Midsem Notes For
computational Intelligence
PE-1 Artificial Intelligence

MANIPAL INSTITUTE OF TECHNOLOGY
BENGALURU
(A constituent unit of MAHE, Manipal)

20/30

VI Semester Re-MidTerm TEST
Artificial Intelligence (PE-I) (CSE_4053)

Time Duration: 2 Hours **Date: 12.04.2024** **Max marks: 30**

Question No	Topic	Marks	BL	CO
1.	Which type of AI can operate in real time without human intervention and learn from its environment? a) Narrow AI b) Reactive AI c) General AI d) Reinforcement Learning Agent	0/1 1	1	1
2.	Which domain of AI is primarily used in devices like Amazon Alexa or Google Assistant? a) Robotics b) Machine Vision c) NLP (Natural Language Processing) d) Expert Systems	1/1 1	2	1
3.	A problem in a search space is defined by _______. a) Initial state b) Initial state and goal state c) Initial state, goal state, and intermediate states d) Sensor inputs	0/1 1	1	1
4.	If a robot can clean the room by analyzing the layout and learning from mistakes, which AI concept is applied? a) General AI b) Supervised Learning c) Reinforcement Learning d) Deep Learning	1/1 1	2	3
5.	In the context of intelligent agents, what are percept sequences? a) Actions performed by the agent b) Goals set by the agent c) Observations received from the environment over time d) Internal states of the agent	1/1 1	2	2
6.	Reinforcement Learning uses ________. a) Decision trees b) Trial and error c) Historical data d) Fixed rules only	0/1 1	2	2
7.	A self-driving car using AI must decide between stopping for a pedestrian or continuing due to high-speed traffic behind. Which AI technique is most suitable for this scenario? a) Supervised Learning b) Rule-based Expert System c) Reinforcement Learning d) General AI	1/1 1	2	1

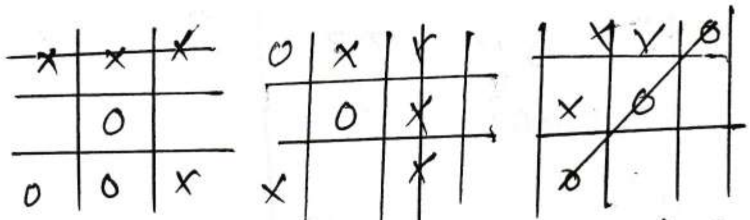
8.	What is the main advantage of Reinforcement Learning over Supervised Learning? a) It does not need labeled data b) It is faster than supervised learning c) It always gives 100% accurate results d) It does not require large datasets	1 1/1	1	1
9.	Which of the following is an example of unsupervised learning? a) Identifying spam emails from labeled examples b) Grouping customers based on their purchase history c) Training a self-driving car through rewards and penalties d) Translating a document from English to French	1/1 1	1	1
10.	Which AI application best represents the use of Computer Vision? a) Predicting customer buying behavior b) Detecting objects in surveillance footage c) Recommending movies on Netflix d) Generating human-like text responses	1/1 1	2	1
11 (a)	Design the specification in terms of *PAGE* for setting vacuum world agent.	**5**	**5**	**3**
11(b)	Write in detail about propositional logic and FOPL. Present suitable examples for each. Present a comparative study on these two schemes in not less than four points.	**5**	**4**	**3**
12(a)	Compare between *simple reflex agent* and *goal-based agent* along with neat sketches.	**5**	**3**	**4**
12(b)	Convert the following expression into clausal form; also, provide the detailed steps with explanation. $\exists x \exists w \forall y (\forall z P(f(x, y), w, z) \rightarrow (\exists u Q(w, u) \& \exists v R(y, v))$	**5**	**4**	**4**

Note: BL refers to Bloom's Taxonomy Level.

Artificial Intelligence.

Tic-Tac-Toe Game playing

- It's a paper and pencil game of two players x and o, who chooses to mark a spaces on a grid of 3x3.
- The game is won by player who succeeds in putting three of their marks in a horizontal diagonal line



x wins x wins o wins

objective: T.W. a computer program in such way that computer wins most of time.

Three approaches

- → complexity
- → use of Generalization
- → clarity of their knowledge
- → Extensibility of approach

Board Data structure

1	2	3
4	5	6
7	8	9

2D Game Board

The cells could be represented as centre square, corner, edge as below

corner	Edge	corner
Edge	centre square	Edge
corner	Edge	corner

1 indicating x player move

2 : indicating o player move

→

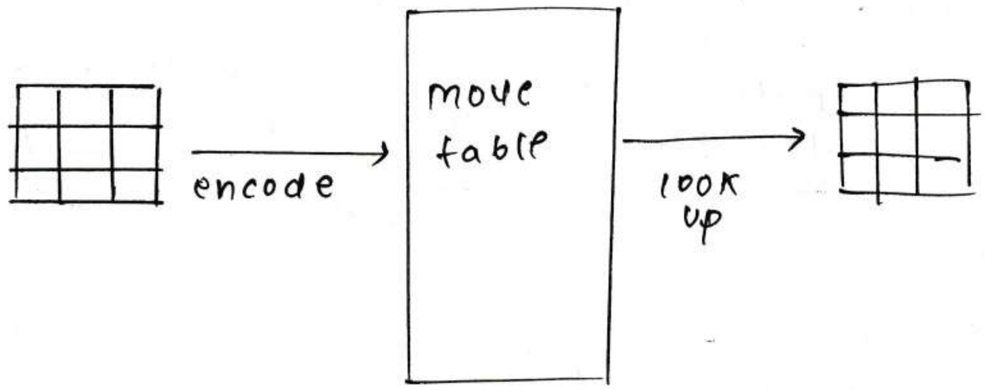
1	2	3	4	5	6	7	8	9	10

1D vector

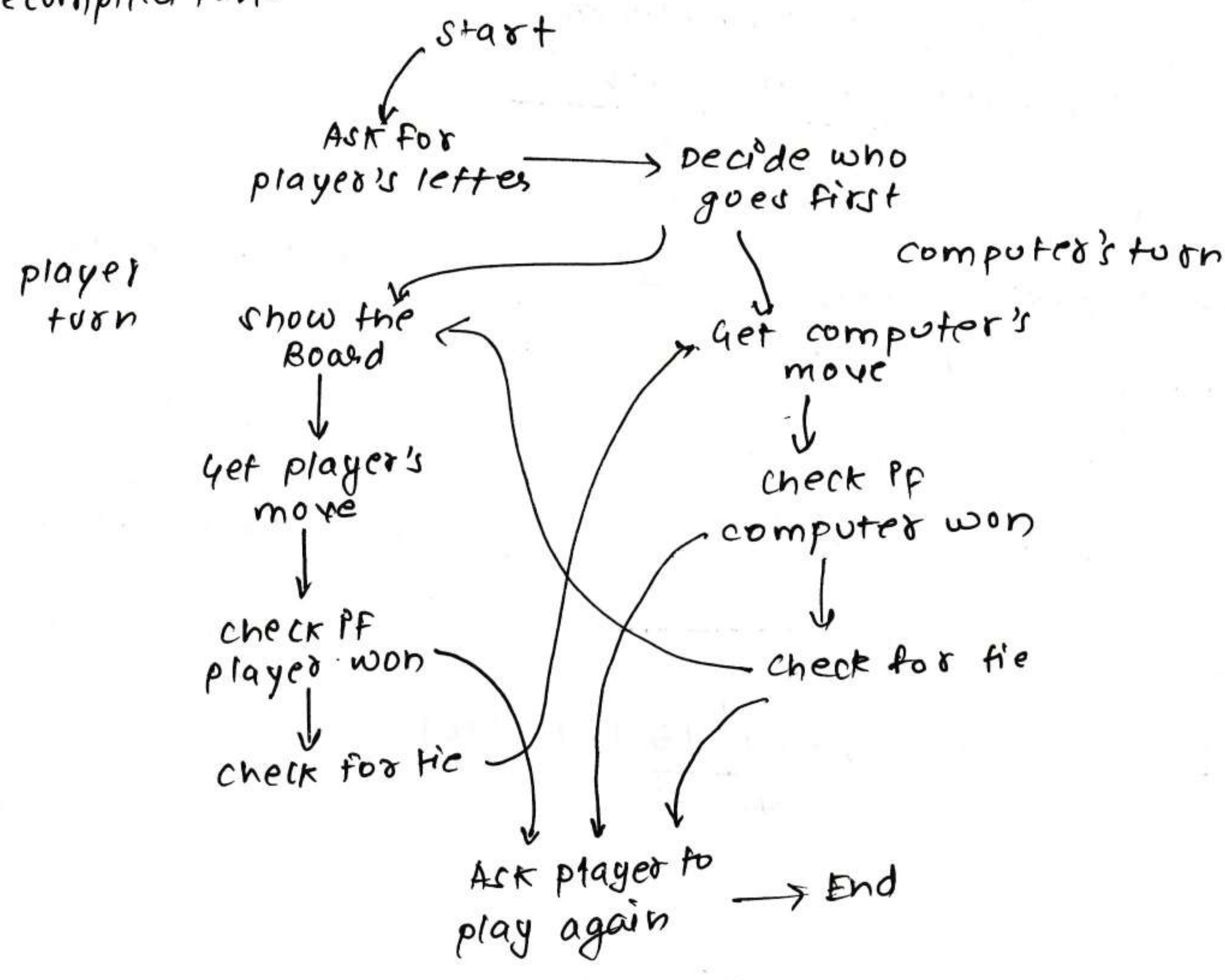
Move Table: It is a vector of 3^9 elements, each element of which is nine-element vector representing board position

