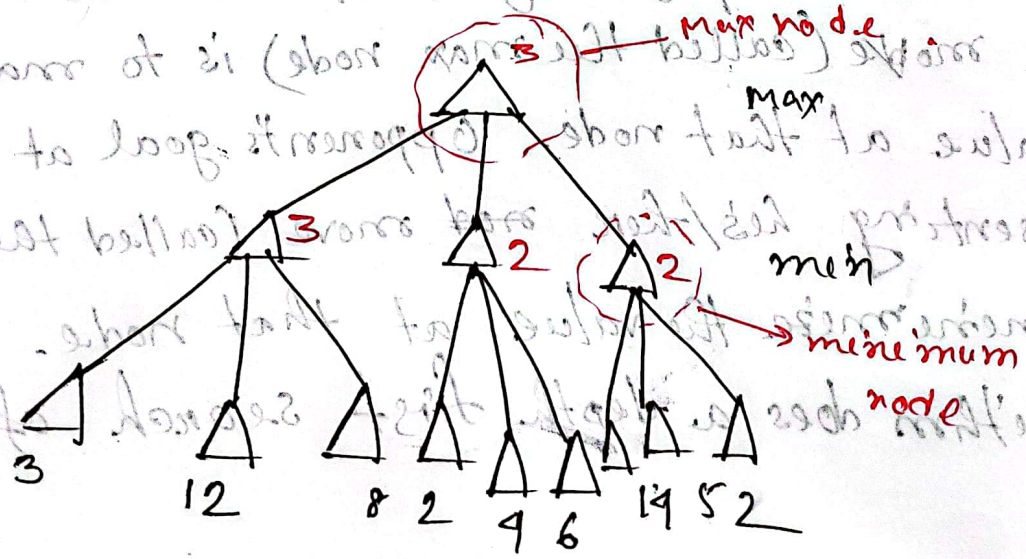
① the maximizer
② the minimizer.

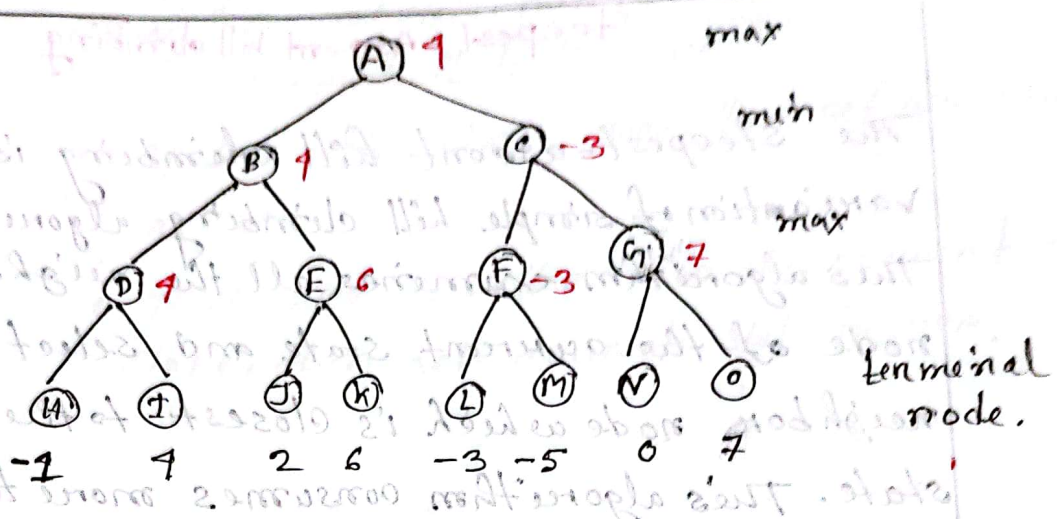
A player is the maximizer and his/her opponent is the minimizer. player goal at the node representing your move (called the max node) is to maximize the the value at that node. Opponent's goal at the node representing his/her ~~not~~ move (called the min node) is to minimize the value at that node. The maxmin algorithm does a depth first search of the game tree

Suppose that the maximizer takes the first turn, the minimizer takes the second turn and so, on.


max
min
2 7 1 8


2
2
1
2 7 1 8


max node
Max
min
minimum node
3 12 8 2 4 6 14 5 2



Perfect order to win the game ⇒ $\boxed{A \rightarrow B \rightarrow D \rightarrow I}$

Disadvantag:

① If the game tree is large, the procedure become too complex.

So, use α, β prunning

2019 3(a)

Steepest - Ascent hill climbing

The steepest-ascent hill climbing is a variation of simple hill climbing algorithm. This algorithm examines all the neighbouring node of the current state and select one neighbor node which is closest to the goal state. This algorithm consumes more time as it searches for multiple neighbors.

Algorithm

step 1: Evaluate the initial state, if it is goal state then return success and stop, else make current state as initial state.

Step 2: loop untill a solution is found or the current ~~the~~ state does not change.

a. let succ be a state such that any successor of the current state will be better than it.

b. for each operator that applies to the current state:

a. Apply the new operator and generate a new state.

b. Evaluate the new state.

c. If it is goal state, then return it and quit, else compare it to the succ.

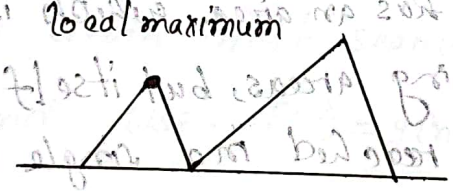
d. If it is better than succ, then set new state as succ.

e. If the succ is better than the current state, then set current state to succ.

Step 5: Exit.

problem in ~~enem~~ hill climbing and solution:

① Local maximum: A local maximum is a peak state in the landscape which is better than each of ~~the m~~ its neighboring states, but there is another state also present which is higher than the local maximum.

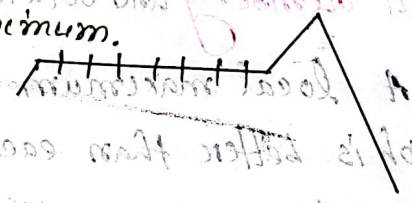


Solution: Back tracking technique can be ~~deo~~ a solution of the local maximum. Create a list of the promising path so that the algorithm can back track the search space and explore other paths as well.

2. plateau: A plateau is the flat area of the search space in which all the neighbor states of the current state contains the same value, because of this algorith does not find any best direction to move. A hill climb search might be lost in the plateau area.

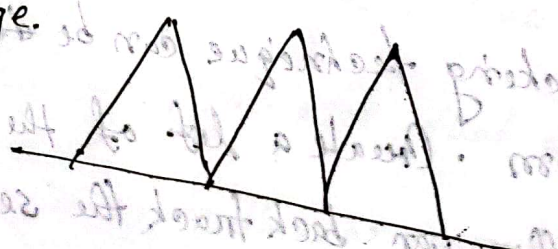
Solution: To solve this problem, Randomly select a state which is far away from the current state so it is possible that ~~that~~ the algorithm could find non-plateau region.

Flat maximum.



3. Ridges: A ridge is a special form of the local maximum. It has an area which is higher than its surrounding areas, but itself has a slope and can not be reached in a single move.

Ridge.



Solution: Wi
indifferen t

3(b) Differ

DFID: It
the drawba
advantages

DFID expan
any nodes
a shortes
state.

It per
some I

Solution: With the use of bidirectional search, or by moving in different direction we can improve the's problem.

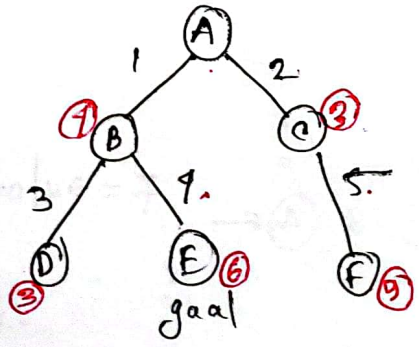
3(b) Differentiate between DFID and IDA*.

DFID: It suffers (Depth first iterative Deepening) : neither the drawbacks of BFS nor of DFS on tree. It takes advantages of both the strategies.

DFID expands all nodes at a given depth before expanding any nodes at greater depth, it is guaranteed to find a shortest-length solution from initial state to goal state.

IDA*

It perform depth first search with limited to some f-bound. Use. $F(n) = g(n) + h(n)$



h(n)

A	7
B	3
C	1
D	0
E	1
F	2

least value = 3

hence, threshold = 3

Stat at A → 3 < 7 ✗

next least threshold = 4.

Stat at A → 4 < 7 ✗

next threshold = 5

Stat at A → 5 < 7 ✗

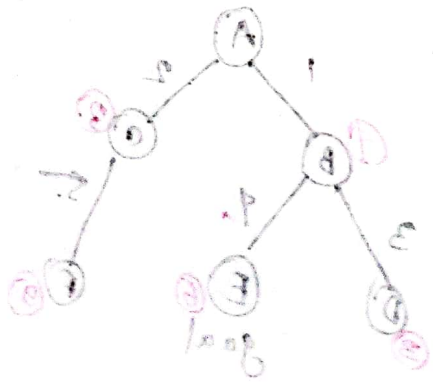
next tresold = 6 ✗

Stat at A → 6 < 7 ✗

next threshold value = 7

Stat at A → 7 = 7

expand A



1. Set certai
2. If f(node) >
3. Set thresh
4. Terminate

Step 1

threshold valu

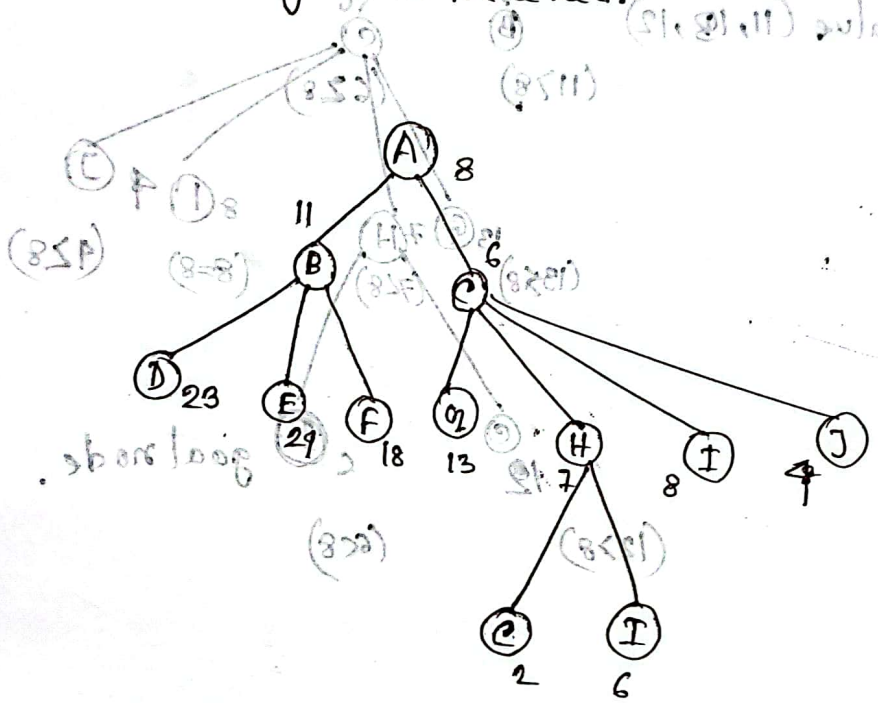
f-value (8)

Step - 2

threshold

f-values (

1. Set certain threshold.
2. If f(node) > threshold ; prune the node.
3. Set threshold = minimum cost of any node that is pruned
4. Terminate when goal is reached.



Step 1

threshold value = 6.

f-value (8) → (A) 8 (8>6)

Step - 2

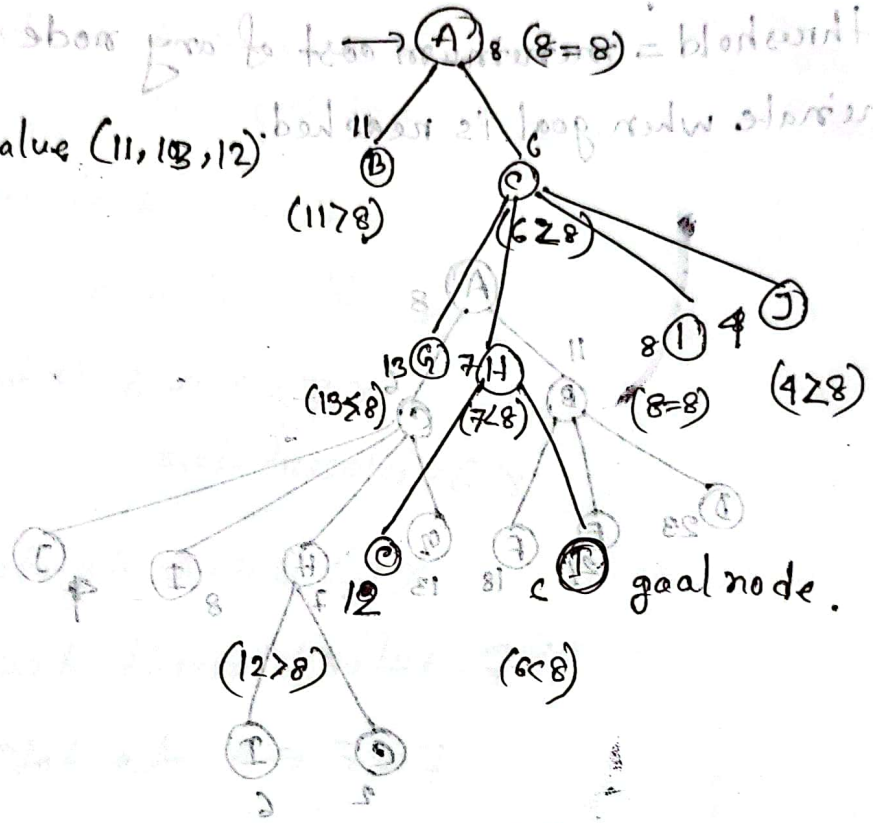
threshold value = 7

f-values (8). → (A) 8 (8>7)

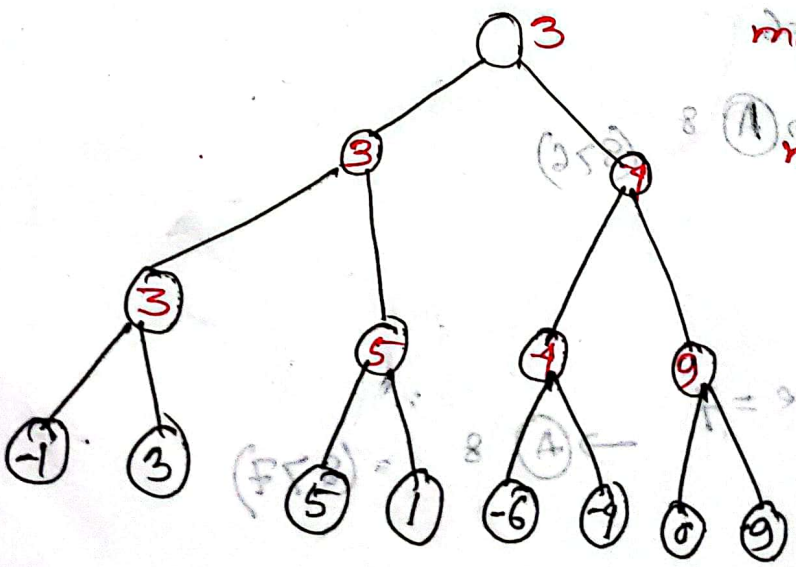
Step 3

threshold value = 8

f-value (11, 13, 12)



Minmax and alph beta search



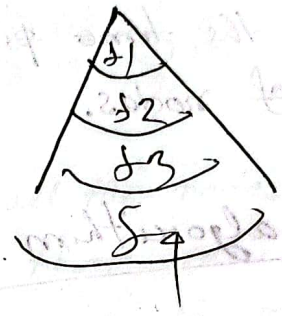
IDA*

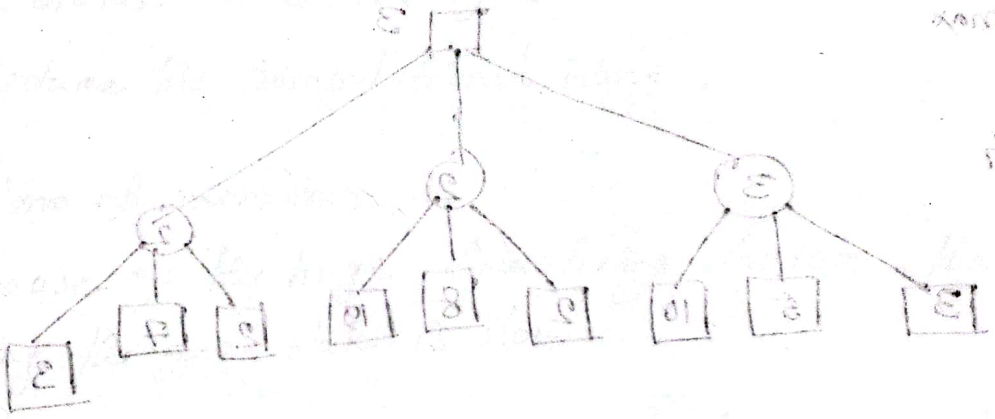
~~perform~~ ~~ADS~~

perform depth first search limited to some f-bound.

if goal found: ok

else: increase f-bound and restart

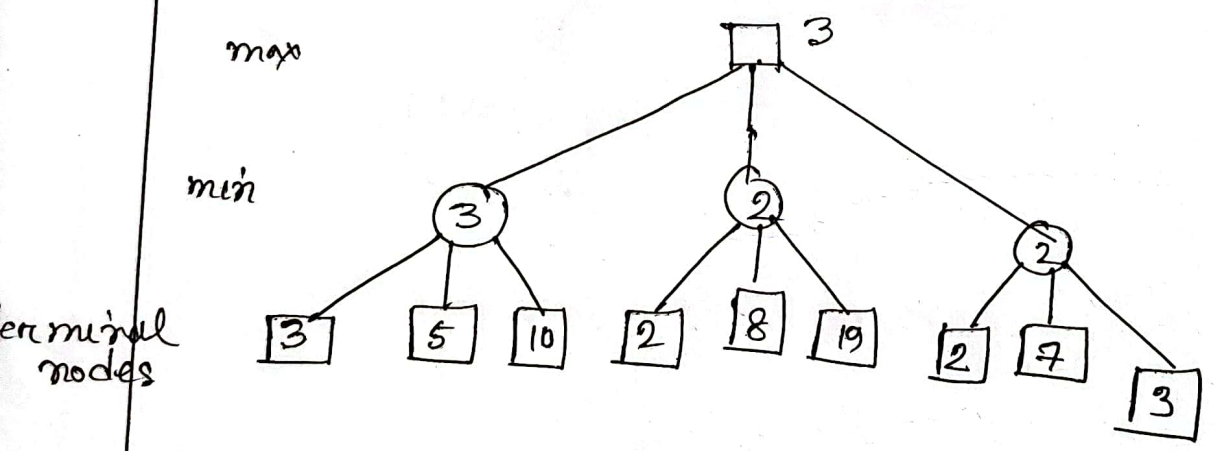




20B

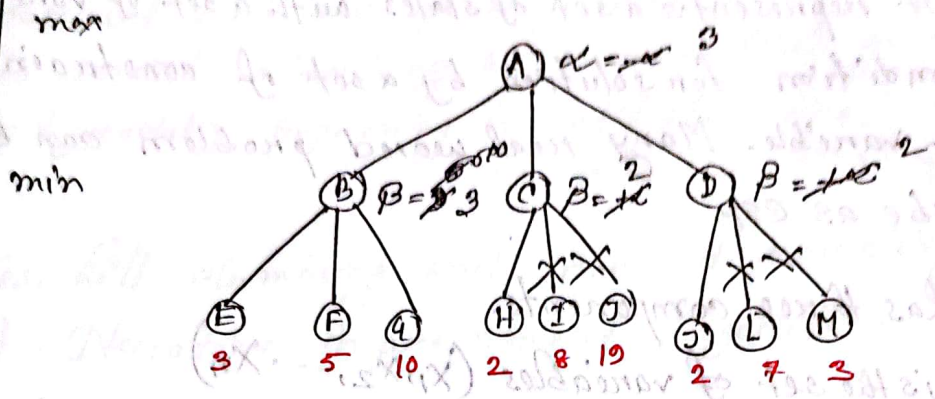
3(c) The alpha-beta algorithm does the same calculation as minimax but it is more efficient than minimax. Because it prunes. It means it allow us to ignore portions of the search tree that make no different to the final state. It makes the algorithm faster, It reeduces the ~~time~~ pi computation time. Decrease the number of nodes.