Total of 3^9 (19683) elements in move table

index	current Board pos	New Board pos
0	000000000	000010000
1	000000001	020000001
2	000000002	000100002
3	000000010	002000010



precompiled table



Program 2: Tic Tac Toe Game Playing Magic square

We Assign board positions to vector position, sum of all rows and diagonals must be:-

$15-(5+4)=6$

- Algorithm :-

1) First machine will check, chance to win. $15-(4+1)=10$

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

Example:

2) If the difference is not positive or If it is greater than 9, then original two squares, were not collinear, so it can be ignored

3) check if opponent winning block chances of winning

- previous state

H		C
	C	
H		

$Diff = 15-(5+4)=6$
6 is empty, hence computer can't win game

$Diff = 15-(8+6)=1$

- Turn computer(C)

H		C
C	C	
H		

1 is empty, hence Human can win the game
Hence computer it
computer - go to 1

MAGIC SQUARE

- Turn-computer

	C	

Turn computer (C)

H		C
	C	

Turn - Human

H		
	C	

Turn-Human CH

H		C
	C	
H		

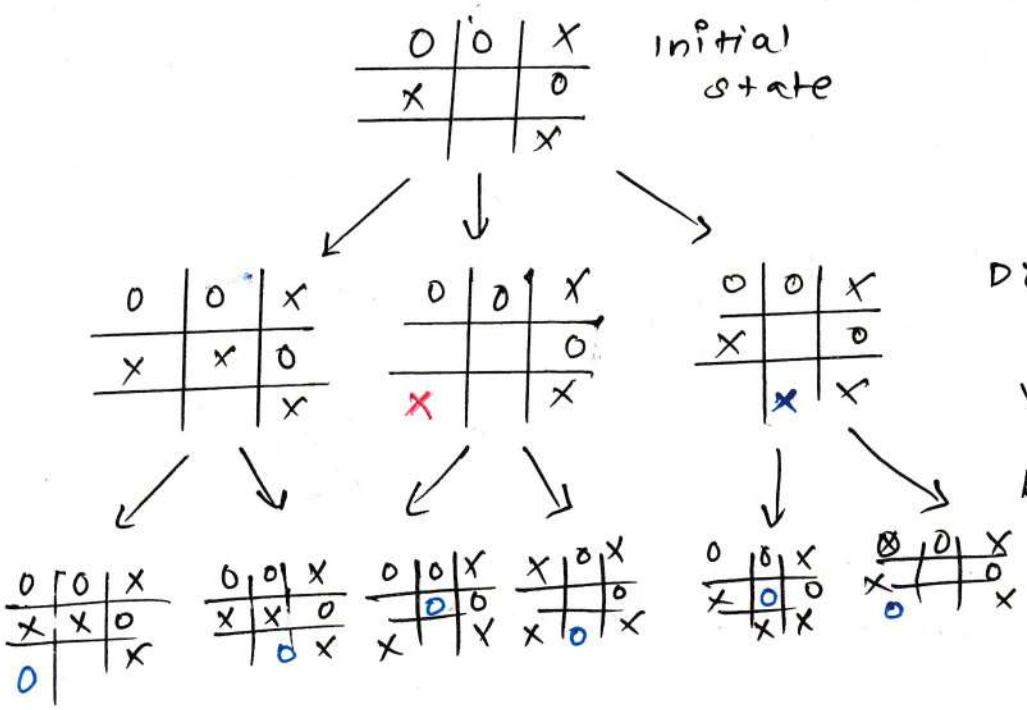
Tic-Tac-Toe: Game playing - Program 3

Data structures

- Board position is a structure containeining

- A 9-element array representing the board
- A List of board positions that could result from next move.

- A number of rating representing an estimate of now likely the board position is to lead an ultimate win for the player to move

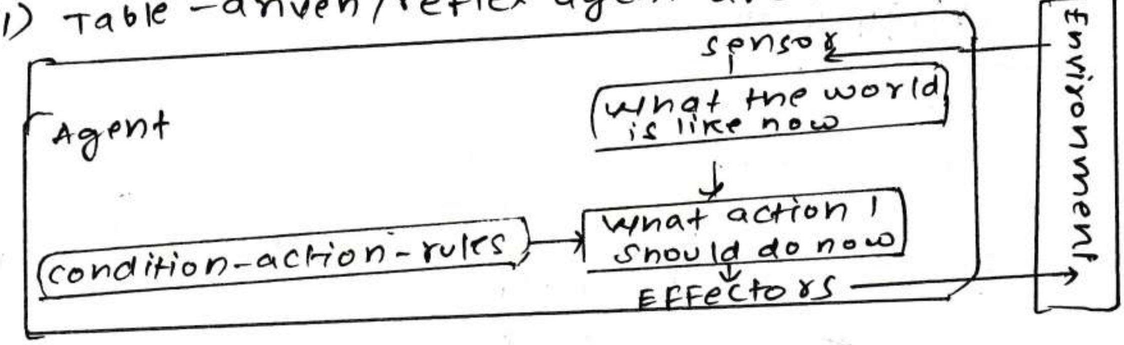


Draw - Return 0 to patent

Win - Return 1 to patent

Backtrack decide which move is best

(0/1) Table-driven/reflex agent architecture



(0) Table-driven agent

Table lookup of percept-action pairs mapping from every possible perceived state to the optimal for every state

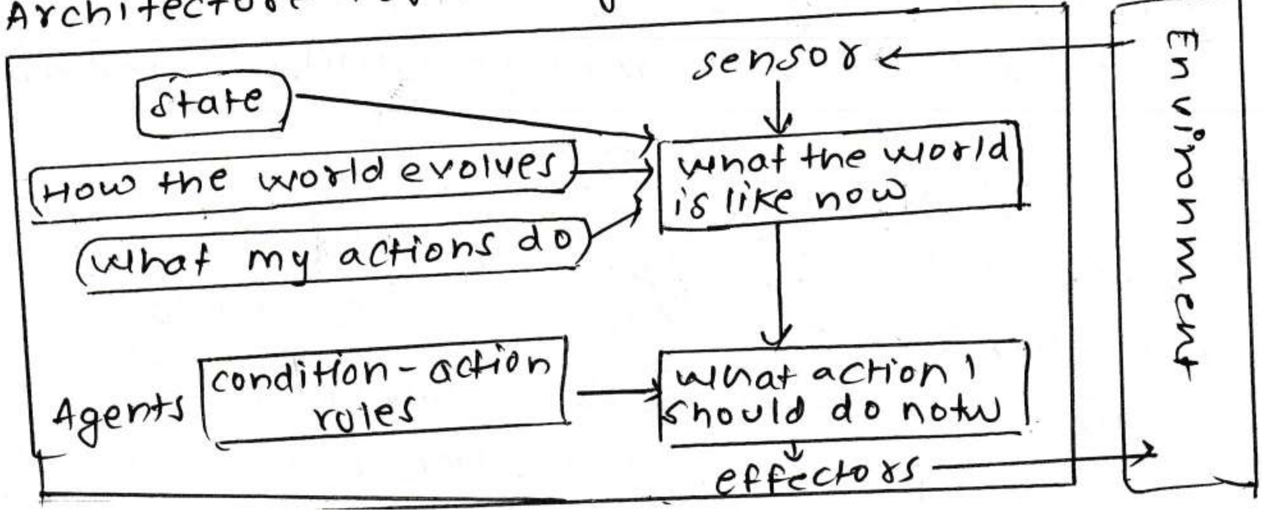
problem
- Too big to generate and to store (chess has about 10^{120} states for example)
- No knowledge of non-perceptual parts of the current state
- Not Adaptive to change in environment; requires entire table to be updated if changes occur.
- Looping: can't make actions conditional or previous actions/states.