In minimax algorithm



Here we visisit all the terminal nodes to check the minimum value. Here we need to check every nodes. So, time complexity is more

In alpha beta prunes algorithm:



Here we need no check all the nodes. We can ignore portions of the search tree that make no difference to the final states. Here we igne node I, J and L, M.

It reduce the computational time.

Limitions of minimax:

① Because of the huge branching factor, the process of reching the goal state is slower.

② It has huge branching factor, and the player has lots of choices to decide.

2019
3(d)

CSP: It is a constraint satisfaction problem.
- It represents a set of states with a set of variables, the condition for solution by a set of constraints. on the variable. Many real word problem can be describe as CSP.

A CSP has three components.

① X is the set of variables $(X_1, X_2, \ldots X_n)$

② D is the set of domains $(D_1, \ldots D_n)$ for each variable

③ C is the set of Constraints that specify allowable combination of value.

Yes coloring map is an example of CSP.

Variable: x, y, z, w ← county list

Domains, D_i = {red, green, blue, yellow}

Constraint: adjacent region must have different colors.

eg. $x \neq y$ or (x, y) in {(red, green), (red, blue), (red, yellow), (green, red), (green, blue), (green, yellow), (blue, red), (blue, green), (blue, yellow), (yellow, red), (yellow, green) (yellow, blue)}

→ local search algorithm.

248 3(a). Iterative improvement algorithm: It keep a single current state and tye to improve it. It is a approach to search, starting with an initial guss of a solution and greadully improving it untill it's one.

Yes, hill climbing and simulated annealing is the branch of Iterative improvement algorithm.

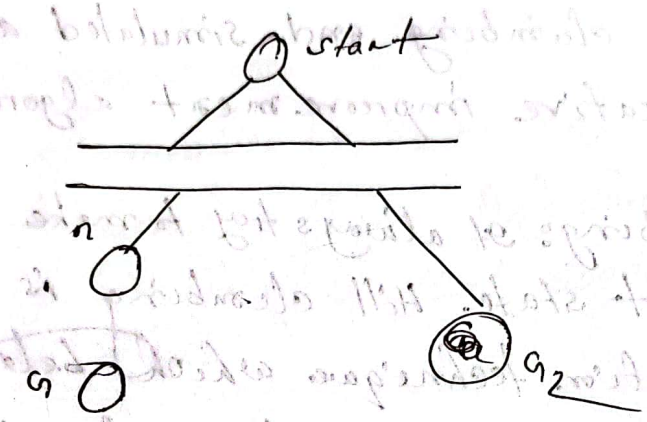
Hill climbing: It always try to make change that improved the current state. Hill climbing is a mathematical optimization technique which belong to local Search algorism. It is a iterative algorithm that start with an arbitary to find solution and then attemp to find better solution to ~~im~~ find optimal solution.

Global optimization technique

Simulated annealing: Sometimes make changes that make things worst, at least temporarily. It is a variant of hill climbing algorithm. ~~Where~~ Hill climbing only allow upword moving, it it allow both upword and downword moving (সরাসরি current state না পেলে back করে অন্য path use করবে).

2017 3(a) prove optimality of A^*

Suppose there is a suboptimal solution G_2 being added to the queue. let n be a node on a path to the optimal solution G.



Suppose suboptimal goal G_2 in the queue.
let n is an unexpected node on a shortest path to optimal goal G.

$f(G_2) = g(G_2)$ Since $h(G_2) = 0$

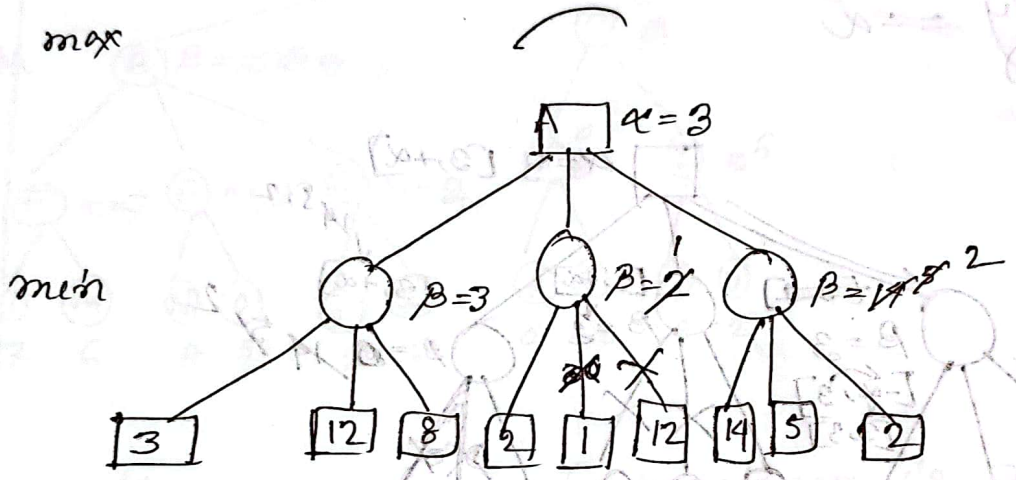
$> g(G)$ Since $h(G_2)$ is suboptimal,

$\geq f(n)$ Since h is admissible

Since $f(G_2) > f(n)$ A^* will never select G_2 for expansion.

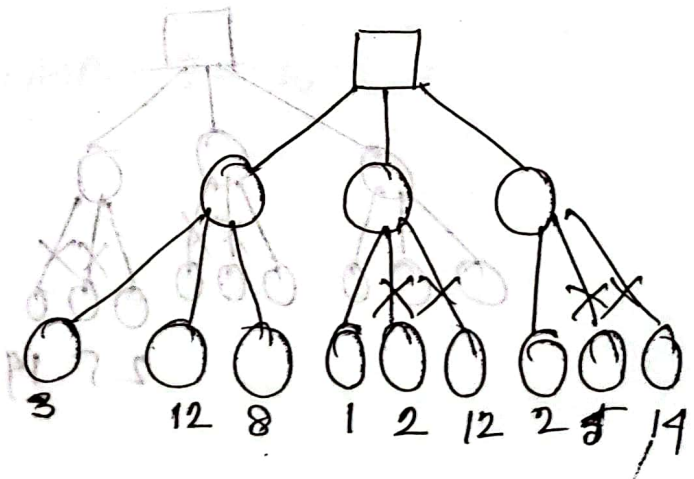
2017

3 (d) Alpha Beta pruning: Alpha Beta pruning is a modified version of minimax algorithm. It works same as the minimax algorithm but it allows to ignore some portion of the tree that can't possibly influence the final result.



Improve effectiveness

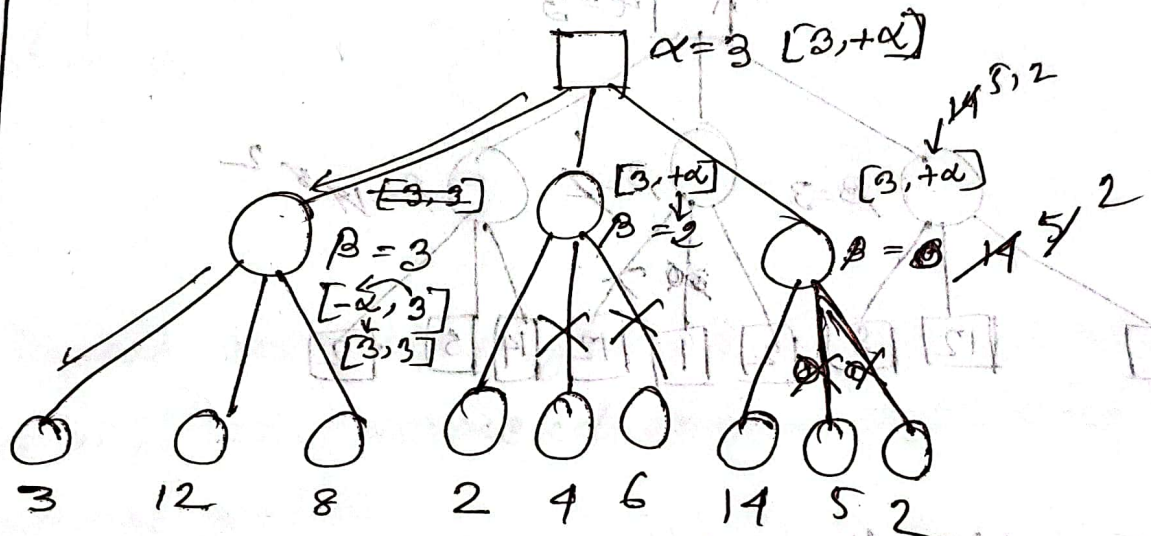
① Highly depend on the order.



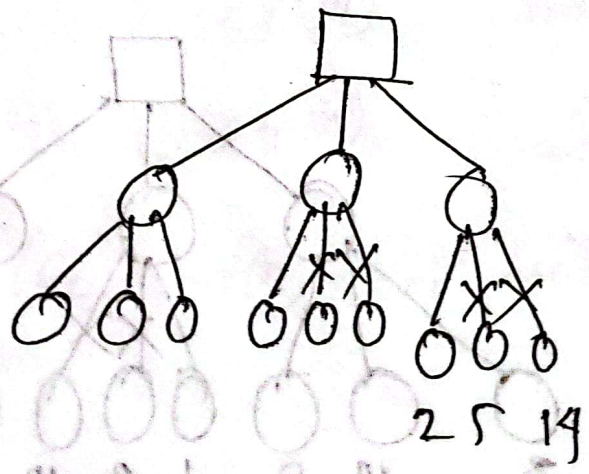
Improved α, β pruning

β = Best lowest value.
Initially = +∞

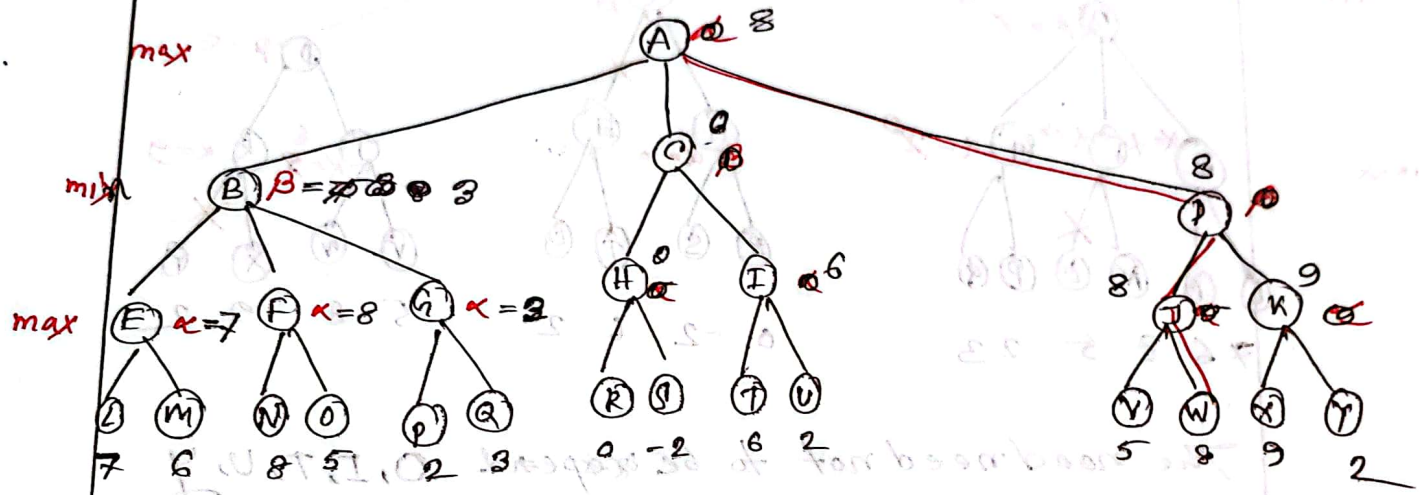
α = Best highest value
Initially = −∞



Improve



2016
4(e) Here, first player is the maximizing player. So, Root node will be considered as max. Next level is min. L2 is max and L3 is min so on.



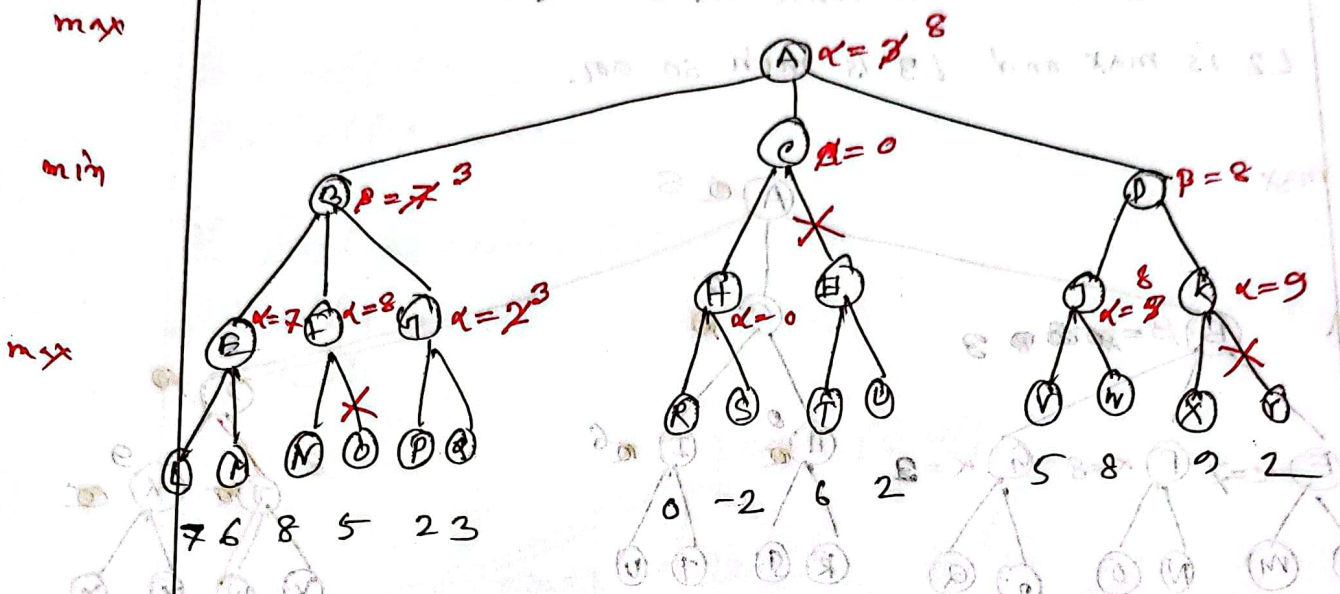
So the move chossen by max player is 8

there path → A → D → J → W.

So, if the 1st Player is the maximizing player, the move move should be choosen is 8

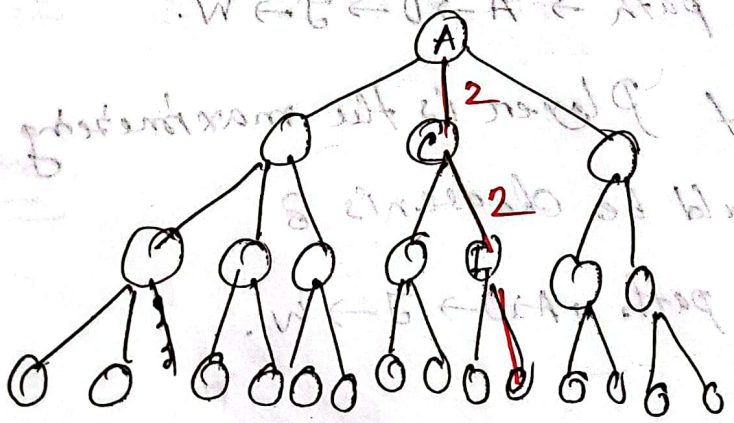
and path, A → D → J → W.

(ii) Nood need not be examined using α-β pruning procedure are:



The nood need not to be expend O, I, T, U, J.

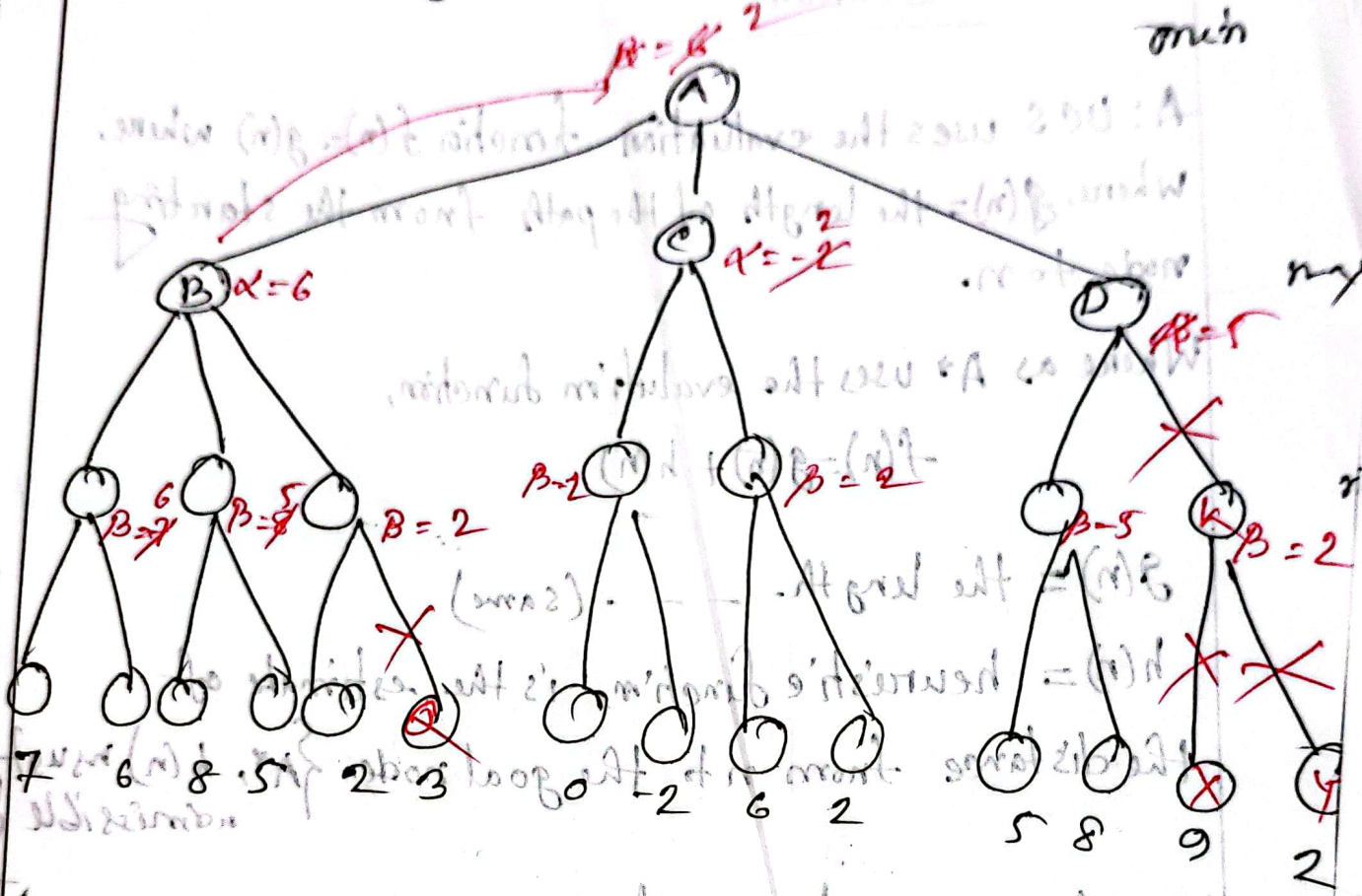
Extra If the 1st player is meneimizer



the chose should be = 2

path A → C → I → U

α-β pruning (examined node, no need node)



node = Q, K, X, Y

2016

26) Uniform cost search special case of A* search.

A: UCS uses the evaluation function $f(n) = g(n)$ where, where, $g(n)$ = the length of the path from the starting node to n.

Where as A* uses the evaluation function,

$$f(n) = g(n) + h(n)$$

$g(n)$ = the length. - - - (same)

$h(n)$ = heuristic function, is the estimate of the distance from n to the goal node. {A*, $h(n)$ must admissible}

Use is a special case of A* which corresponds to having $h(n) = 0$, $\forall n$. A heuristic function h which has $h(n) = 0$, $\forall n$ is clearly admissible because it always "underestimate" the distance to the goal, which can not be smaller than 0. So, indeed UCS is a special case of A*, and its heuristic function is even admissible.

Futility cut off: Terminating the exploration of sub tree that ~~combines~~ offers little probability for improvement over other known path is called ~ futility of cut off.

Waiting for Quiescence: Continue the search until no drastic change occurs from one level to the next.

2019

4(a) Briefly explain why FOL considered as the generalization of PL.

Propositional logic	First order logic (FOL) (predicate logic)
① We use proposition in which complete sentence is denoted by symbol.	FOL used predicates which involve constants, variable, function and relation.

In propositional logic, we use letter to symbolized entire propositions. Propositions are the statement of the form "x is y" where x is a subject and y is a predicate. In prediction logic we symbolize subject and predicate seperately man (Socrates). proposition logic dest work with for statement like All, everyone, someone. So, FOL is the advance version of Propositional logic.

The limitation of PL is that it doesn't represent any individual entities whereas FOL can easily represent the individual entities.

Propositional logic	FOL
It can not represent small worlds like vacuum cleaner world	It can represent very well small world problem.
It can't represent properties of individual entities or relation between entities i.e. Hiral is short	FOL can directly represent properties of individual entities or relation between entities. A: short (Hiral)

proposition logic is the simplest form of logic where all the statements are made by propositions. which is either true or false. It is a technique of representation of knowledge in logical or mathematical form.

Example:
a) It is sun day
b) The sun is rises from west (False)
c) 3+3 = 7 (False proposition)
d) Si's a prime

FOL is more powerful than PL
↓
It has variables and quantifiers, that express the sentences in much more natural way which is not possible in PL.

2019

4(b)(i) If a perfect sq is divisiable by prime p, then it is also divisible by square of p.

(ii) Every perfect square is divisible by some prime.

(iii) 36 is a perfect square.

(iv): Does there exist a prime q such that square of q is divides by 36.

Ans

(i) $\forall x, y$ perfect_square$(x) \wedge$ prime$(y) \wedge$ divisible(x, y)
$\rightarrow$ divisible$(x,$ square$(y))$. —— (1)

(ii) $\forall x \exists y$ perfect-squar$(x) \wedge$ prime$(y) \wedge$ divible(x, y) —— (2)

(iii) perfect-square(36).

(iv) $\exists y$ prime$(y) \wedge$ divide$(36,$ square$(y))$

2019
7(b)

(i) If the perfect square is divisible by a prime p, then it is also divisible by square of p.

$\forall x,y$ (perfect_square(x) $\wedge$ prime(y) $\wedge$ divisible(x,y) $\rightarrow$ divisible(x, square(y)). —— ①

(ii) Every perfect square is divisible by some prime.

$\forall x \exists y$ (perfect_square(x) $\wedge$ prime(y) $\wedge$ divisible(x,y)) —— ②

(iii) 36 is a perfect square.

perfect_square(36). —— ③

(iv) Does there exist a prime q such that square of q divide 36.

$\exists y$ (prime(y) $\wedge$ divide(36, square(y)). —— ④

Eliminate bidirectional,

$\alpha \Leftrightarrow \beta = (\alpha \to \beta) \wedge (\beta \to \alpha)$

$(\alpha \to \beta) \Rightarrow \neg\alpha \vee \beta$

Drop universal quantifier.

$\forall x\, Person(x)$ become $Person(x)$ [nothing change]

$(a \wedge B) \vee C = (a \vee C) \wedge (B \vee C)$

$\neg(a \vee b) = \neg a \wedge \neg b$

$\neg(a \wedge b) = \neg a \vee \neg b$

$\neg(\forall x P) = \exists x \neg P$

$\neg(\exists x P) = \forall x \neg P$

$\neg\neg a = a$

(ii) Convert into clausal form,

$\neg perfect_square(x) \vee \neg prime(y) \vee \neg divisible(x, y) \vee divisible(x, square(y))$. — (5)

$\neg perfect_square(36) \vee \neg prime(p) \vee \neg divisible(36, p) \vee divisible(36, square(p))$

$\neg perfect_square(36) \vee \neg prime(p) \vee \neg divisible(36, p)$ — (6)

$perfect\ square(36)$ — (7)

$\neg prime(p) \vee \neg divisible(36, square(p))$ — (8)

(iii) Using resulation, we need to solve,

$\neg prime(p) \vee \neg divisible(36, square(p))$.

Let's disprove.

$prime(p) \vee divisible(36, square(p))$.

prime(p) ∨ divisible(36, square(p)).

¬perfect_square(36).

∨ ¬prime(p) ∨ ¬ divisible(36,

∨ divisible(36, square(p)).

¬ divisible(36, p) ∨ divisible(36, square(p))

planning

2019
B1(a) planning: The planning in Artificial Intelligence is about the decision making task performed by robots or computer program (an intelligent agent) to achieve a specific goal. For executing the plan, we have sequences of actions. Where action are represent as logical description of precondition and effects.

Factors for formulating planning problem,

i) Representation of states.
ii) Representation of goals.
iii) Representation of actions
- Precondition
- Effects.

i) Representation of states: planners decompose the agent world into logical conditions, and represent a state as a conjunction of positive literals. Literals are ground and function fr

ii) Representation of goals: A goal is a partially specified state, represent as a conjunction of positive ground literals

1. Rich ^ Famous 2. At (P2, Delhi).

A propositional state s, satisfies a goal g, if s contains all the atoms in g.

The propositional state Rich ^ Famous ^ Happy satisfies the goal state Rich ^ Famous.

(iii) Representation of actions: To perform action we need to precontions and effects.

- precondition: How environment should be to perform the actions.

- Effect: How environment will be after performing the's action.

Example: An action for flying a plan from one location to another is.

Action : Fly (P, from, to)

Precond : At (P, from) ∧ plane (P) ∧ Airport (from) ∧ Airport (to)

Effect : At (P, from) ∧ A(P, to).

Extra Action performed:

① UNSTACK (A, B)

- Remove block A from B.
- The arm must be empty.
- Block A must have no block on the the top of it.

② STACK (A, B):

- Put Block A on Block B.
- The arm must already be holding A.
- The surface of B must be clear.

③ PICKUP (A):

- Pickup block A from the table.
- The arm must be empty and there must be nothing on the top of A.

④ PUSHDOWN (A):

- put block A on the table.
- The arm must be holding block A.

Some predicates:

ON (A, B) → Block A on the Block B.

ONTABLE (A) → Block A is on the table.

CLEAR (A) → There is nothing on the top of Block A.

HOLDING (A) → The arm is holding block A.

ARMEMPTY → The arm is holding nothing

2016
5(b) Write down the precondition for the following operations:

(i) UNSTACK (A, B): It means that remove block A from block B. The preconditions:

ON (A, B) ∧ CLEAR (A) ∧ ARMEMPTY.

(ii) STACK (A, B): It means that put block A on the block B. For this the precondition are.

precondition: CLEAR (B) ∧ HOLDING (A)

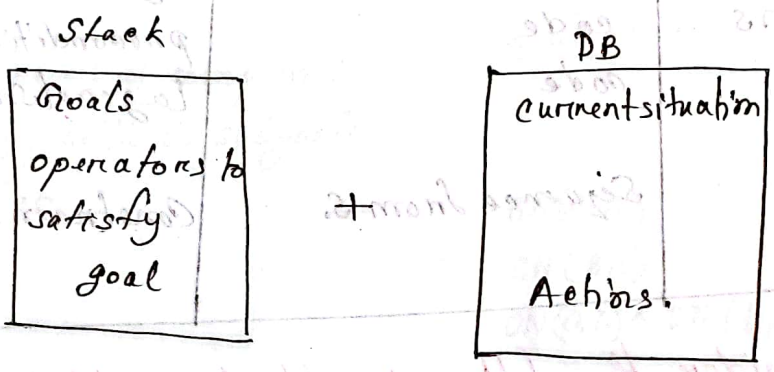
(iii) PICKUP (A): It means that pickup Block A from the table.

precondition: CLEAR (A) ∧ ARMEMPTY.

(iv) PUSHDOWN (A): put Block A on the table.

precondition: HOLDING (A)

Goal stack planning: For solving compound goals we use the technique as goal stack planning. In this method, the problem makes use of a single stack that contains both goals and operators that have proposed to satisfy these goals.



2018 7(b) Explain how planning differ from classical search problem.

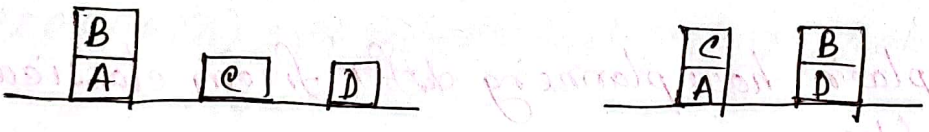
planning is not just the use of a search algorithm (e.g. state-space search algorithm such as A*) to find the plan but planning also involves the modeling of a problem by describing it in some ways. eg. with action language. planning is the process of computing several steps of a problem solving procedure before executing any of them.

The main difference between search and planning is the representation of states.

In search, states are presented as a single entity. whereas in planning states have structured representation that are used by planning algorithm.

	Search	Planning
States	data structures	logical sentences.
Actions	code	precondition/outcomes.
Goal	code	Logical sentence (conjuncts)
plan	Sequence from S.	Constraints on actions.

(c) Consider the following block world problem.



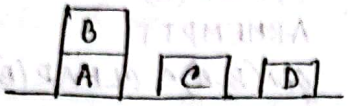
Using Goal stack planning

Initial state: ON(B,A) ∧ ONTABLE(A) ∧ ONTABLE(C) ∧ ONTABLE(D) ∧ CLEAR(B) ∧ CLEAR(C) ∧ CLEAR(D) ∧ ARMEMPTY.

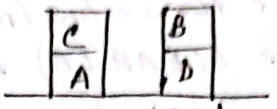
Goal state: ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D) ∧ CLEAR(C) ∧ CREAR(B) ∧ ARMEMPTY.

The following goal are already true in the initial state:
ONTABLE(A) ∧ ONTABLE(D) ∧ CLEAR(C) ∧ CLEAR(B) ∧ ARMEMPTY.

The only goal to solve are ON (C, A) and ON (B, D).



Initial state



Goal state.

push the original compound goal along with its subgoals on the stack.

Goal stack

ON(C, A)
ON (B, D)
ON(C, A) ∧ ON (B, D)

To solve ON (C, A), we can apply STACK (C, A) action, so replace it.

Goal stack

~~ON(C, A)~~ STACK (C, A).
ON (B, D)
ON(C, A) ∧ ON (B, D)

STACK (C, A) can be applied if its precondition are true, so we add precondition on the stack.

Goal stack

CLEAR (A)
HOLDING (C)
CLEAR (A) ∧ HOLDING (C).
STACK (C, A)
ON (B, D)
ON (C, A) ∧ ON (B, D)

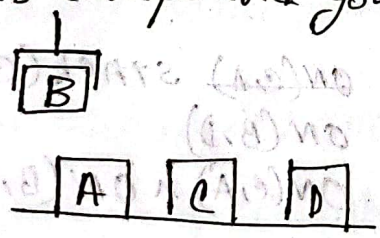
The top of the stack contains CLEAR(A) which is not true.

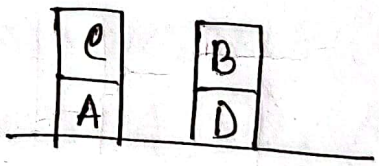
So, make the status true we replace CLEAR (A) by UNSTACK (B, A) and write the precondition of UNSTACK Action.

Goal stack

ON (B, A)
CLEAR (B)
ARMEMPTY
ON(B,A) ^ CLEAR(B) ^ ARMEMPTY
UNSTACK (B, A)
~~CLEAR(C)~~
HOLDING (C)
CLEAR (A) ^ HOLDING(C)
STACK(C, A)
ON (B, D)
ON(B, A) ^ ON (B, D)

ON (B, A), CLERL (B). and ARMEMPTY are all true, So, pop these along with its compound goal.





Now goal stack contains

QUEUE = USTACK (B, A)

Goal stack

HOLDING (C)
CLEAR (A) ^ HOLDING (C)
STACK(C, A)
ON(B, D)
ON (C, A) ^ ON (B, D).

Now, the top of the stack contains HOLDING (C), which is not true.

To make the statement true we use the operator stack (B,D) along with its precontions on the stack.

All the pre condition are true for STACK (B,D). and add it to the queu of a sequence of Actions.

QUEUE = UNSTACK (B,A), STACK (B,D)

The top of the stack HOLDING(C) is not True, to make the state true we use the action PICKUP (C) and write the precondition on the stack.

HOLDING (B)
CLEAR (D)
CLEAR(D) ∧ HOLDING (B)
STACK (B,D)
HOLDING (C)
CLEAR (A) ∧ HOLDING (C)
STACK (C, A)
ON (B,D)
ON (C,A) ∧ ON (B,D)

HOLDING (C)
CLEAR (A) ∧ HOLDING (C)
STACK (C,A)
ON (B,D)
ON (C,A) ∧ ON (B,D).

Goal state

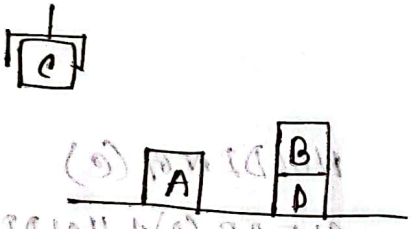
~~CLEAR(C)~~
~~ARMEMPTY~~
~~CLEAR(C) ∧ ARMEMPTY~~
~~PICKUP (C)~~
HOLDING (C) ∧ CLEAR (A)
STACK (C,A)
ON (B,D)
ON (C,A) ∧ ON (B,D)

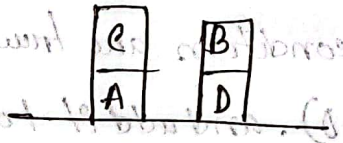
Here, all the precondition of PICKUP operator are true, so add PICKUP(C) to the Queue of sequence action.