(1) Simple reflex agents

Rule-based reasoning to map from percept to optimal action; each rule handles a collection of perceived states.

problems
same problems as Table driven agen

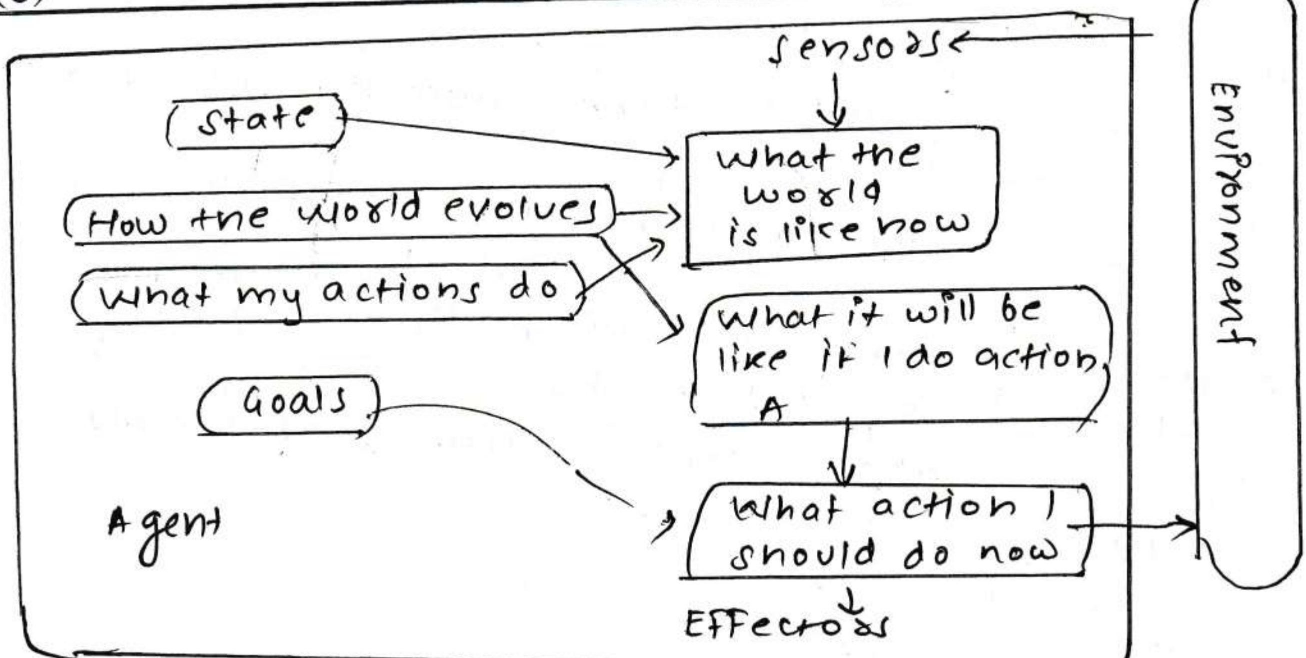
(2) Architecture for an agent with memory



(2) Agents with memory

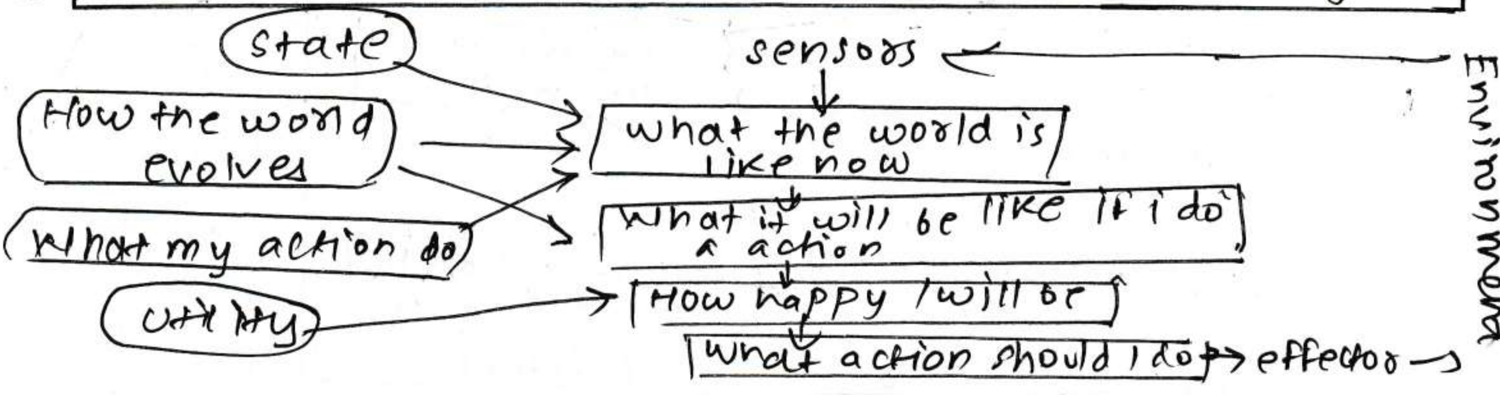
- Encode "internal state" of the world to rember the past as contained in earlier percepts.
- Needed because sensors do not usually give the entire state of the world at each input, so perception of environment is captured over time. "state" is used to encode different "world states" that generate same immediate percept

(3) Architecture for goal-based agent



- choose actions so as to achieve a (given or computed) goal.
- A goal is a description of desriable situation or state.
- Deliberative instead of reactive.
- May have to consider long sequences of possible actions before deciding if ~~ga~~ goal is achiad

(4) Architecture for a complete utility-based agents



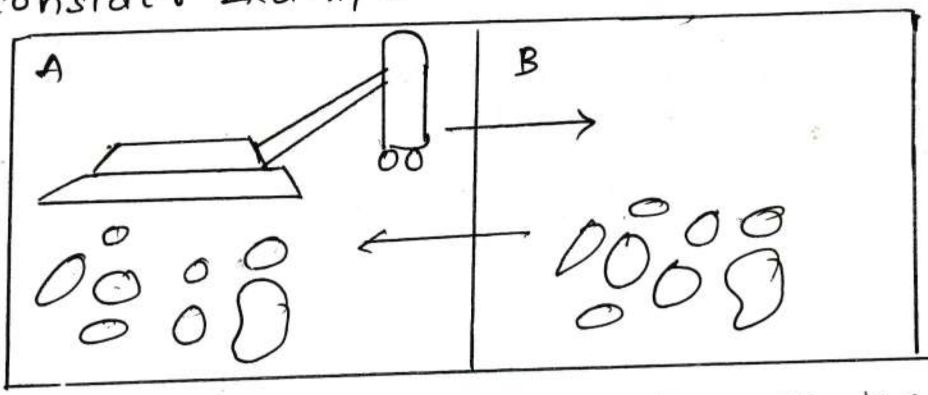
Module 2: Problem solving by Search and Exploration

A problem can be defined formally by six components

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

State: The state is determined by both the agent location and the dirt locations.