HOLDING(C)
CLEAR(C) ∧ HOLDING(C)
STACK(C,A)
ON(B,D)
ON(C,A) ∧ ON(B,D).

Queue = UNSTACK(B,A), STACK(B,D), PICKUP(C).

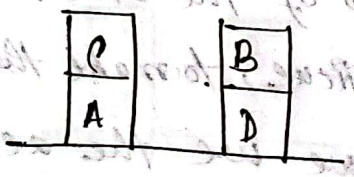
Now the goal stack contain.





Here all the precondition of STACK(C,A) is true, so add STACK(C,A) in the Queue.

Queue: UNSTACK(B,A), STACK(B,D), PICKUP(C) STACK(C,A)



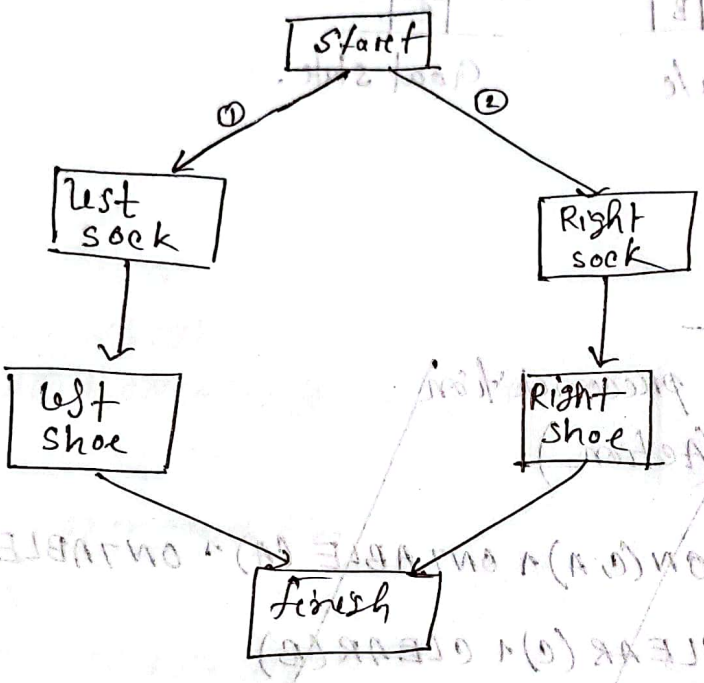
Goal state reached.

1. UNSTACK(B,A)
2. STACK(B,D)
3. PICKUP(C)
4. ~~STACK(C,A)~~ pushdown(C)

Using four action we reach the Goal state

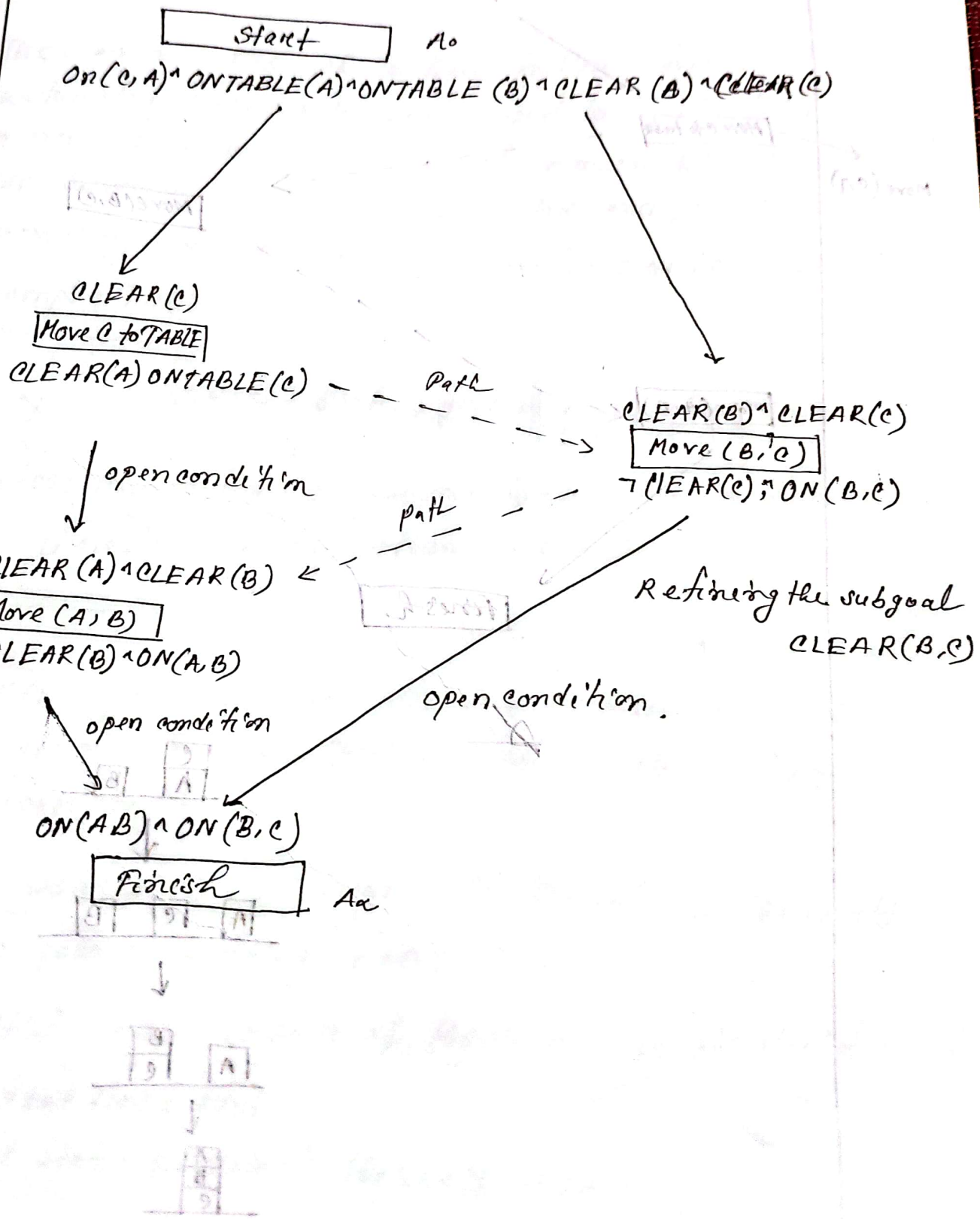
~~Feb 2020~~

6(c) partial order planning: Partial order planning doesn't specify which action will come first out of two actions which are in plan.

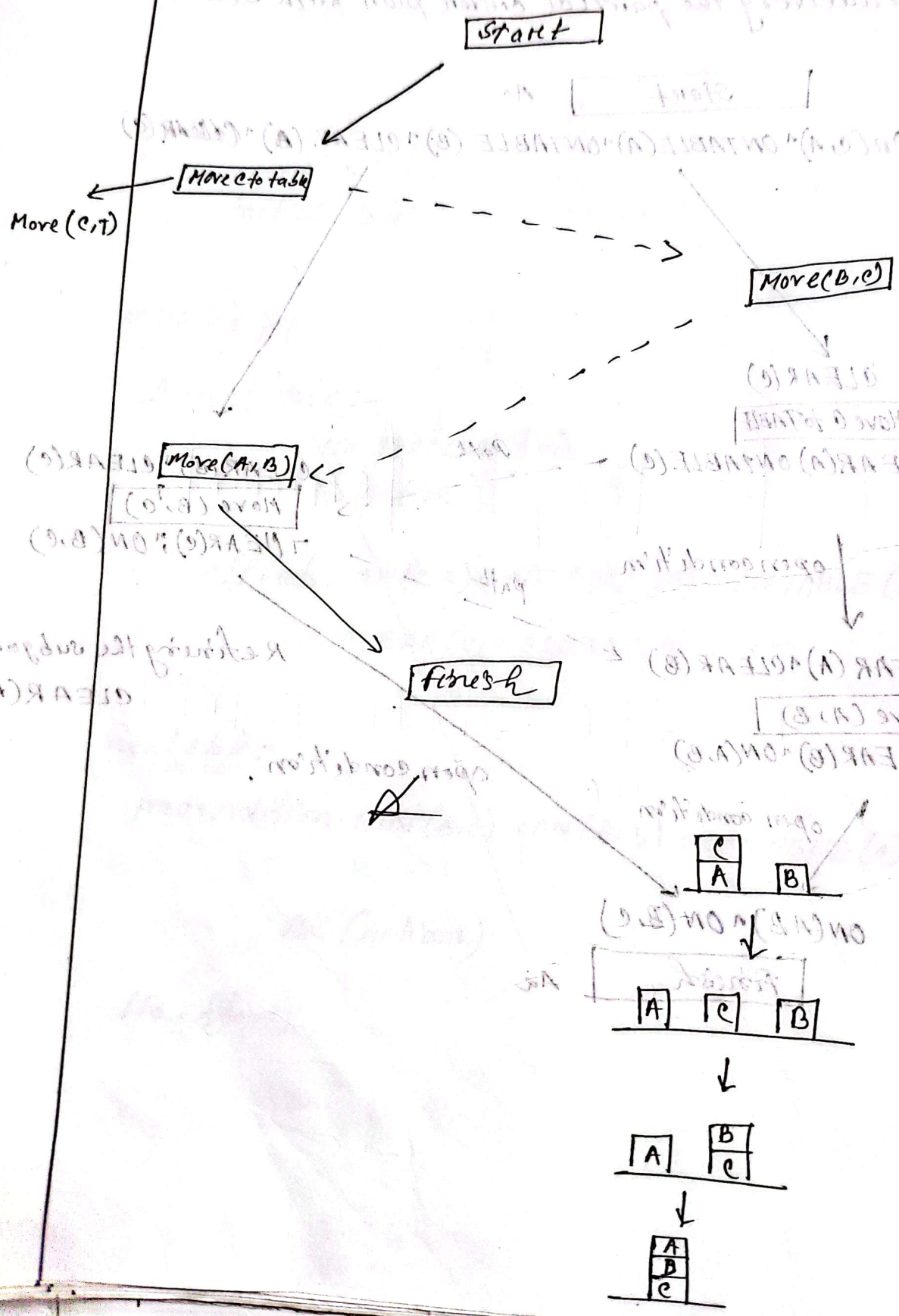


using pop (partial order planning).

* Initializing the partial order plan with start and finish



A totally ordered plan:



2020
6(a) Describe the strips language for representing states, goals and operators within a planning system.

The standford Research Institute Problem solver (STRIPS) is an automated planning technique used to find goal, by executing a domain from initial state of domain. With STRIPS we can describe the world (Initial state and goal state) by providing objects, actions, preconditions and effects.
Example: PDDL.

STRIPS – States, goal and operators/Actions.

States: Conjunctions of ground, function free and positive literals, such as At(Home) ∧ Have (Banana).

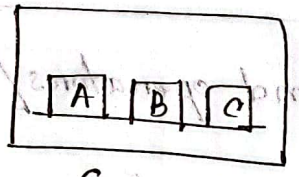
STRIPS representation planning problems as a set of state description and operators expressed in first-order-predicate logic.

State description: It represent the state of the world at three points during plan:

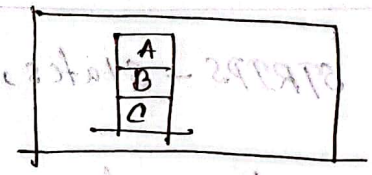
① Initial state: The state of the world at the start of the problem.
② Current state: and
③ Goal state: The state of the world we want to go.

Operators: Operators are actions that can be applied to change the state of the world.

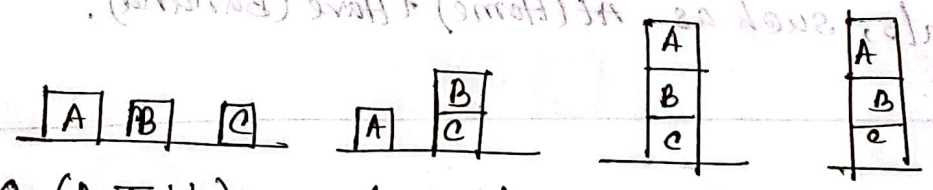
- Each operators has outcome. (How it effect the world).
- Each operator can only be applied in certain circumstances. These are the pre conditions of operator.



S_0



g



On(A, Table)	On(A, table)	On(A, B)	On(B, A)
On(B, Table)	on(B, C)	on(B, C)	On(B, C)
On(C, Table)	clean(A)	On(C, Table)	
clean(A)	clean(B)	clean(A)	
clean(B)	clean(Table)	clean(Table)	
clean(C)			

$$\boxed{S_0} \xrightarrow{\text{move}(B, \text{Table}, C)} \boxed{S_1} \xrightarrow{\text{move}(A, \text{Table}, B)} \boxed{S_2} \longrightarrow g$$

2020

8(b) Bayes theorem: It determine the probability of an event with uncertain knowledge. It allow updating probability prediction of an event by observing new information of the real world. It derived from the chain rule:

$$P(A,B) = P(A|B)P(B) = P(B|A)P(A)$$

with this powerful rule, the Bayes theorem constructed,

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Here, $P(B/A)$ is probability of evidence given in the hypothesis.

$P(A)$ is the Prior propability of hypothesis.

$P(B)$ is the Prior Propability of evidence.

$P(A|B)$ posterior

$$P(\text{Cause}|\text{Effect}) = P(\text{Effect}|\text{Clause})\,P(\text{clause})/P(\text{Effect}).$$

Here given,

$P(\overset{S}{\text{Stiff Neck}} / \overset{M}{\text{Meningitis}}) = 0.4$

$P(\overset{M}{\text{Meningitis}}) = 1/50000$

$P(\overset{S}{\text{Stiff Neck}}) = 1/25$

$P(\underset{M}{\text{Meningitis}} / \underset{S}{\text{Stiff Neck}}) = ?$

We know;

$$P(M/S) = \frac{P(S/M) \cdot P(M)}{P(S)}$$

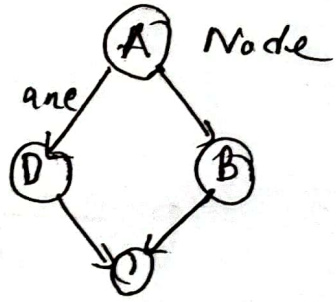
$$= \frac{0.4 \times 0.00002}{0.04}$$

$$= 0.0002$$ Ans.

8(a) Bayesian network: It is a probabilistic (It uses distributed probability) graphical model which represent a set of variables and their conditional dependencies using a directed acycle graph.

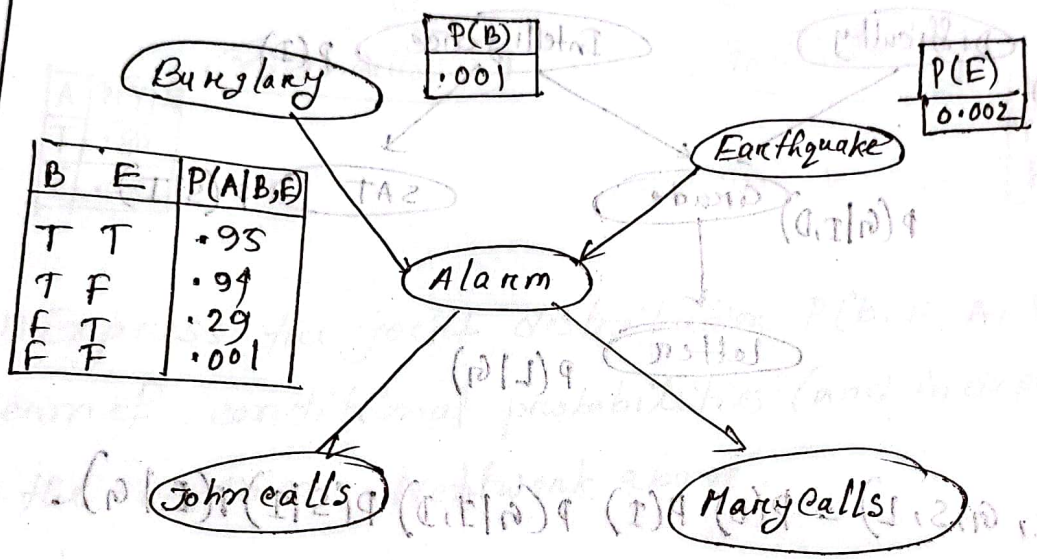
It consist of 2 parts,

1. Directed acycle graph.
2. Table of conditional probability



Syntax of Bayesian network:

- A directed acyclic graph whose node represent random variable $X = (X_1, X_2, X_3 \cdots X_n)$.
- For each node x_i, a conditional propability distribution $P(x_i \mid Parents(x_i))$ given. Example:



P(B)
.001

P(E)
0.002

B	E	P(A|B,E)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

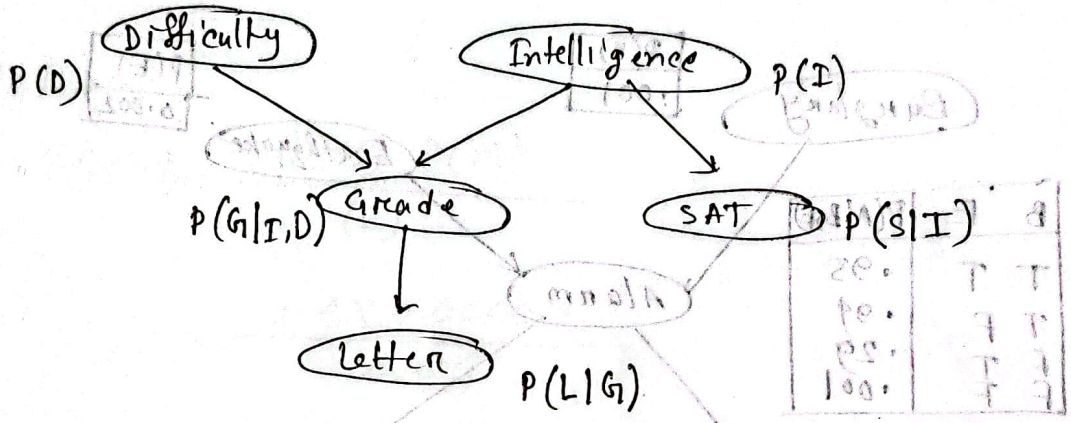
A	P(J|A)
T	.90
F	.05

A	P(M|A)
T	0.70
F	0.01

Semantics: A bayesian network represents a joint distribution.

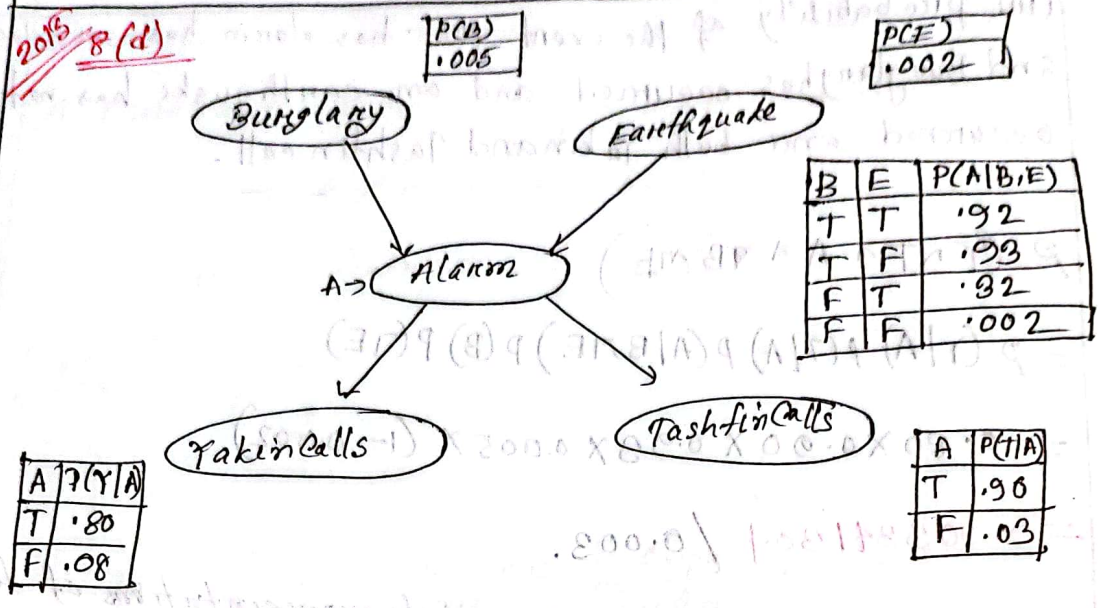
$$P(x_1, x_2 \cdots x_n) = \prod P(x_i \mid parents(x_i))$$

Example:



$$P(D, I, G, S, L) = P(D)\, P(I)\, P(G \mid I, D)\, P(S \mid I)\, P(L \mid G).$$

2015 8 (d)



B	E	P(A|B,E)
T	T	.92
T	F	.93
F	T	.32
F	F	.002

A	P(Y|A)
T	.80
F	.08

A	P(T|A)
T	.90
F	.03

(I) Express the joint distribution $P(B, E, A, Y, T)$ in the term of conditional probabilities (and independence) express in the bayesian Network above..

$$P(B,E,A,Y,T) = P(B)\,P(E)\,P(A|B,E)\,P(Y|A)\,P(T|A).$$

(II) probability of the event that the alarm has sounded, but neither a Burglary nor an earthquake has occured and both Yakin and Tashfin call.

$$P(Y \wedge T \wedge A \wedge \neg B \wedge \neg E)$$

$$= P(Y|A)\,P(T|A)\,P(A|\neg B, \neg E)\,P(\neg B)\,P(\neg E)$$

$$= 0.80 \times 0.90 \times 0.001 \times (1-0.005) \times (1-0.002)$$

$$= 0.0007149672$$

(iii) probability of the event that has alarm has sounded and Burglary has occurred and an earthquake has not occurred and both Yakin and Tashfin call.

$P(Y \wedge T \wedge A \wedge B \wedge \neg E)$

$= P(Y|A)\, P(T|A)\, P(A|B, \neg E)\, P(B)\, P(\neg E)$

$= 0.80 \times 0.90 \times 0.93 \times 0.005 \times (1 - 0.002)$

$= 0.003341304 / 0.003.$

2019

5(c) knowledge base : A knowledge is a set of representation of what one know about the world. (objects and classes of objects, the facts about objects, relationship among objects etc). Each individual representation is called sentence. The sentences are expressed in knowledge representation language.

It is a set of representations of facts about the world known as sentence.

Example of Sentences:

① If A is true then B is true.

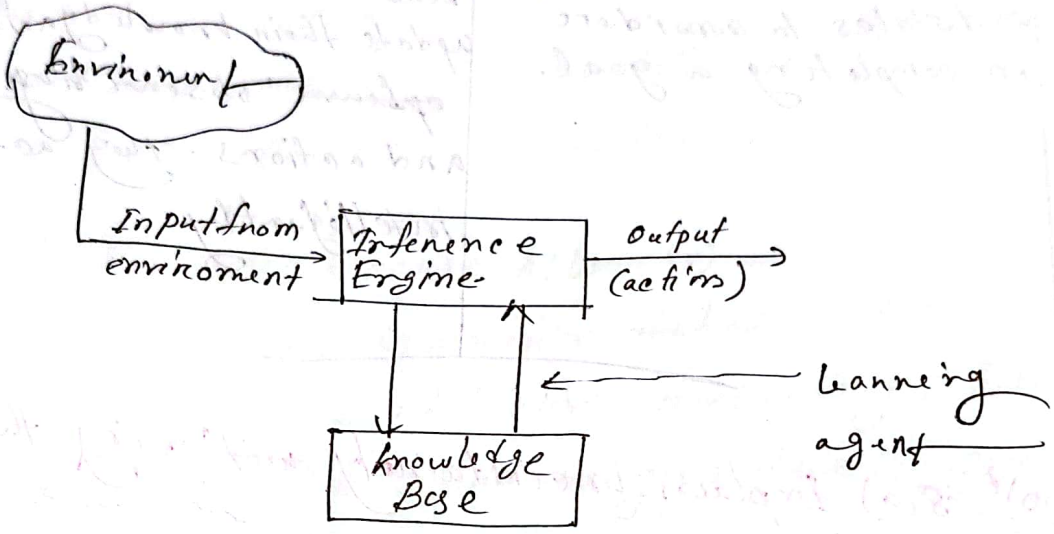
② All humans are mortal.

Draw the architecture of knowledge base agent.

A knowledge based ~~archite~~ agent consist of

① KB / knowledge base;

② Inference engine (IE)



The above diagram is representing a generalized architecture for knowledge based agent. The KBA take input from the environment by percieving the environment. The input is taken by the Inference engine of the agent which ~~will~~ also communicates with KB to decide. The learning agent of KBA regularly updates the KB by learning new knowledge.

(iii) probability of the event that has alarm has sounded and Burglary has occurred and an earthquake has not occurred and both Yakin and Tashfin call.

$P(Y \wedge T \wedge A \wedge B \wedge \neg E)$

$= P(Y|A)\, P(T|A)\, P(A|B, \neg E)\, P(B)\, P(\neg E)$

$= 0.80 \times 0.90 \times 0.93 \times 0.005 \times (1 - 0.002)$

$= 0.003341304 / 0.003.$

2019 5(c) knowledge base: A knowledge is a set of representation of what one know about the world. (objects and classes of objects, the facts about objects, relationship among objects etc). Each individual representation is called sentence. The sentences are expressed in knowledge representation language.

It is a set of representations of facts about the world known as sentence.

Example of Sentences:

① If A is true then B is true.

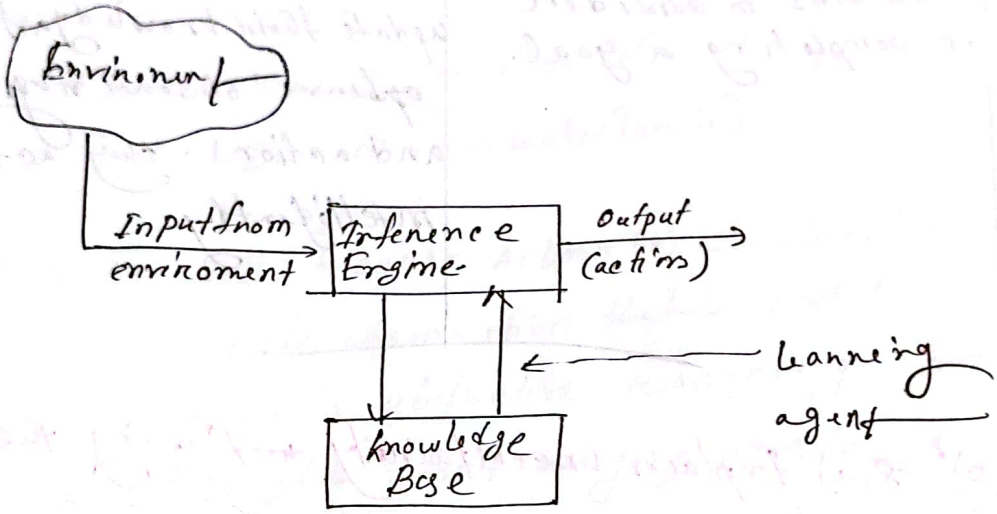
② All humans are mortal.

Draw the architecture of knowledge base agent

A knowledge based ~~archite.~~ agent consist of

① KB / knowledge base;

② Inference engine (IE)



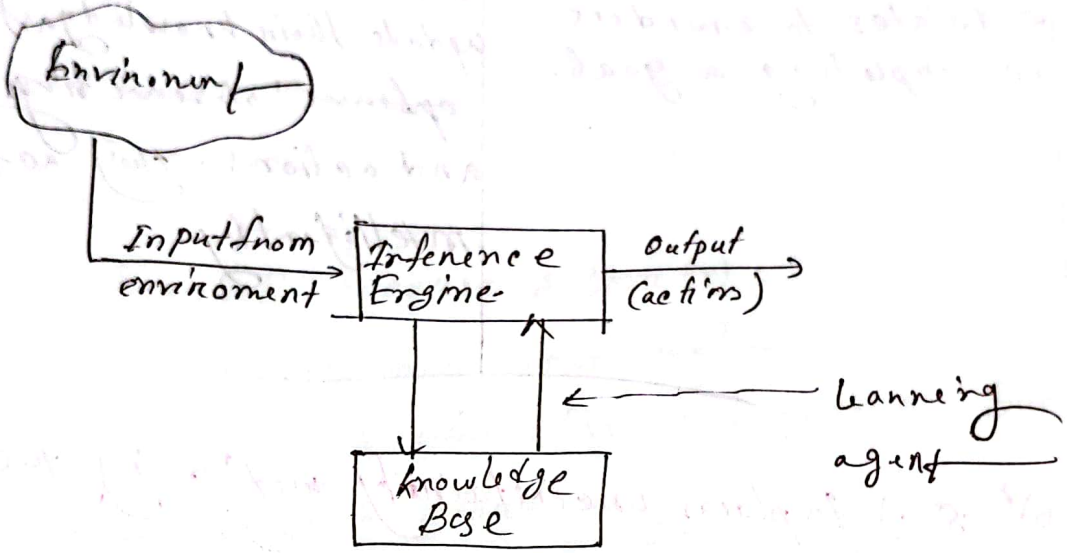
The above diagram is representing a generalized architecture for knowledge based agent. The KBA take input from the environment by percieving the environment. The input is taken by the Inference engine of the agent which ~~will~~ also communicates with KB to decide. The learning agent of KBA regularly updates the KB by learning new knowledge.

Draw the architecture of knowledge base agent

A knowledge based ~~archite~~ agent consist of

① KB / knowledge base ;

② Inference engine (IE)



The above diagram is representing a generalized architecture for knowledge based agent. The KBA take input from the environment by percieving the environment. The input is taken by the Inference engine of the agent which ~~will~~ also communicates with KB to decide. The learning agent of KBA regularly updates the KB by learning new knowledge.

Problem solving agent	knowledge based agent
Solve a specific problem and provide best result. PSA decide which action and states to consider in completing a goal.	who has the capability of maintaining an internal state of knowledge, reason over the knowledge, update their knowledge after ~~observe~~ observe things and actions. They act intelligently

2018 8(a) Explain uncertainty ~~wit~~ using the example,

"The doorbell rang at 12 o'clock in the midnight
Was someone there at the door?
Did Karim wake up?"

Uncertainty is defined as the lack of the exact knowledge that would enable us to reach a perfectly reliable conclusion. In fact, It is a lack of ~~to~~ formate to formulate a decision.

Here the example is,

The doorball rang at 12'o clock in the midnight
- was the someone there at the door?
- Did kareim wake up?

From here we find,

Prop1 : AtDoor (x) $\rightarrow$ Doorbell.

Prop2 : Doorbell $\rightarrow$ wake(kareim).

Given doorbell, because AtDoor (x) $\rightarrow$ Doorbell?
It is incomplete observation ~~that is~~ and prediction may be true. It is abductive reasoning, But, no, the doorbell might start ringing due to some other reason.

Again, Doorbell we can say, wake(kareim)
because Doorbell $\rightarrow$ wake (kareim)

According to deduction reasoning Proposition 2 is always true according to general Rule.

However, in general kareim may not always wake up, even if the bell rings.

Prop 2 is often true, but not a tautology
(Incomplete knowledge)

2018

8(b)

Abductive Reasoning

Incomplete observation → Best prediction (May be true)

① logical reasoning

② The premises do not guarantee the conclusion

Implication: Cricket ground is wet if it is raining

Axiom: Cricket ground is wet

Conclu: It is raining

Deductive Reasoning

General Rule → Specific Conclusion (always True.)

① The truth of the premises guarantees the truth of the conclusion

P1: all human eats vegetable

P2: Suresh is human

Con: Suresh eats vegetable

Inductive/defeasible reasoning

Specific Observation → General Conclusion (may be true).

① Arrive at a conclusion using limited sets of facts by the process of generalization. (bottom up reasoning)

② The truth of premises doesn't garantee the truth of the conclusion.

Reasoning is the process of deriving logical conclusions from given facts.

P: All the pigeons we have seen in the zoo are white.

C: Therefore we can expect all the pigeons to be white.

এবং এর conclusion গুলো real world এ all time true নাও হতে পারে।

Sound resoning: Allways true in the real world.

Heuristics: It is a technique that is used to solve a problem faster than the classical method. It find the approximate solution of problem when classical methods do not. It said to be the problem solving technique that result in practical and quick solution.

Two types of heuristic:

① $h_1(n)$: Number of misplace tiles.

② $h_2(n)$: Total Manhattan distance

7	2	4
5		6
8	3	1

1	2	3
4	5	6
7	8	

$h_1(n) = 6$

$h_2(n) = \cancel{4+0+5+0+1+3+3+}$

$3+1+2+3+1+1+2+2 =$

2017 4(b) Write the different types of inference procedure in BN.

Types of interference.

① ~~Intere~~ Inference by enumeration: Inference by listing on recording all variables.

② Inference by variable elimination: Inference by variable removal.

① I by E : Any conditional probability can be computed by summing terms from the full joint distribution

More specifically, a query $P(X|e)$ can be answered using equation,

$$P(X|e) = \alpha P(X,e) = \alpha \sum_{y} P(X,e,y).$$

where, α is a normalize constant.

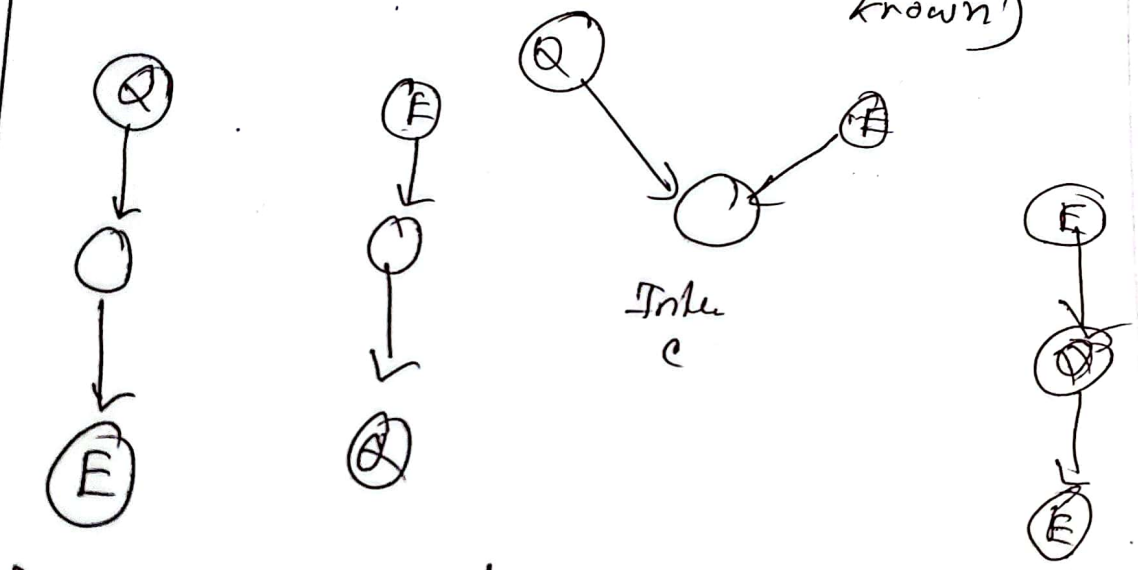
X = Query variable.

e — event

y = number of terms.