Consider Example



Goal is that vaccum cleaner has to clean the dirt in both the states

The agent is in one of two locations, each of which might or might not contain dirt

Thus there is $2 \times 2^2 = 8$ possible world states

A larger Environment has n location has $n \times 2^n$ states.

1. {LOC(A), Dirty(A), Dirty(B)}
2. {(LOC(A), clean(A), Dirty(B)}
3. {LOC(A), Dirty(A), clean(B)}
4. {LOC(A), clean(A), clean(B)} – Goal state.

5. {(LOC(B), Dirty(A), Dirty(B)}

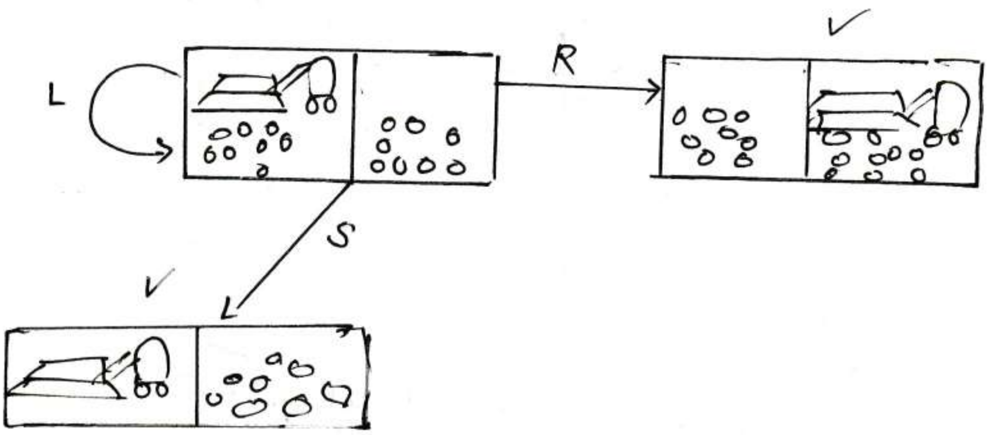
6. {(LOC(B), Clean(A), Dirty(B)}

7. {(LOC(B), Dirty(A), Clean(B)}

8. {(LOC(B), Clean(A), Clean(B)} | Goal state

Action: Three actions : Left, Right, suck

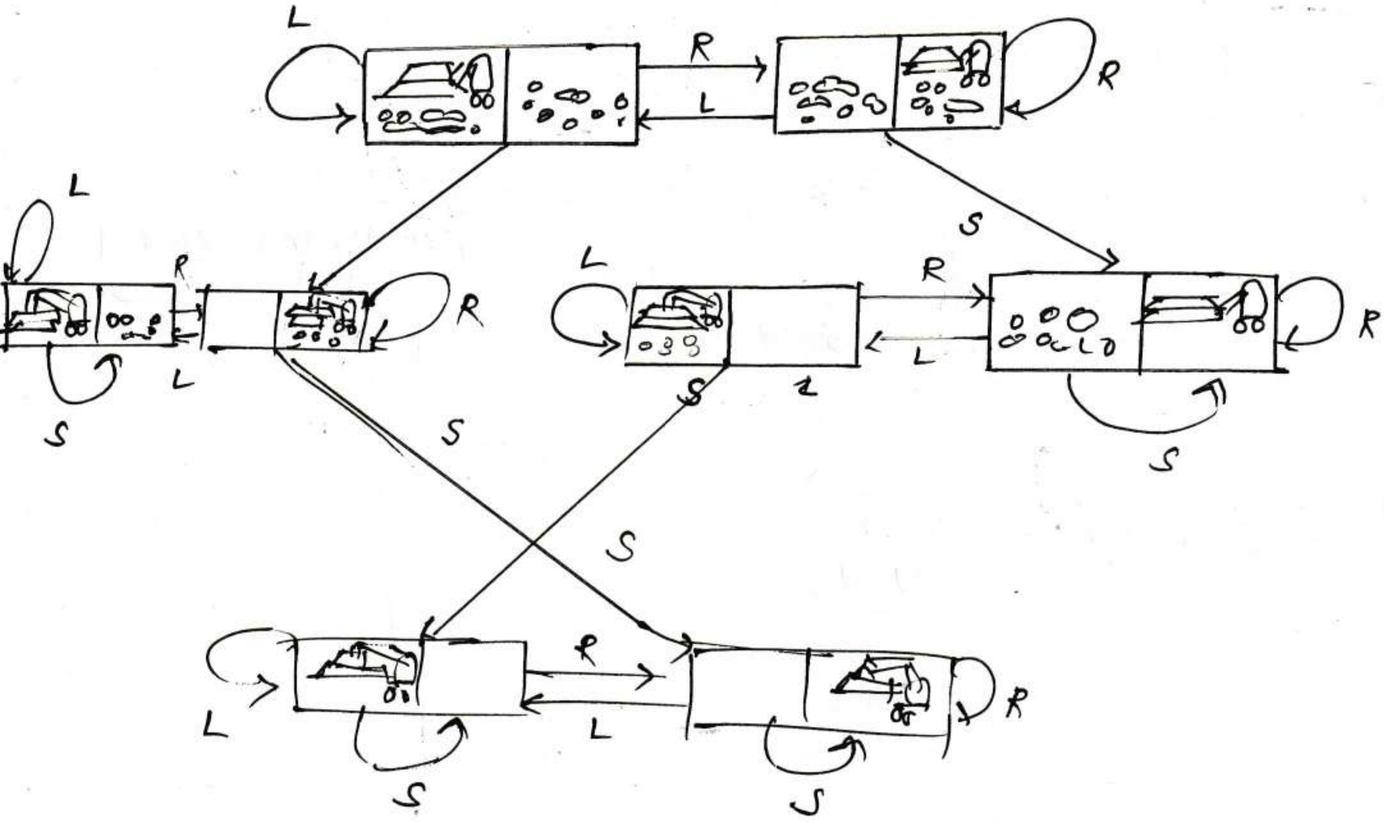
Transition model: The actions have their expected effects that moving Left in the leftmost square, moving right in the rightmost square, & sucking in clean square have no effect



Goal test: This check whether all squares are clean

path cost: Each step kost 1, so cost is number of steps in path.

Transition Model - vaccum world



States ?

actions ?

goal of test ?

path cost ?

problem solving agent

problem formulation

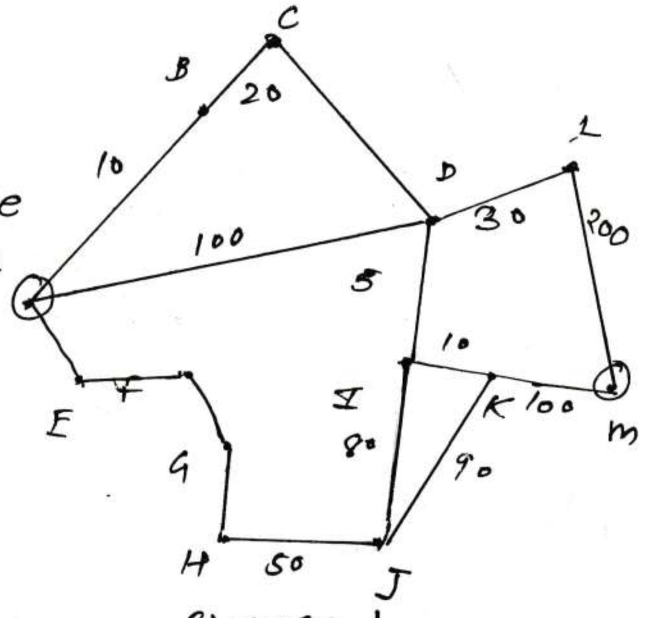


figure. 1.