Inferencing in BN

1. Diagnostic inference : (from effect to causes)
2. Causal inference (from causes to effects)
3. Intercausal " (between causes of a common event)
4. Mixed inference (Some causes and some effect known)



Diagnostic causal

2020
7(a)

(i) Man (marcus)

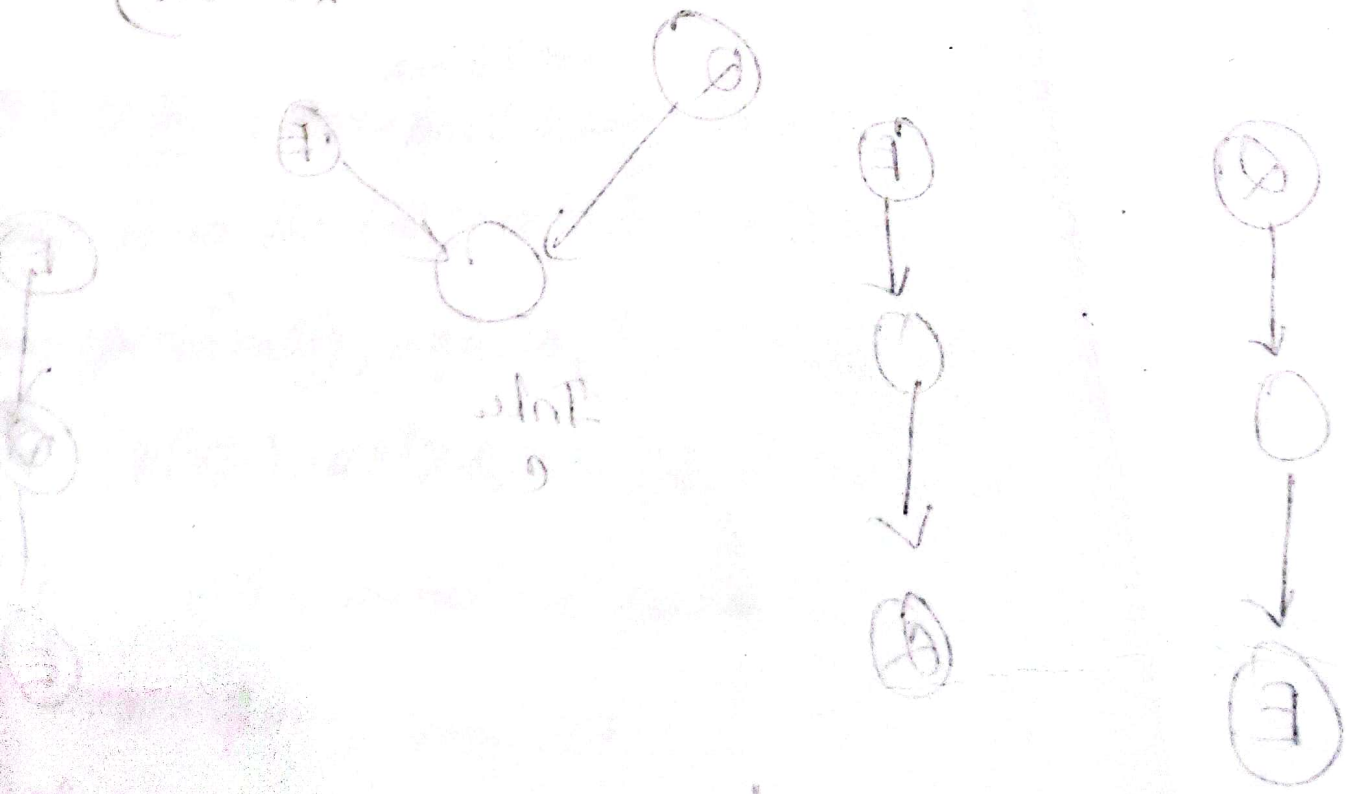
(ii) Pompeian (Marcus)

(iii) Born (Marcus, 40 AD)

(iv) $\forall x (man(x) \rightarrow mortal(x))$

(v) $\forall x (Pompeian(x) \wedge erupted(Volcano, 79) \rightarrow died(x, 79\ AD))$

(vi)



2018
1(a)

P	Q	¬Q	P∨Q	P∨¬Q	(P∨Q)∧(P∨¬Q)	(P∨Q)∧(P∨¬Q)∨P
T	T	F	T	T	T	T
T	F	T	T	T	T	T
F	T	F	T	F	F	F
F	F	T	F	T	F	F

(i) Satisfiable: If a sentence is true for some world and can also false timetimes. So the sentence is satisfiable. (Sometimes true and somtimes false)

(ii) Contradictory: If a sentence is not true any of the world then it is called contradictory. The following sentence is true for some world. So, it is not contradictory.

(iii) Valid: If a sentence is true for all the possible world, then the sentence is valid. It means always true for all the world. But the following sentence is not true for all the possible world. So, it is not valid sentence

$\alpha \leftrightarrow \beta$
$(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$

2020
1(b)

R1: $\neg P_{11}$
R2: $B_{11} \leftrightarrow (P_{21} \vee P_{12})$
R3: $B_{21} \leftrightarrow (P_{11} \vee P_{22} \vee P_{31})$
R4: $\neg B_{11}$
R5: B_{21}

Elimination bidirection from R_2 we obtain R_6

$R_6 : (B_{11} \rightarrow (P_{21} \vee P_{12})) \wedge ((P_{21} \vee P_{12}) \rightarrow B_{11})$

Apply And-elimination to R_6 to obtain,

$R_7 : ((P_{12} \vee P_{21}) \rightarrow B_{11})$

Logical equabalance for contrapositive,

$p \rightarrow q$
$(p \rightarrow q) \wedge (q \rightarrow p)$

$R_8 : (\neg B_{11} \rightarrow \neg (P_{12} \vee P_{21}))$

Apply modus Ponens on R_8 and R_4 ($\neg B_{11}$):

$R_9 : \neg (P_{12} \vee P_{21})$

According to De Morgans law,

$R_{10} : \neg P_{12} \wedge \neg P_{21}$

That is neither [1,2] nor [2,1] contains a pit.

And elimination,

$R_{11} : \neg P_{12}$

OOP 6(a) **Rule-based System:** A System which KB is represented as a set of rules and facts is called a Rule based system.

A rule-based System is a system that applies human made rules to store, sort and manipulate data.

A rule based system consists of a collection of IF-THEN rules. and a collection of facts.

Rules are represented in the following form:

IF <antecedent> THEN <consequent>.

When the antecendent part is NULL then Rule becomes a fact.

~~R1: $\neg P_{1,1}$~~

~~R2: $\neg B_{1,1}$.~~

R_6 : ~~$B_{1,1} \Leftrightarrow$~~ $(B_{1,1} \rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \rightarrow B_{1,1})$ Biconditional elimination.

R_7 : $((P_{1,2} \vee P_{2,1}) \rightarrow B_{1,1})$ and elimination.

R_8 : $(\neg B_{1,1} \rightarrow \neg(P_{1,2} \vee P_{2,1}))$ Equivalence for contrapositive.

R_9 : $\neg(P_{1,2} \vee P_{2,1})$ Modus Ponens with R_8 and R_7

R_{10} : $\neg P_{1,2} \wedge \neg P_{2,1}$ De' mangan's Rule.

R_{11} : $\neg P_{1,2}$ and elimination.

2019 6(a) **Rule-based System:** A System which KB is represented as a set of rules and facts is called a Rule based system.

A rule-based System is a system that applies human made rules to store, sort and manipulate data.

A rule based system consists of a collection of IF-THEN rules. and a collection of facts.

Rules are represented in the following form:

IF <antecedent> THEN <consequent>.

When the antecendent part is NULL then Rule becomes a fact.

~~R1: $\neg P_{1,1}$~~

~~R2: $\neg B_{1,1}$.~~

R_6 : ~~$B_{1,1} \Leftrightarrow$~~ $(B_{1,1} \rightarrow (P_{1,2} \vee P_{2,1})) \wedge ((P_{1,2} \vee P_{2,1}) \rightarrow B_{1,1})$ Biconditional elimination.

R_7 : $((P_{1,2} \vee P_{2,1}) \rightarrow B_{1,1})$ and elimination.

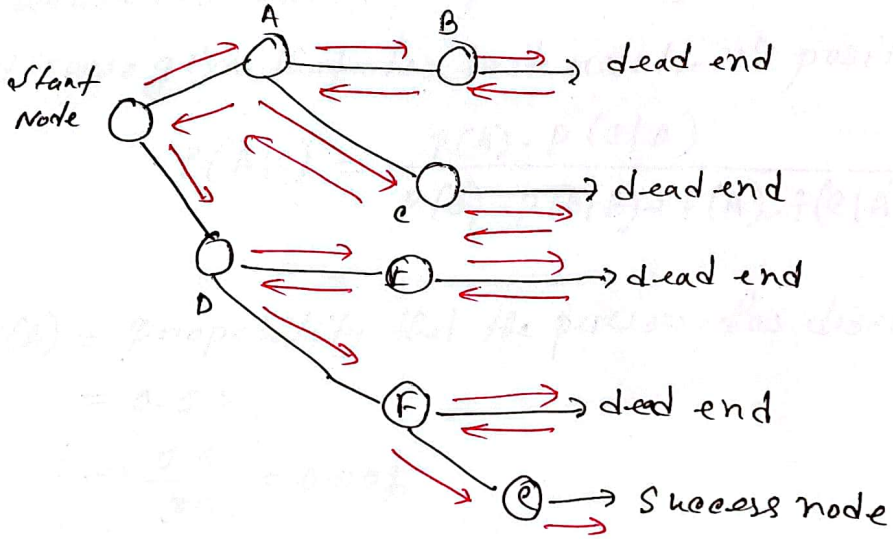
R_8 : $(\neg B_{1,1} \rightarrow \neg(P_{1,2} \vee P_{2,1}))$ Equivalence for contrapositive.

R_9 : $\neg(P_{1,2} \vee P_{2,1})$ Modus Ponens with R_8 and R_7

R_{10} : $\neg P_{1,2} \wedge \neg P_{2,1}$ De' mangan's Rule.

R_{11} : $\neg P_{1,2}$ and elimination.

2019
5(b) **Backtracking:** Backtracking in AI is a technique that is used in an algorithmic way to solve problems in a recursive fashion. It is a systematic method of trying various sequence of decision untill we find out that goal.



2019
7(a)

let,

A: Person has the disease.

B: Person doesn't have the disease.

C: Test result positive.

We need to find the probability that a person has the disease given that his test result is positive.

$$P(A|C) = \frac{P(A) \cdot P(C|A)}{P(B) \cdot P(C|B) + P(A) \cdot P(C|A)}$$

$P(A)$ = Propability that the person has disease,

$= 0.5\%$

$= \frac{0.5}{100} = 0.005$

$P(B)$ = probability that person doesn't have the disease.

$= (1 - 0.005)$

$= 0.995$

$P(C|A)$ = Probability that test result is positive if the person has the disease

$= 95\%$

$= 0.95$

$P(C|B)$ = The result positive if the patient doesn't have the dise

$= 0.001$

Represent causal connection with probabilities

Belief network

$$P(A|c) = \frac{0.005 \times 0.95}{0.995 \times 0.001 + 0.005 \times 0.95}$$

$$=$$

We need extension of bayes theorem where there are multiple hypothesis and one evidence. one there is one hypothesis and multiple evidence.

Regarding to the byas theorm,

$$P(H_i|E) = \frac{P(E|H_i) \cdot P(H_i)}{P(E)}$$

$P(H_i|E)$ = Probability Hypothesis is true given evidence E.

$P(E|H_i)$ that we will alway observe evidence E that H_i is true

$P(H_i)$: H_i is true adsent of any specific evidence

$P(E)$

problem of byas theorm: When we have any evidence, the size of joint probibility we need to compute grows 2^n

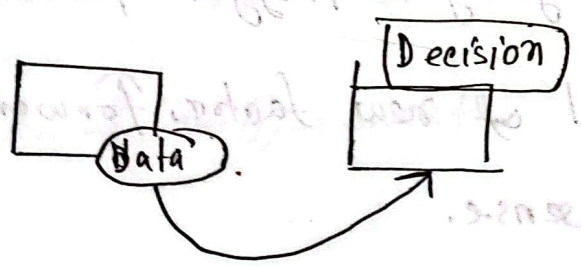
By. Using Bayesian Belief network we overcom the

2020 5(a) **Forward channing:** It is also called data driven. Start from the given facts try to arrive at the goal. In other word, It start from the stat/initial state applying rules on by one to arrive at the goal state. It is the strategy that "what can happen next".

Forward channering

When we have some data and we make a decision based on this data then this process is called as forward channering

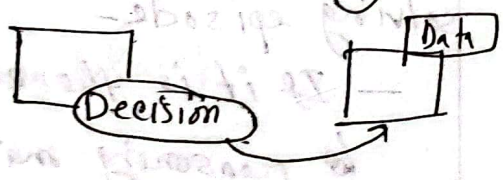
Example: "If it is raining then, we will take umbrella" Here, it is raining is the data and we will take umbrella is a decision. This means that we already knew that it is raining that's why we take umbrella. This is forward channering.



Backward channering

When we have a decision and based on the decision If we fetch the initial data then process is called as backward channering.

Ex: If we make a decision of taking umbrella. Then based on this decision it can be guessed that it is raining. where taking umbrella is a decision and based on which we guessed that the data can be "it is raining". This is backward channering.



Data driven technique	decision driven technique.
Used in design expert system.	Used by the diagnostic expert system.

Selection of forward and backward reasoning depens on which direction offeres less less branching factor.

The following factor are determin the direction

① Are there more possible start states on goal states? We would like to move from the smaller set of states to the larger set of states.

② In which direction is the branching factor greater. (Go the direction of lower branching factor).

③ Will the program justify reasoning.

④ What kind of event is going to trigger a problem solving episode.

— If it is the arrival of new factor, forward reasoning makes sense.

most movement supervised learning.

- If it is a query to which a response is desired, Backward chaing is more natural.

2019

8(a) Learning: Learning or machine learning is an application of artificial intelligence that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. In other world learning is the process of converting experience into expertise or knowledge.

Inductive Learning: Inductive learning also known as discovery learning is a process where the learns discovers rules by observing example. In inductive learning system tries to make a 'general rule' from a set of observation instance.

Example:

Mango → f(Mango) → Sweet (e1)

Banana → f(Banana) → Sweet(e2)

- - -

Fruit → f(Fruit) → Sweet (General Rule)

most inductive learning is supervised learning.

From a set of example an agent tries to create an evaluation function.

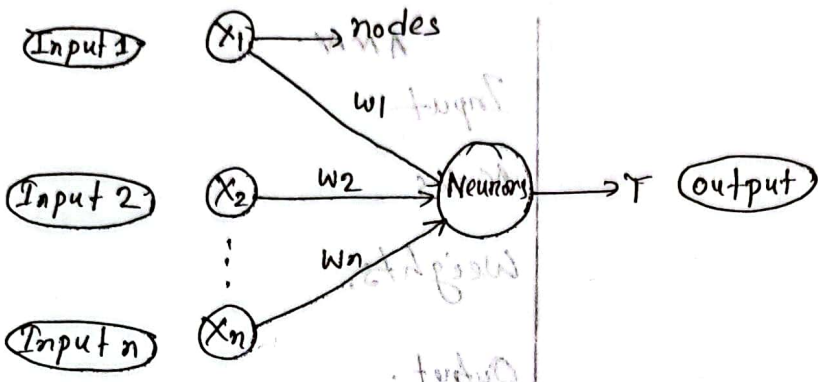


figure: Artificial neural network.

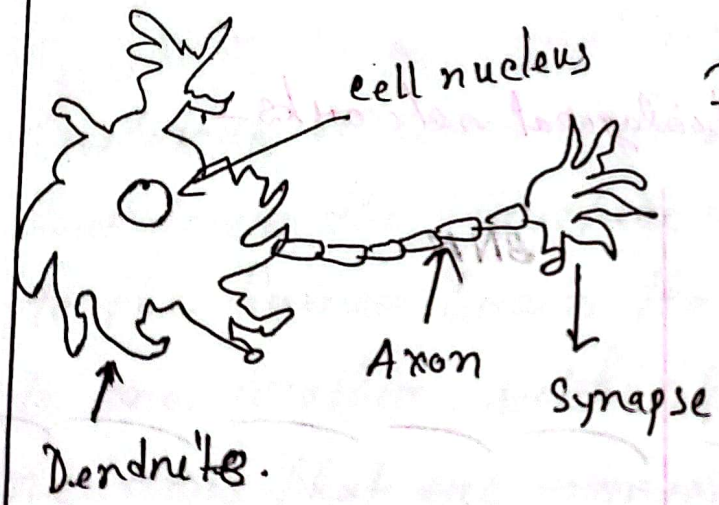
Compare artificial and biological networks –

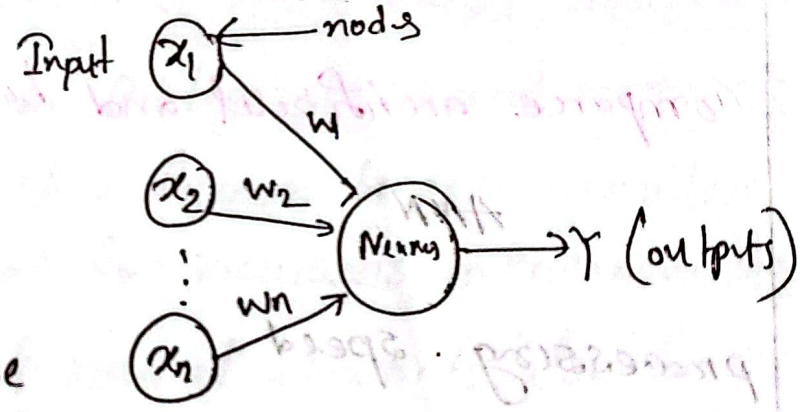
ANN	BNN
processing speed	

The brain is a neural network. It's a network of neurons. Artificial network, usually just referred to as a neural networks, are computer simulations which process information in a way similar to how we think the brain does it. BNN slow. ANN fast. Human neurons work parallely but ANN can't work parallelly.

Relationship between ANN and BNN.

BNN	ANN
Dendrites	Input
Cell nucleus	Nodes
Synapse	Weights.
Axon	Output.

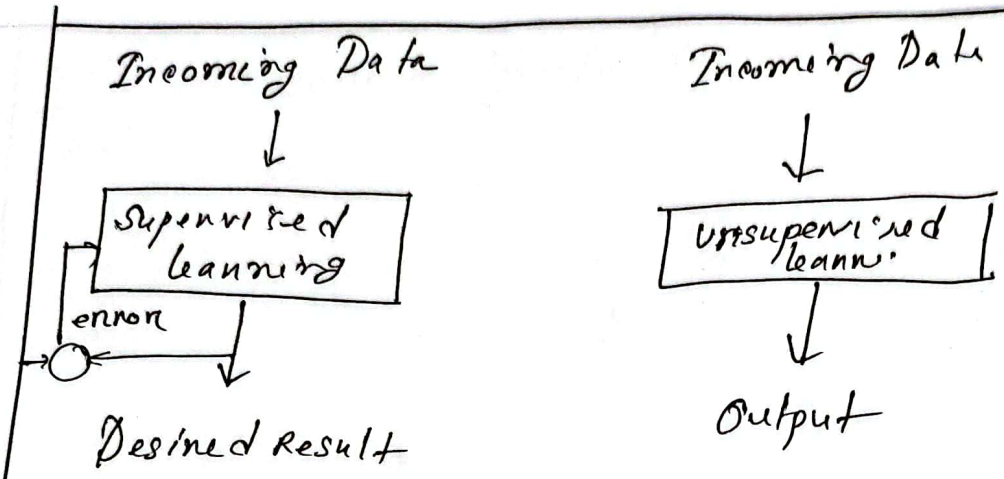


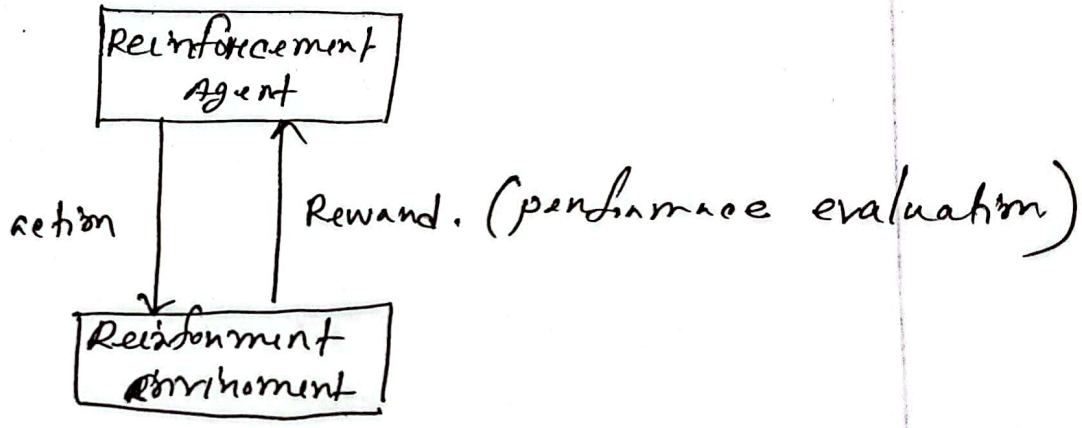


Learneing

Supervised | Unsupervised | Reinforcement

Criteria	Supervised	Unsupervised	Reinforcement
Defination	learn by using ~~leveled~~ labelled data. Both i/p are known	Trained using unlabelled data without any guidance	Works on interacting with the environment.
Type of data	Labelled data	Unlabelled data	No predefined data.
Types of problems	Regression and classification	~~No supervised~~ Association and clustering	Exploitation or exploration
Aim	calculate outcomes	Discover underlying pattern	Learn a series of actions.


Incoming Data
Supervised learning
error
Desired Result
Incoming Data
Unsupervised learning
Output


Reinforcement Agent
action
Reward. (performance evaluation)
Reinforcement environment

2019 8(f) Sources of uncertainity:

Slide-12

① Uncertain data: Missing data, unreliable, ambiguous, imprecise presentation, inconsistent, subjective, derived from defaults noisy.

② Uncertain knowledge: Multiple caused lead to multiple effects.
- Incomplete knowledge of causality in the domain.
- probabilistic / Stochastic effect.

③ Uncertain knowledge representation:
- restricted model of the real system.
- limited expressiveness of the representation mechanism.

④ Inference process:
- Derived result is formally correct, But wrong in the real world.
- Incomplete, default reasoning method
- New conclusions are not well founded.

① Data
② Expert knowledge
③ Inference process
④ knowledge representation

variable quantifiers.

FOL is the extension of PL more powerful

2018 4(c) FOL considered as the generation of PL.

propositional logic can deal with only a finite number of propositions. If there are three dogs Tommy, Jimmy and Linkon then

T: Tommy is faithful.

J: Jimmy " "

L: L " "

Express definite finite sentence

- All dogs are faithfull $T \wedge J \wedge L$

We cann't express is for the infinite number of dogs.

FOL/predicate logic is the generation of PL that allow us to express and infer arguments in infinite models. like,

- All men are mortal.
- Some birds cann't fly.
- At least one planet has life on it.

propositional logic assumes that there are facts that either hold or not hold. On the other hand, FOL assumes more. where it assumes the world consists of objects with certain relationships among them that do or do not hold. (Object, Relations and Functions). Like natural language.

Elimination $\forall$ and $\exists$

Universal elimination:

UE says that we can produce any sentence obtained by substituting a ground term of the variable. We use the notation substitutions for these instantiations. Let SUBST(θ, α) where substitution is θ and α is ~~the~~ sentence.

$$\frac{\forall v\ \alpha}{SUBST(\{v/g\}, \alpha)}$$

for any variable v and ground term g.

For example: "All greedy king are evil"

We can write: $\forall x\ King(x) \wedge Greedy(x) \Rightarrow Evil(x)$

For yields $\forall$,

SUBST (x/John)

- $King(John) \wedge Greedy(John) \Rightarrow Evil(John)$

- SUBST (x/Richard)
- $King(Richard) \wedge Greedy(Richard) \Rightarrow Evil(Richard)$,

Elimination of ∃.

For any sentence α, variable v and constant symbol k that doesn't appear elsewhere in the knowledge base:

$$\frac{\exists v\, \alpha}{SUBST(\{v/k\}, \alpha)}$$

Ex: $\exists x\, Crown(x) \wedge OnHead(x, John)$ yield:

$$Crown(C_1) \wedge OnHead(C_1, John)$$

provide C_1 is a new constant symbol which is not in KB. but satisfy all properties of 'x' called Skolem constant. We replace this variable with ground terms.

(Skolemization)

More powerful than PL with the variables and quantifiers, capture the sentence more natural way which is not possible in PL.

2016 3(a) Knowledge representation and mapping.

In order to solve complex problem encounted in AI, one need both a large amount of knowledge and some mechanism for manipulating that knowledge, which is called knowledge representation and the mechanism is mapping.

Facts are the things we want to represent.

Represention of facts: Things we can manipulate.

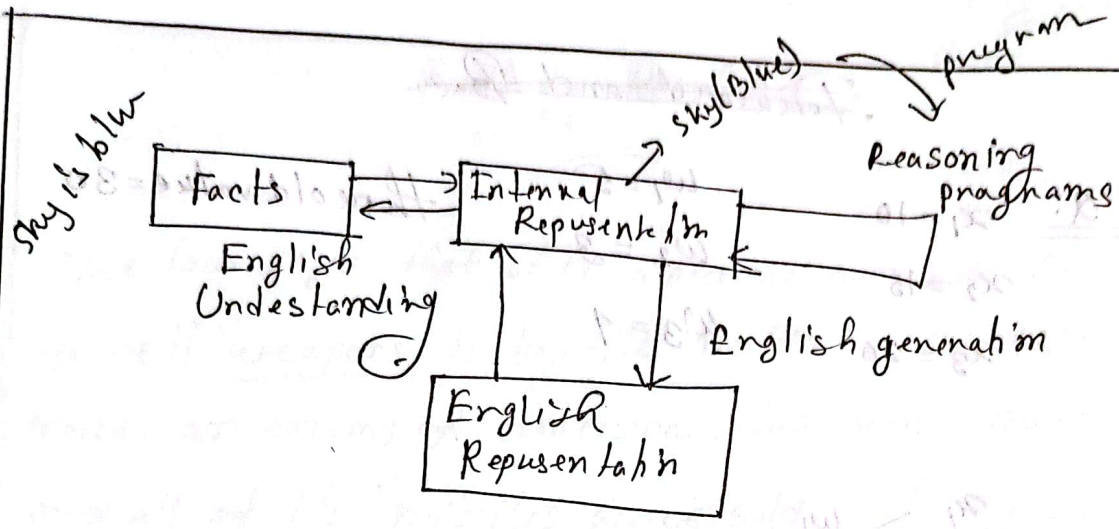
Sky is Blue
Sky(Blue)

Thus knowledge representations can be considered at two ~~lae~~ levels.

① knowledge level: At which facts are described and

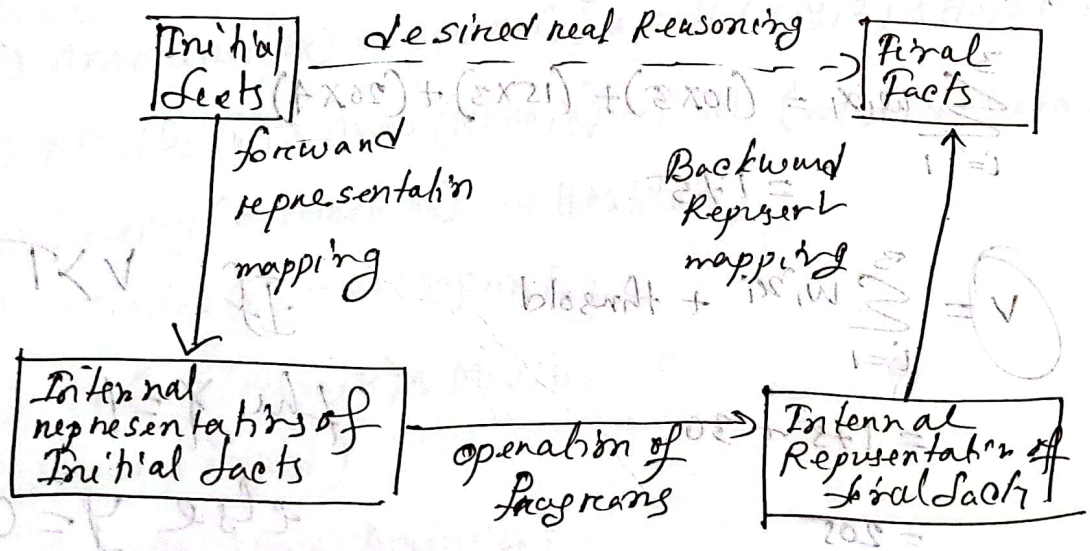
② Symbol Level: At which the representation of the objects, defined in the term of symbols, can be manipulated in the program.

Therefore we must be able to map "facts of symbols" and "symbols of facts" using forward and backward representation mapping.



KR & Mapping

Forward and backward Representation.

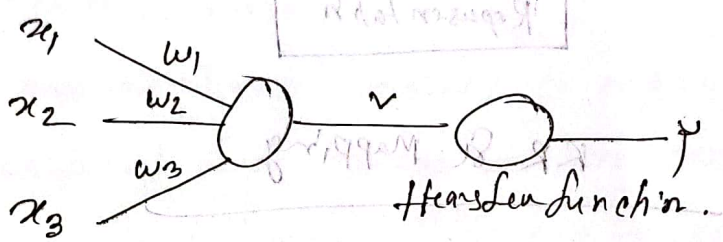


~~Forward and Bac~~

Q: $x_1 = 10$ $\quad w_1 = 5$ $\quad$ threold value $= 30$

$x_2 = 15$ $\quad w_2 = 3$

$x_3 = 20$ $\quad w_3 = 4$



Find out the value of y both sigmoid and threshold transfer function.

$$\sum_{i=1}^{3} w_i x_i = (10\times 5) + (15\times 3) + (20\times 4)$$

$$= 175$$

$$v = \sum_{i=1}^{3} w_i x_i + \text{threshold}$$

$$= 175 + 30$$

$$= 205$$

If $v > T$

the $y = 1$

else $y = 0$

Sigmoid function $\sigma S(x) = \dfrac{1}{1+e^{-x}}$

then $y = S(v)$

$$= \frac{1}{1+e^{-v}} \approx \frac{1}{1+e^{-205}}$$

$$= 1$$

① standardize variable
~~Skolem~~ ② ∃ ③ ∀

Proof by Resulation

The law says that it is a crime for an American to sell weapons to hostile nations. The country Nona, an enemy of America, has some missiles, and all of its missiles were sold to it by colonel West, who is American.

Prove: West is a Criminal

Criminal (West)

FOL

① American (x) ∧ weapon (y) ∧ sell (x, y, z) ∧ Hostile (z) → Crime (x)

② Missiles (n) ∧ Owns (Nono, n) → Sell (West, y, Nono)

③ Enemy (n, America) → Hostile (n)

④ missiles (n) → Weapon (n)

⑤ ∃n Own (Nono, n) ∧ Missiles

⑥ America (West)

⑦ Enemy (Nona, America).

① $\neg$ American(x) $\vee$ $\neg$ weapon(y) $\vee$ $\neg$ sells(x, y, z) $\vee$ $\neg$ hostile(z) $\vee$ criminal(x)

② $\neg$ Missile(x) $\vee$ $\neg$ owns(Nono, x) $\vee$ sells(West, x, Nono)

③ $\neg$ Enemy(x, America) $\rightarrow$ Hostile(x)

④ $\neg$ Missile(x) $\vee$ weapon(x)

⑤ owns(Nono, M1) $\wedge$ Missile(M1)

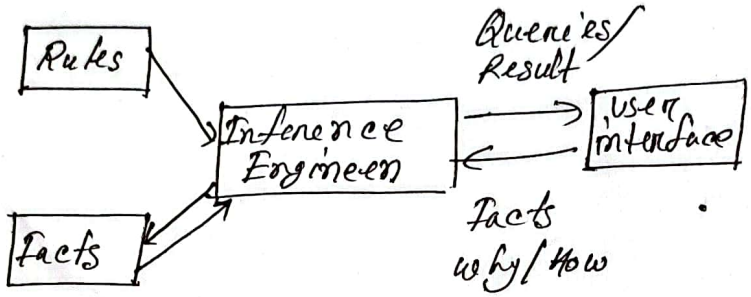
⑥ American(West)

⑦ Enemy(Nono, America).

2018 ⑥ ⓐ

Rules Based System

Rule-based system provide automatic problem solving tools for capturing the human expertise and decision making. It is used as a way to store and manipulate knowledge to interprete information in useful way.



Describe the way in which if-else rules can be used as a basis for knowledge representation and resoning in RBS.

Rule Based Experet System.

Any rule → IF (contion) Match
→ THEN (Action) ← then

Rules have two parts. First part is IF (condition part) and another is THEN. → Action part. If a condition is matched we will take a action.

OG
OR
OT

Example:

Rule 1: IF Traffic light is green THEN the action is go.

Rule 2: IF the T.L is red. then the action is stop.

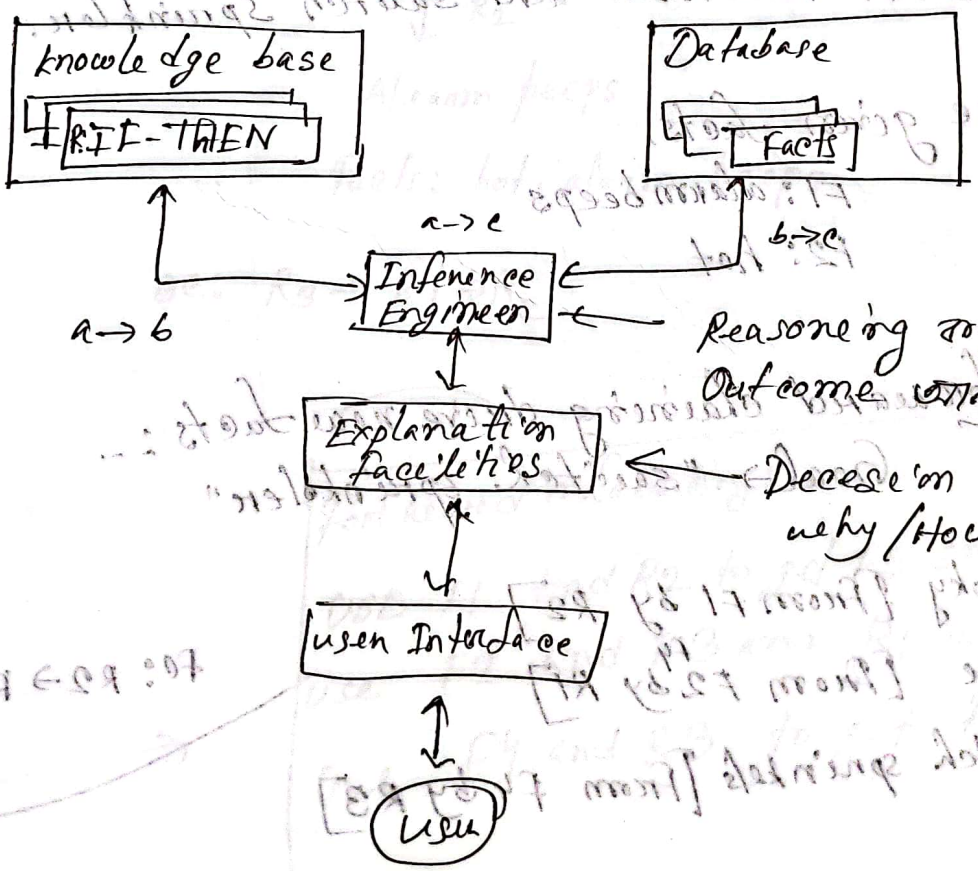
Rule 3: If the T.R is red. Then the action is go.

knowledge base এ এই IF-then rules গুলো store থাকবে।

conflict set

& Conflict Resulation method

Structure



If <antecedent> then <consequent>

[R2, R3]

→ Highest priority.
→ Most specific rules.
→ Data most recently entered.

6(b)

Forward chaining

① Starts with facts.
See what rules apply.

? R1: IF hot AND smoky THEN Add fire
R2: IF alarm_beeps THEN add smoky.
R3: IF fire THEN add switch sprinkler.

Here given facts,
F1: alarm beeps
F2: hot

Use forward chaining derive new facts :-
Goal → "switch sprinkler"

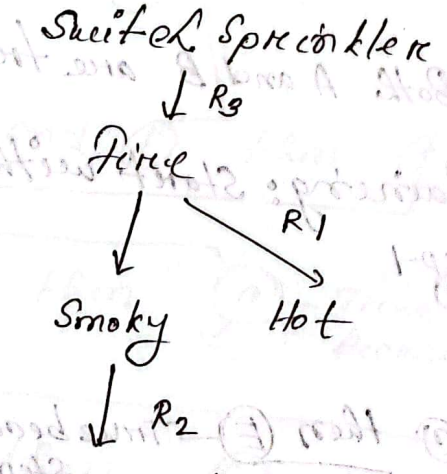
Data Driven

smoky [From F1 by R2]
fire [From F2 by R1] F4
switch sprinkler [From F4 by R3]

F1: R2 → R1 → R3

Backward chaining.

① Start with hypothesis.
② Goal driven.
③ looks for rules that will helps in answering it.



Facts: hot, alarm beeps.