RESULT (S, G)

1. Initial state: The initial state is A.

2. Action set: From A the Applicable actions are {Go(B, D, E}

3. successor function:

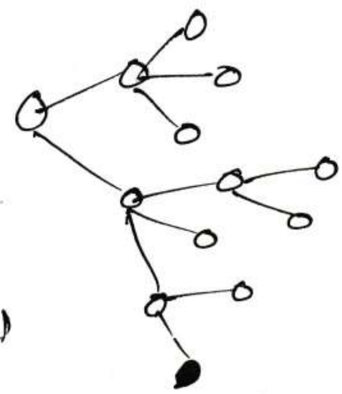
RESULT # (In(A), Go(B)) = In(B)

4. Goal state :- The agents goal state is {In(m)}

5. Path cost: Distance between two transition is path cost

uninformed searching

1. searching without information (Blind search)
   Has informed about only two things
   - # what is start state
   - # what is goal state



Dosen't has any guide to take it to goal state

- can distinguish between goal state & Non-goal state

2. More time consumption and High chances of optimal solution

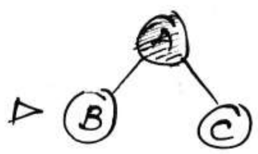
Example: BSF, BFS, DFS, etc.

Breadth First Search

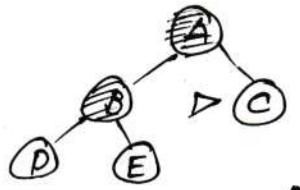
1. uninformed search technique

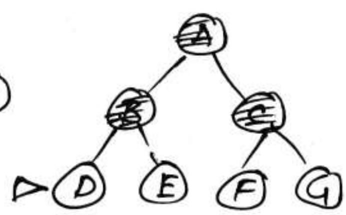
- The root node is expanded first (FIFO)

- All the nodes generated by the root node are then expanded.

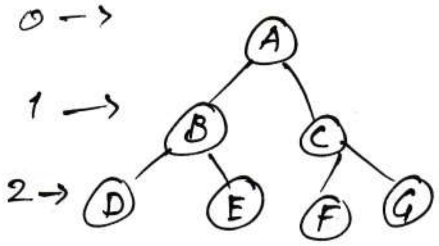


- And their successor and so on.





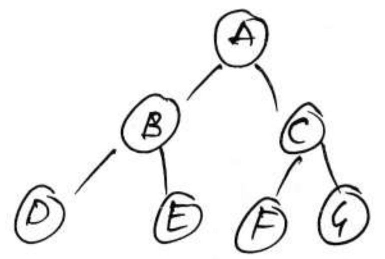
2. Working (FIFO)



Gives shortest solution.

Time complexity

DS → O(V+E)
AI → O(b^d)

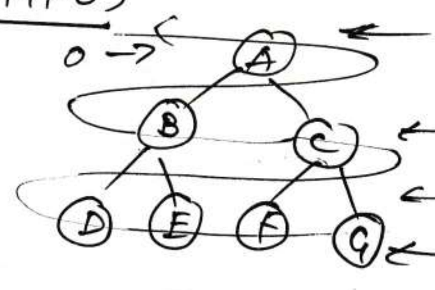


{~~A~~}
A {~~B~~ C}
AB {~~C~~, D, E}
ABC {~~D~~ E F G}
ABCD {~~E~~ F G}
ABCDE {~~F~~ G}
ABCDEFG

(6) Algorithm.

i) Enter starting nodes in queue
ii) if → queue empty return fail and stop
iii) if → first element in queue is goal node then return success & stop.
Else
iv) Remove and expand first element in queue and place children nodes at the end of the queue
v) Goto step (ii)

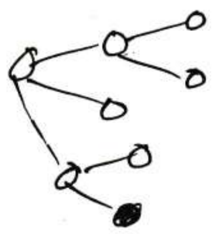
Working (FIFO)



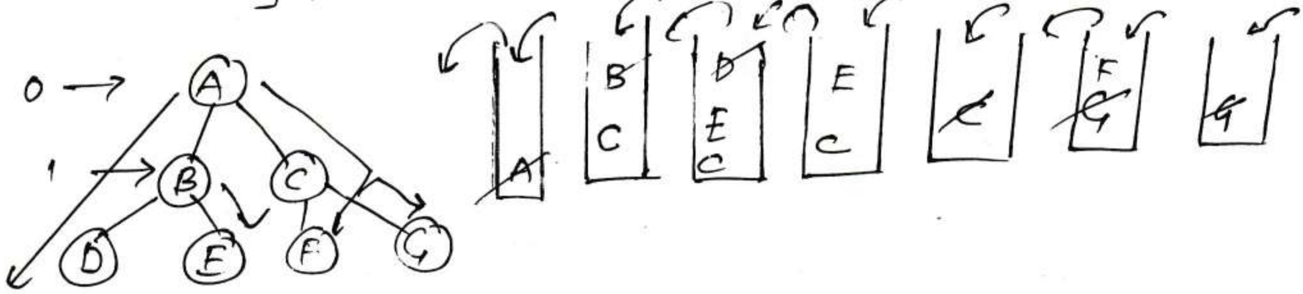
A {~~A~~}
{~~B~~, C}
{~~C~~, D, E}
{~~D~~ E F G} {E, F, G}
{F, G}
{~~G~~}

Depth First search (DFS)

1. uninformed search techniques

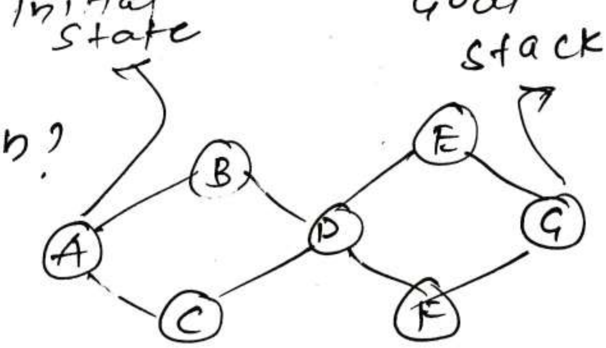


2. working (LIFO)



3) works vertically
4) Doesn't guarentee a solution always
5) Time complexity
   DS → $O(V+E)$
   AI → $O(b^d)$
6) Algorithm.
   1. push the root node in the stack
   2. while (stack is not Empty)
      a) pop a node from the stack
         i) If node is a goal node
            return success
         ii) push all children of node
            onto the stock
   3. return Failure.

Bidirectional search



1. what is Bidirectional search?
2. Time complexity: $2(b^{d/2})$
3. Pros
   i) Much Efficient and Fast
   ii) Reduce time requirement
   suppose $b=10$ $d=6$