BC: $R3 \rightarrow R1 \rightarrow R2$

Forward chaining:

USE F1 and R2 to get F3 smoky.
use F2 and F3 and R1 to get F4 fire
Use F4 and R3 to get F5 sprinkler on.

2019

R1: If A and C then E

R2: If C and D then F

R3: If B and E then F

R4: If B then C

R5: If F then G.

Initial facts Both A and B are trues.

① Forward chaining: Start with A.

Start step-1 (1)

Fact true ↓

R1: if (A) and (C) then (E) — True because of Step-2

R2: If (D) ~~then~~ (C) then (F) — Step-3

R3: If (B) and (E) then (F) — Step 4

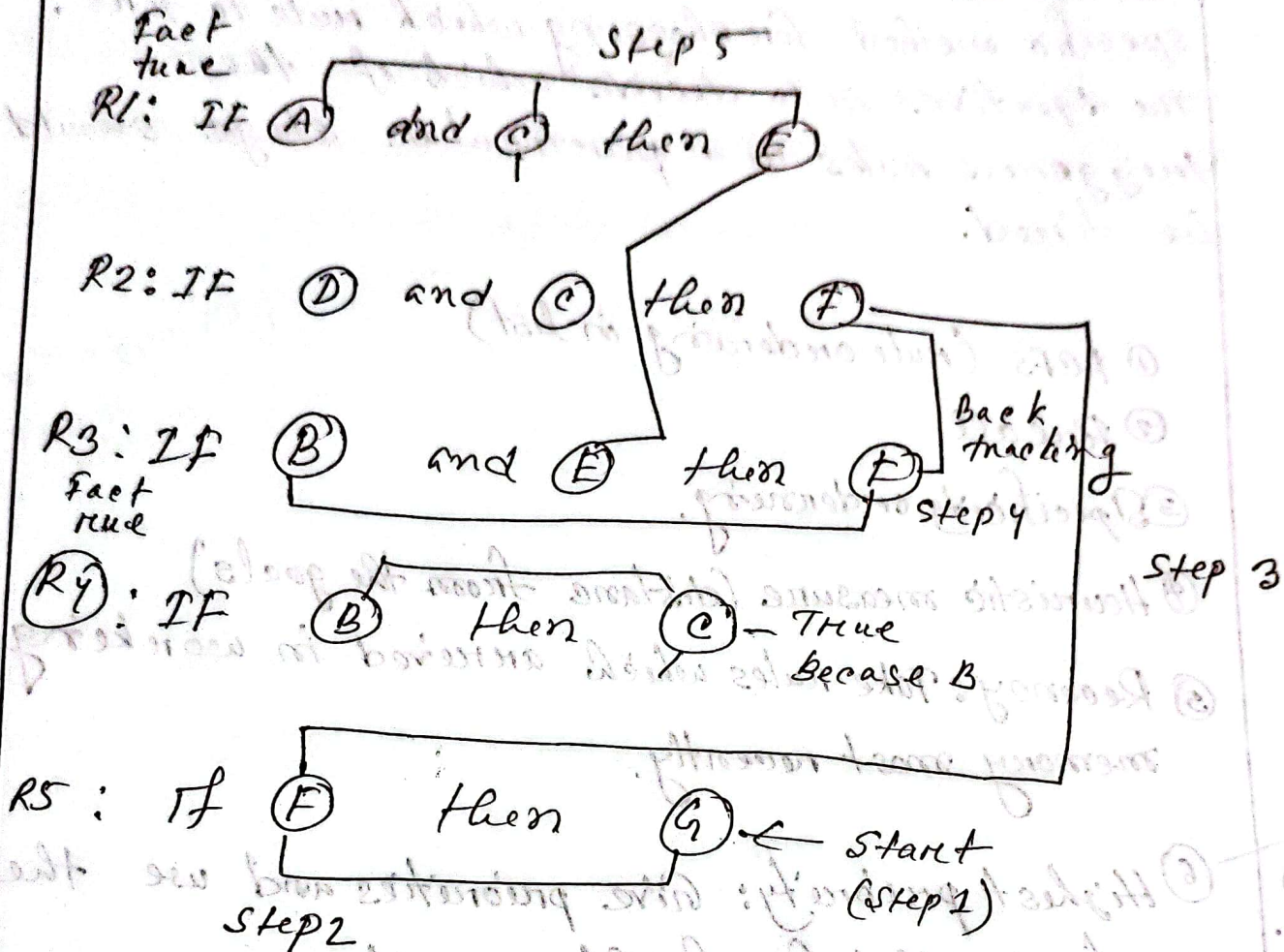
Fact true

R4: If (B) then (C) — True because of B — Step-2

(R5): If (F) then (G)

Step 5

Backward chaining: Start with G.



Conflict resulation: Conflict resulation provides a specific method for choosing which rule to fire. The objectives is to decide which of the triggered rules in a particular stage should be fired.

① FCFS (rule ordering in list)
② Fire all.
③ Specificity ordering.
④ Heuristic measure (distance from the goals)
⑤ Recency: Take rules which arrived in working memory most recently.
⑥ Highest priority: Give priorities and use the conclusion that has heighest priority.

which rule is more important which is not.

⑦ More specific rules:

Rule:
1. If it is raining.
2. If it is raining

Rule:
1. If you are going to school,
2. "
3. "
⋮
n

more premises execute

Remaining Question:

⑦ Refractoriness: Rules once fired will not be fired later.

⑧ Meta rules: Rules about rules embedded within the inference mechanism that provide the information about which of the rules apply under that conditions.

* Backtracking reduce

2020 6(b) Why planning is difficult to cast a state space search problem.

State space search problem is a most straight forward approach. planning is difficult to cast a state space problem because the description of actions in planning problem specify both the precondition and effect. It is possible to search in either forward and backward direction which is more complex.

planning also involves the modelling of a problem with an action language such as PDDL or ADL. or even with the propositional logic.

7(b) What will be happened if h underestimates and overestimate h in A* algorithm:

A* algorithm finds optimal solution if the heuristic function is carefully designed as the values is underestimated.

We we can generate h(n) which will never overestimate the actual value from current to goal then A* algorithm ensure to find an optimal path if one exists.

Overestimation of h(n) may or may not find the shortest path.

* Expert System: It is designed to solve complex problems. and provide decission making ability like a human expert. Normaly expert systems are applied to similar problems for which no algorithmic solution exist.

Essential capabilities of expert System:

① Demonstrate a device: It is capable of demonstrating any new product such as its features, specifications, how to use that product etc.

② problem solving: It has problem solving capabilities.

③ providing decision-making capabilities: making financial, ~~deci~~ etc.

④ Explaining a problem: It is also capable of providing a detailed description of an input.

⑤ prediction result: It is capable of predict result.

Application of Baye's theorm.

① It is used to calculate the next step of the robot when the already executed step is given.

② It is helpful in weather forecasting.

③ It can be solve the Monty hall problem.

FOL clauses

① elimination biconditions and implications.

$$\alpha \rightarrow \beta$$
$$\neg\alpha \vee \beta$$

② move $\neg$ inwards. (outside the close)

$$\neg(\alpha \vee \beta) = \neg\alpha \wedge \neg\beta$$

③ Skolemize.

$\exists \rightarrow$ variable into another variable

③ Universal qnt

$\forall n$ Person(n)

= Person(n)

③ $(\alpha \wedge B) \vee C = (\alpha \vee C) \wedge (B \vee C)$

2012 4(b)

perfect square

FOL

① $\forall x,y$ (perfect_square $(x) \wedge$ prime $(y) \wedge$ divisible (x,y)
$\rightarrow$ divisible $(x,$ square$(y))$ — ①

② $\forall x \exists y$ (perfect square $(x) \wedge$ prime $(y) \wedge$ disible (x,y)) — ②

③ perfect_square (36) — (III)

(II) For ① Clausal Form

eleminating Universal quantifiers,

perfect_square $(36) \wedge$ prime $(P) \wedge$ divisible $(36, P)$
$\rightarrow$ divisible $(36,$ square$(P))$.

eleminating implication sign,

$\neg$ Perfect_square $(36) \vee$ prime $(P) \vee \neg$ divisible $(36, P)$
$\vee$ divisible $(36,$ square $(P))$. — (IV)

② perfect_square $(36) \wedge$ prime $(P) \wedge$ divisible $(36, P)$ — (V)

Prove

(IV) $\exists y$ prime $(y) \wedge$ divisible $(36, \text{square}(y))$

eleminating $\exists$,

prime $(p) \wedge$ divisible $(36, \text{square}(p))$.

(III) prove by resulation,

So, we need to prove,

$\neg$ (prime $(p) \wedge$ divisible $(36, \text{square}(p))$)

$\neg$ prime $\vee \neg$ divisible $(36, \text{square}(p))$.

$\neg prime(p) \vee divisible(36, square(p))$

$perfect_square(36) \wedge prime(p)$
$\wedge divisible(36, p)$

$perfect_sq(36) \wedge divi$

$\neg perfect_sq(36) \vee$
$\neg prime(p) \vee divisible(36, p)$
$\vee divisible(36, sq(p))$.

$\neg prime(p) \vee divisible(36, p) \vee divisible(36, sq(p))$

(I) $\neg$ P.S(x) $\vee$ pr(y) $\vee$ $\neg$ div(x, y) $\vee$ div(x, su(y))

(II) perf_sq. (W)

(III) prim (P1)

(IV) div. (W, P1)

(V) perfect_sq (36)

(VI) pr (P2)

(VII) div (36, su(P2))

(I) $\neg$ Perfect_square(x) $\vee$ $\neg$ prime(y) $\vee$ $\neg$ divisible(x, y) $\vee$ div(x, s(y))

(II) prime(P1)

(IV) Div.. (W, P1)

(V) perf (36)

(VI) pr.

2020

(4) (1)

1. If a perfect square is divisible by prime p then it is also divisible by square of p.

$\forall x, y$ (Perfect-square $(x) \wedge$ prime $(y) \wedge$ divisible $(x, y) \rightarrow$ divisible $(x, \text{square}(y))$ — (I)

2. Every perfect-square is divisible by some prime.

$\forall x \exists y$ (perfect-square $(x) \wedge$ prime $(y) \wedge$ divisible (x, y)) — (II)

3. 36 is a perfect square.

Perfect-square (36) — (III)

4. Does there exist a prime Q such that square of Q divisible 36.

$\exists y$ (prime $(y) \wedge$ divisible (36, square (y))) — (IV)

(ii) clusal form..

For (i)

elimination (∀)

Perfect_square (x) ∧ prime (y) ∧ divisible (x,y) → divisible (x, square (y)).

eliminating (→)

¬Perfect_square (x) ∨ ¬prime (y) ∨ ¬divisible (x,y) ∨ divisible (x, square (y)).

For (ii)

elimination ∃ and ∀ → variable → new constant or arg or name

Perfect_square (x) ∧ prime (P1) ∧ divisible (x, P1)

can be written ~~two clau~~ 3 clauses.

Perfect_square (x)

prime (P1)

disible (x, P1)

(iii) Perfect_square (36).

(iv) ∃y (prime (y) ∧ divisible (36, squa (y)).

elimi ∃

Prime (P2) ∧ divisible (36, squa (P2)). 2 clause

(iii) Using resolution method:

We have to prove.

~~∃y (prim (y))~~

prime (P_2) ∧ divisible (36, prime (P_2))

~~We we need~~ We disprove -

¬prime (P_2) ∨ ¬divisible (36, prime (P_2))

¬prime (P_2) ∨ ¬divisible (36, prime (P_2)) prime (P_1)

P_1/P_2

P_1

¬divisible (36, prime (P_2)) ¬perfect_square (k) ∨ ¬prime (y) ∨ divisible (n, y) ∨ divisible (x, prime (y))

¬perfect_square (n) ∨ ¬prime (y) ∨ ¬ divisible (n, y) perfect_square (n)

¬prime (y) ∨ ¬ divisible (n, y) divisible (n, P_1)

¬prime (y) prime (P_1)

null

2020

Resolution is a proof method for classical propositional and FOL.

(4)

(a) Translate these facts into well formed formulas in predicate logic.

(1) Marcus was a man

— man (Marcus).

(2) Marcus was a Pompeian

— Pompeian (Marcus)

(3) Marcus was born in 40 AD.

— born (Marcus, 40)

(4) All man are mortal.

$\forall x$ (man(x) $\rightarrow$ mortal(x))

(5) All Pompeians died when all the volcano erupted in 79 A.D.

~~$\forall x$ died Pompeians(x) $\rightarrow$ died~~

erupted (volcano, 79) $\wedge$ $\forall x$ (Pompeian (x) $\rightarrow$ died (x, 79))

(6) No mortal lives longer than 150 years.

$\forall x \forall t1 \forall t2$: mortal (x) $\wedge$ born (x, t1) $\wedge$ gt (t2 − t1, 150) $\rightarrow$ dead (x, t2)

(7) It is now 2021

now = 2021

(8) Alive means not dead.

$\forall x \forall t\ (alive(x,t) \rightarrow \neg dead(x,t)) \wedge (\neg dead(x,t) \rightarrow alive(x,t))$

(9) If someone dies, then he is dead at all later times.

$\forall x \forall t1 \forall t2\ (died(x,t1) \wedge gt(t2,t1) \rightarrow dead(x,t2))$

(b) ~~Answer~~ "Is Marcus alive now?" using backward ~~chaining~~. reasoning.

¬alive(Marcus, now)

¬alive(Marcus, now)

↑ (8, substitution)

dead(Marcus, now)

↑ (9, substitution)

dead(Marcus, t1) ∧ gt(now, t1)

↑ (4, substitution)

pompeian(Marcus) ∧ gt(now, 79)

(1)

↑

gt(now, 79)

↑ (7, substitution)

gt(2021, 79)

↑ Nil Marcus is dead.

$\forall n \forall t1 \forall t2$ man $(n, t1) \wedge$ born $(n, t1) \wedge$ gt $(t2-t1, 150)$.

Convert formulas into clausal form,

① man (marcus)

② Pompeian (Marcus)

③ born (Marcus, 40)

④ $\neg$ man (n) $\vee$ mortal (n).

⑤ erupted (volcano, 79) $\wedge \neg$ Pompeians (n) $\vee$ dead (n, 79)
can be written as,

erupted (volcano, 79)

$\neg$ Pompeians $\vee$ dead (n, 79)

⑥ $\neg$ mortal (n) $\vee \neg$ born (n, t1) $\vee \neg$ gt (t2 − t1, 150) $\vee$ dead (n, t2)

⑦ now = 2021

⑧ $\neg$ alive (n, t) $\vee$ dead (n, t)
$\neg$ dead (n, t) $\vee$ alive (n, t)

⑨ $\neg$ deal (n, t1) $\vee$ gt (t2, t1) $\vee$ dead (n, t2)

(4) prove "Marcus is not alive now using resolution.

¬alive (Marcus, now)

We need to prove

alive (Marcus, now)

alive (Marcus, now)

¬alive (x,t) ∨ ¬dead (x,t)

¬dead (x,t)

~~¬dead (x,t) ∨ gt(t2,t1) ∨ dead (x,t)~~

~~dead~~

¬pompeian (x) ∨ dead (x,t9)

~~¬dead (x,t) ∨ gt(t2,t1)~~

~~¬mortal (x) ∨ ¬born (x,t) ∨ gt(~~

¬pompeian (x)

pompeian (Marcus)

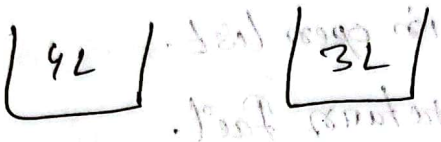
null.

* Write down A* Search algorithm.

1) Enter starting node in open list.
2) If open list is empty return Fail.
3) Select node from the openlist which has smallest value (g+h)
   - If it is goal node return success.
4) Expand node n and generate all successor
   - compute (g+h) for each successor node.
5) If node is already in the open list (closed, attach back pointer)
6) Go to 3

~~Simple relational knowledge:~~

Water jug problem.



How can you get exactly two gallon of water in the 4 gallon jug.

$(x, y) \rightarrow (4, y)$

$(x, y) \rightarrow (x, 3)$

$(x, y) \rightarrow (x-d, y) \quad x > 0$

$(x, y) \rightarrow (x, y-d) \quad y > 0$

$(x, y) \rightarrow (0, y) \quad x > 0$

$(x, y) \rightarrow (4, y-$

$x = 0, 1, 2, 3, 4$ number of gallon of water in 4 g jug

$y = 0, 1, 2, 3$ " " " " 3 "

Start state $(0, 0)$ goal state $(2, n)$

4 gallon jug		3 gallon jug
0		0
0		3 fill
3	← Transfer	0
3	← Transfer	3 fill
4		2
empty 0		2
2	← transfer	0

Entailment: It is the relationship between two sentences. where the truth of one requires to truth of another. A sentence logically follow another sentence.

A ⊨ B.

A entails B if for every model in which A is true, B is also true as well.

Hypothesis: Claim করা by verifying with the data. (available information দিয়ে)

কোনো ~~কিছু~~ কিছু verify করতে চাচ্ছি,

claim কে test করার

চেষ্টা করতেছি (or

প্রতিপন্ন করা by

using given set of

information.

Approach to knowlage representation

① Simple relational knowledge

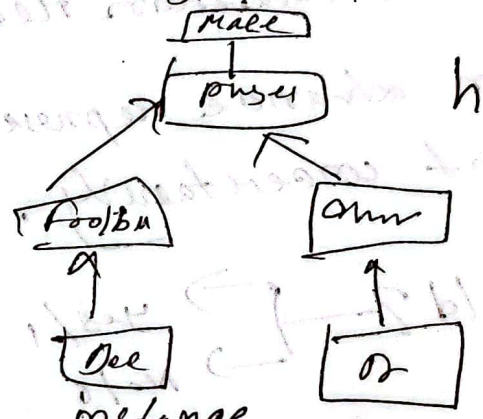
Simplest way of data storeing using relational table.

player	Weight	Age

Representing information in the form of table

② Inheritable knowledge

Data গুলো কে একটি heirachy of class অনুযায়ী represent করা হয়



hierarchy

instance → 4

Relation between instace and class.

unclean image
noise

Fuzzy logic:

not clean

Fuzzy logic is a method of reasoning that resembles human reasoning.

It is an advance representation technique to represent uncertainty.

It is cold? → yes/1, No/0 → Boolean

It is cold? → very much/0.9, little, very little: .25/0.1 → Fuzzy logic