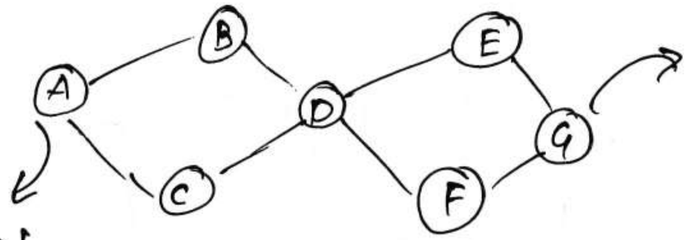
B.F.S :- $b^d = 10^6 = 1000000$ nodes

B.S = 2×10^3 = 2000 nodes

cons

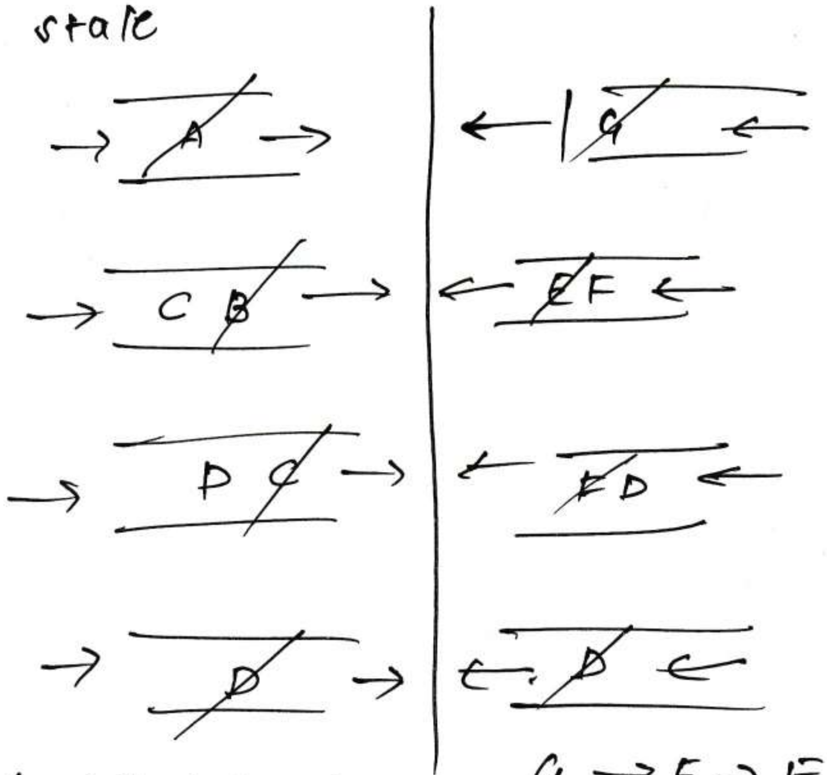
i) Implementation is difficult

ii) Goal state must be known in advance



Goal state

Initial state



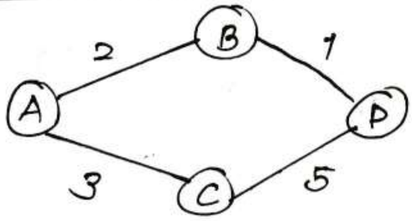
A → B → C → D

G → E → F → D

combine both

path A → B → C → D → F → E → G

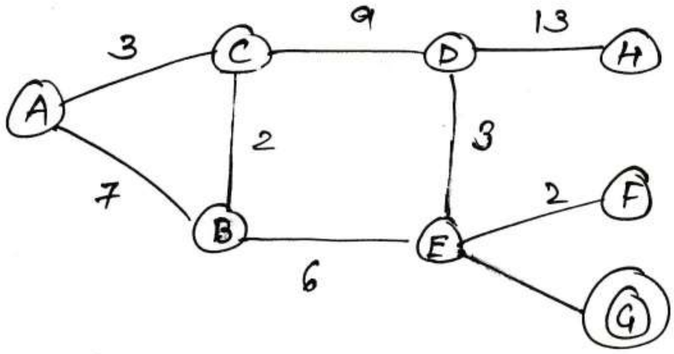
Uniform cost search (UCS)



A → B : 2

A → C : 3

A → B → D path cost : 3



A → C → B → E → G
3 2 6 1

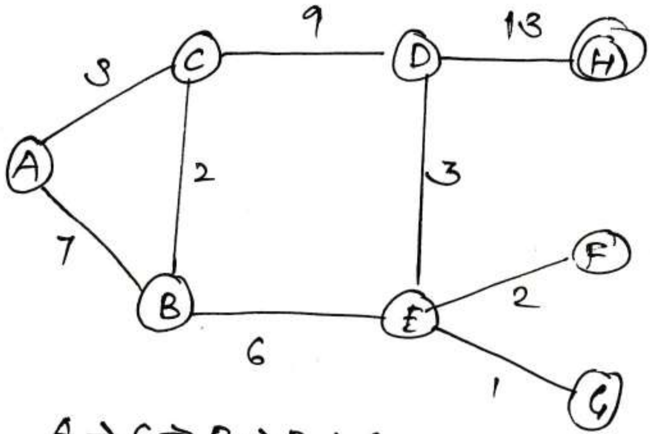
path cost : 12

Advantages :-

1. Mostly gives optimal solutions

Disadvantages

1. Can get stuck itself an infinite loop.



A → C → B → E → G → E → F → E → D → H

path cost : 33

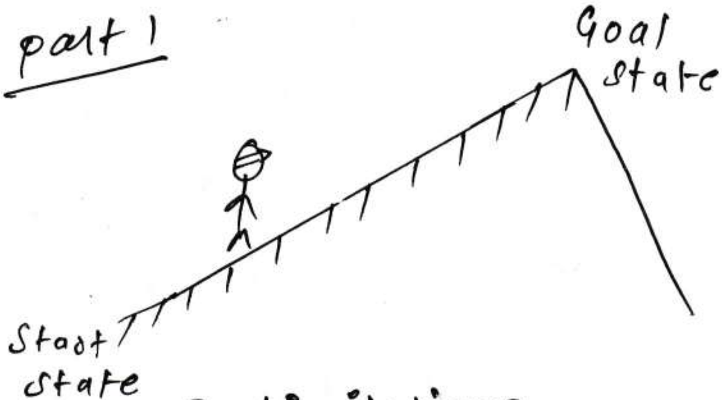
Features

1. Reaches to Goal node
2. Back Hacking is possible
3. Uses Priority Queue
4. Node Expansion is based on path cost

Hill climbing Algorithm

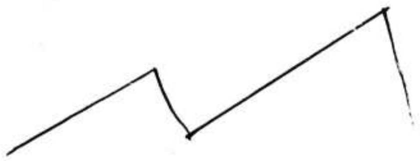
1. Introduction

part 1

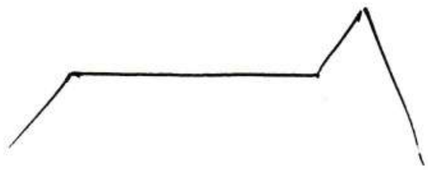


2. Limitations

i) Local Maxima



ii) Plateau:-



iii) Ridge

3. Solutions

4. Important Points

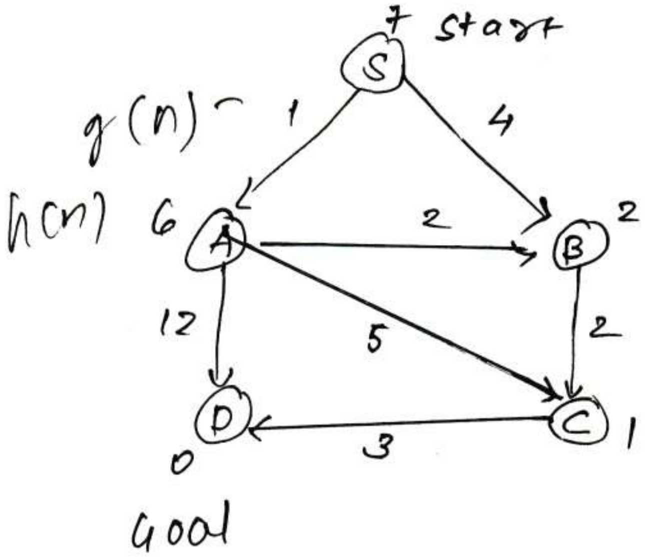
i) Local search Algorithms

ii) Memory Efficient

iii) Used for problem with multiple solutions

part 2

A* Algorithm



$S \rightarrow B \rightarrow C$

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

$= 6 + 1$

$= 7$

A* Algorithm

$S \rightarrow A$

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

$= 1 + 6$

$= 7$

$S \rightarrow B$

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

$= 4 + 2$

$= 6$

$S \rightarrow A = 7$

$\times$ $S \rightarrow B \Rightarrow 6$

$S \rightarrow B \rightarrow C = 7$

$S \rightarrow B \rightarrow C$

$f(n) = g(n) + h(n)$
$= 6 + 1$
$= 7$

$S \rightarrow A \rightarrow B$ $f(n) = g(n) + h(n)$
$= 3 + 2 = 5$

$S \rightarrow A \rightarrow \cancel{B}C$

$f(n) = g(n) + h(n)$
$= 6 + 1$
$= 7$

$S \rightarrow A \rightarrow D$

$f(n) = g(n) + h(n)$
$= 13 + 0$
$= 13$

$S \rightarrow A \rightarrow B \rightarrow C$

$f(n) = g(n) + h(n)$
$= 5 + 1$
$= 6$

$S \rightarrow B \rightarrow C \rightarrow D$

$f(n) = g(n) + h(n)$
$= 9 + 0$
$= 9$

Final path $S \rightarrow A \rightarrow B \rightarrow C \rightarrow D$

× $S \rightarrow A \Rightarrow 7$
× $S \rightarrow B \Rightarrow 6$
$S \rightarrow B \rightarrow C : 7$
× $S \rightarrow A \rightarrow B = 5$
$S \rightarrow A \rightarrow C = 7$
× $S \rightarrow A \rightarrow D \rightarrow 13$
$S \rightarrow A \rightarrow B \rightarrow C : 6$
$\boxed{S \rightarrow A \rightarrow B \rightarrow C \rightarrow D = 8}$
× $S \rightarrow B \rightarrow C \rightarrow D \Rightarrow 9$

$S \rightarrow A \rightarrow B \rightarrow C \rightarrow D$

$f(n) = g(n) + h(n)$
$= 8 + 0$
$= 8$

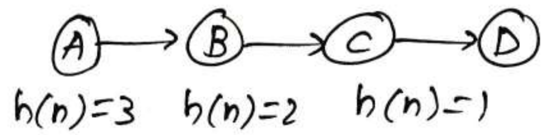
Heuristic search & Function

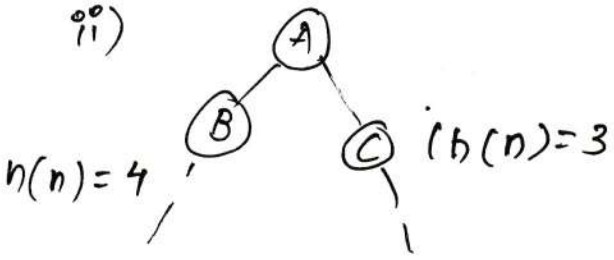
1. What is Heuristic search?

→ It tries to optimize a problem with the use of heuristic function.

2. What is Heuristic Function?

→ It is a function that gives an Estimation of the cost of getting from node n to the goal node.

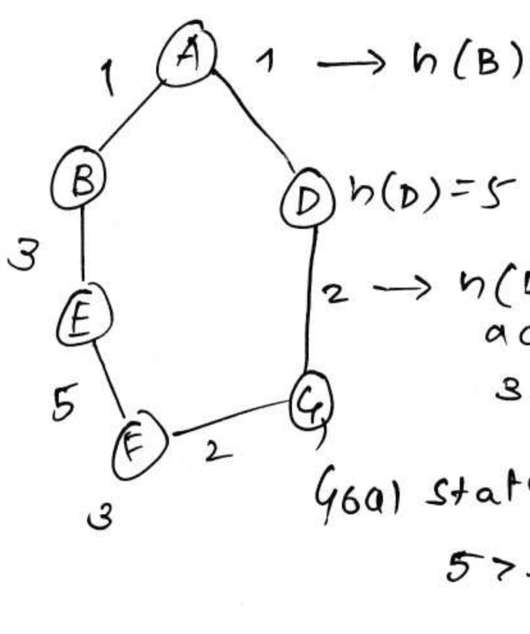
i) 

ii) 

3) Types of Heuristic function

i) Admissible $(h(n) \le h'(n))$

ii) Non-Admissible $(h(n) > h'(n))$



→ h(B) = 3, h(D) = 5

→ h(B) = 3

actual = 11

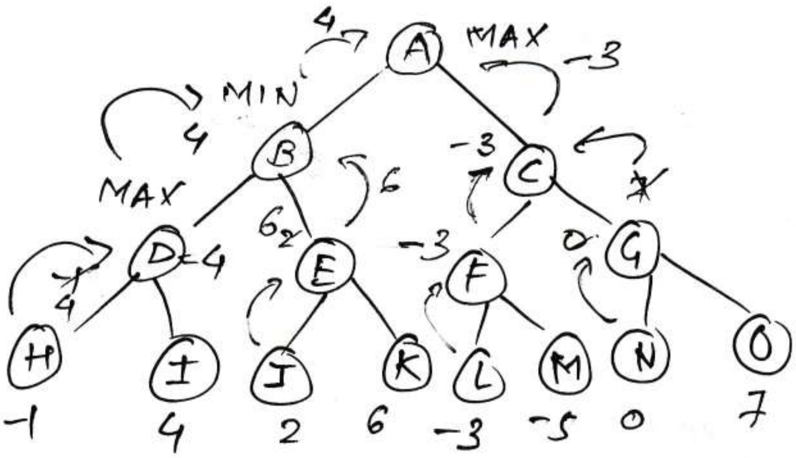
3 < 11

h(D) = 5

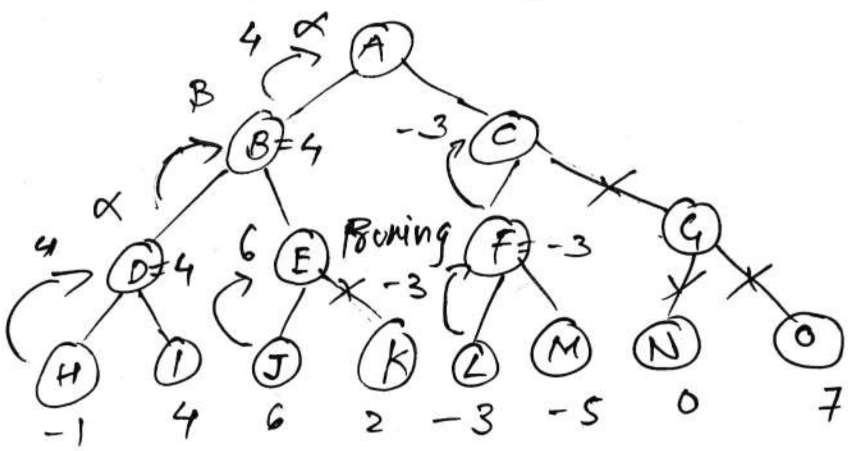
actual = 3

5 > 3

Min-Max Algorithm



Alpha-Beta pruning



PEAS FOR WUMPUS IN AI

Environment

A 4*4 grid of rooms.

- The agent initially in room square [1,1], Facing toward the right
- Location of wumpus and gold are choosen randomly except the first square [1,1]

Actions/Actuators

- The agent can move Forward, TurnLeft, by 90° or TurnRight by 90°
- The action Grab can be used to pick up the gold if its there in same square as agent
- The action shoot can be used to fire an fire an arrow in a straight line in the direction the agent is Facing
- The action climb can be used to climb out of the cave, but only from square [1,1]

Sensor

The Agent has five sensor :-

- In square containing the wumpus and in the directly (not diagonally) adjacent square, the agent will perceive a stench.
- In square directly adjacent to a pit, the agent will ~~pe~~ perceives as Breeze.
- In the square where the gold is, the agent will perceive Glitter.
- When an agent walks into a wall, it will perceive a Bump.

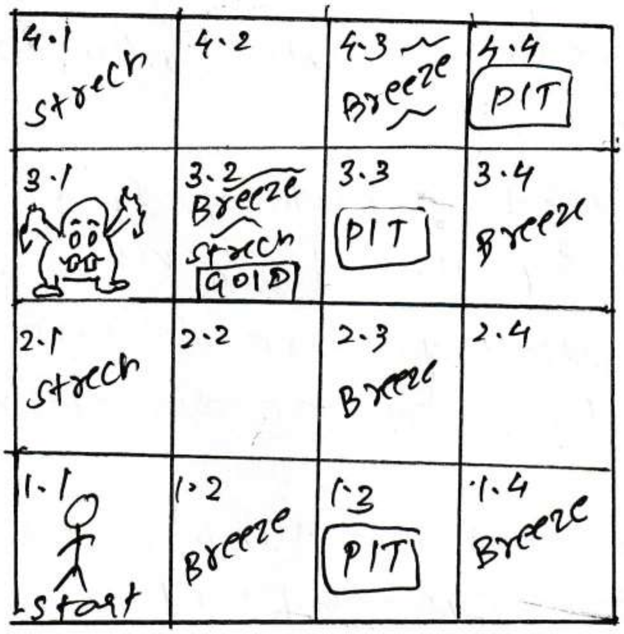
WUMPUS WORLD

1. Introduction:-

* wumpus, strech, Agent, pit, Breeze, gold, Arrow.

Awards and Punishment

i) +1000points if Agent comes out with gold.
ii) -1 point on agents Every Action.
iii) -10 point if arrow is used.
iv) -1000point if agent dies./or eaten by wumpus or falling into pit



P.E.A.S PROPERTIES

performance measure

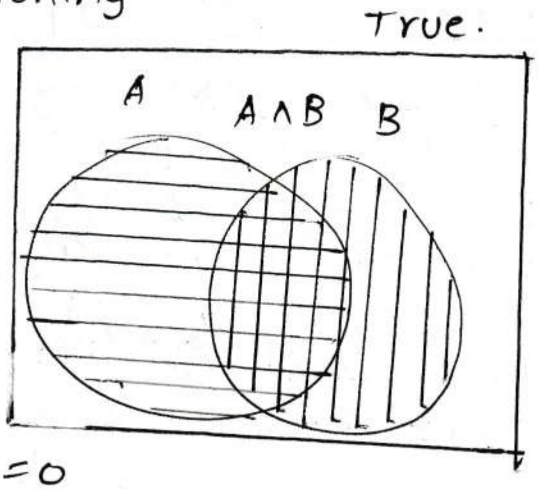
i) +100

Environment: i) Empty Room
ii) Room with WUMPUS
iii) strechy rooms
iv) Breezy rooms
v) Room with Gold
vi) Arrow

sensors i) camera, ii) odor sensors iii) Audio sensors

Effectors: i) motor to move R, L. ii) Robotic Arm to grab gold
iii) Robotic mechanism to shoot arrow.

probablistic Reasoning



Axioms of probability

For any propositions A, B

- $0 \leq P(A) \leq 1$
- $P(true) = 1$ and $P(false) = 0$
- $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$

Multivalued Random variable

suppose A can take on more than 2 values

A is a random variable with arity K if it can takes on exactly one value out of $\{v_1, v_2 \ldots v_K\}$

. Thus.

$$P(A = v_i \wedge A = v_j) = 0 \text{ if } i \neq j$$

$$P(A = v_1 \vee A = v_2 \vee A = v_K) = 1$$

$$\sum_{j=1}^{K} P(A = v_j) = 1$$

conditional probability

conditional probability

$$P(a|b) = \frac{P(a \wedge b)}{P(b)} \quad P(b) > 0$$

product Rule gives an alternative

$$P(a \wedge b) = P(a|b)P(b) = P(b|a)P(a)$$

general version holds for whole distribution

$$P(weather, cavity) = P(Weather | cavity) P(cavity)$$

Conditional probability

chain rule is derived by successive application of product rule:-

$$P(X_1, \ldots X_n) = P(X_1, \cdots X_{n-1})\, P(X_n \mid X_1, \ldots\ldots X_{n-1})$$

$$= P(X_1, \ldots, X_{n-2})\, P(X_{n-1} \mid X_1 \ldots, X_{n-2})\, P(X_n \mid X_1, \ldots X_{n-1})$$

$$= P(X_1)\, P(X_2 \mid X_1)\, P(X_3 \mid X_1, X_2) \ldots P(X_n \mid X_1, \ldots, X_{n-1})$$

or $\boxed{\prod_{i=1 \text{ to } n} P(X_i \mid X_1, \ldots X_{i-1})}$

Inference by enumeration

start with the join probability distribution

	toothache		¬toothache	
	catch	¬catch	catch	¬catch
cavity	0.108	0.012	0.072	0.008
¬cavity	0.016	0.064	0.144	0.576

φ sum the atomic events where it is true $P(\varphi) = \Sigma_{\omega : \omega \models \varphi} P(\omega)$

$P(\text{toothache}) = 0.108 + 0.012 + 0.016$

conditional probabilities

$$P(\neg\text{cavity} \mid \text{toothache}) = \frac{P(\neg\text{cavity} \wedge \text{toothache})}{P(\text{toothache})}$$

$$= \frac{0.016 + 0.064}{0.108 + 0.012 + 0.016 + 0.064} = 0.4$$

in class exercise

Given the joint distribution shown below and definition $P(a|b) = \frac{P(a \wedge b)}{P(b)}$

$P(\text{cavity} = \text{True}) = ?$

$P(\text{weather} = \text{sunny})?$

$P(\text{cavity} = \text{True} \mid \text{weather} = \text{sunny})$

$P(\text{weather}, \cancel{\text{cap}}\ \text{cavity}) = P(\text{weather} \mid \text{cavity})\ P(\text{cavity})$

weather=snow	sunny	rainy	cloudy
cavity = true	0.144	0.02 0.016	0.02
cavity = False	0.576	0.08 0.064	0.08

$\boxed{\text{independence}}$

A and B are independent iff

$\boxed{P(A|B) = P(A) \mid P(B) \mid \text{or } P(B|A)}$

cavity
Tootache catch
weather

decomposes into $\longrightarrow$

cavity
Toothache catch

weather

$P(\text{Toothache}, \text{catch}, \cancel{\text{catv}}\ \text{cavity}, \text{weather})$

$= P(\text{Tootache}, \text{catch}, \text{cavity})\ P(\text{weather})$

Bayes' Rule:

product rule $P(a \wedge b) = P(a|b)P(b) = P(b|a)P(a)$

Bayes' Rule $P(a|b) = P(b|a)\dfrac{P(a)}{P(b)}$

Distribution form

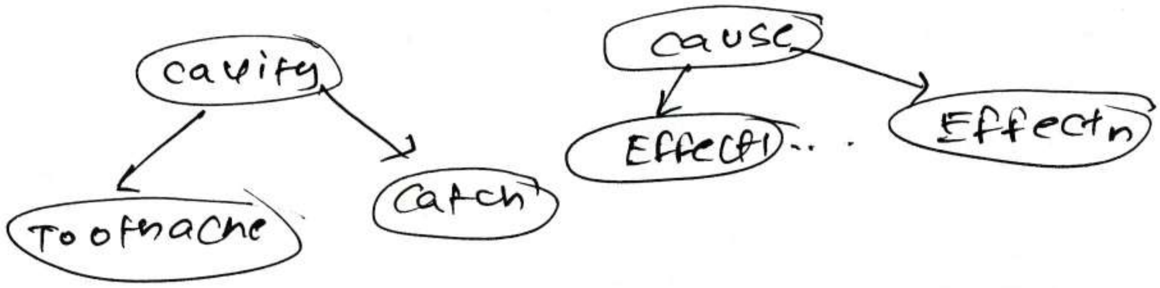
$$P(Y|X) = P(X|Y)\frac{P(Y)}{P(X)} = \alpha P(X|Y)P(Y)$$

$P(\text{Cavity} | \text{toothache} \wedge \text{catch})$

$= \alpha P(\text{toothache} \wedge \text{catch} | \text{Cavity}) P(\text{cavity})$

$= \alpha P(\text{toothache} | \text{cavity}) P(\text{catch} | \text{cavity}) P(\text{cavity})$

navive Bayes



conditionally independent

$P(\text{catch} | \text{Toothache}, \text{cavity}) = P(\text{catch} | \text{cavity})$

$P(\text{Toothache}, \text{catch} | \text{cativity}) = P(\text{Toothache} | \text{cavity})$
$P(\text{catch} | \text{cavity})$

Bayesian network

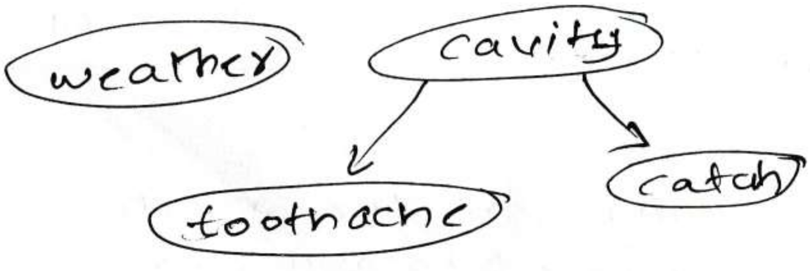
A simple, graph, graphical notation for conditional independence assertions and hence for compact specification of full joint distributions.

Syntax

- a set of nodes, one per variable.
- a directed, acyclic graph (links = "directly influences")
- a conditional distribution for each node given its parents : $P(x_i \mid parents(x_i))$

Example

Topology of network encodes conditional independence assertions.



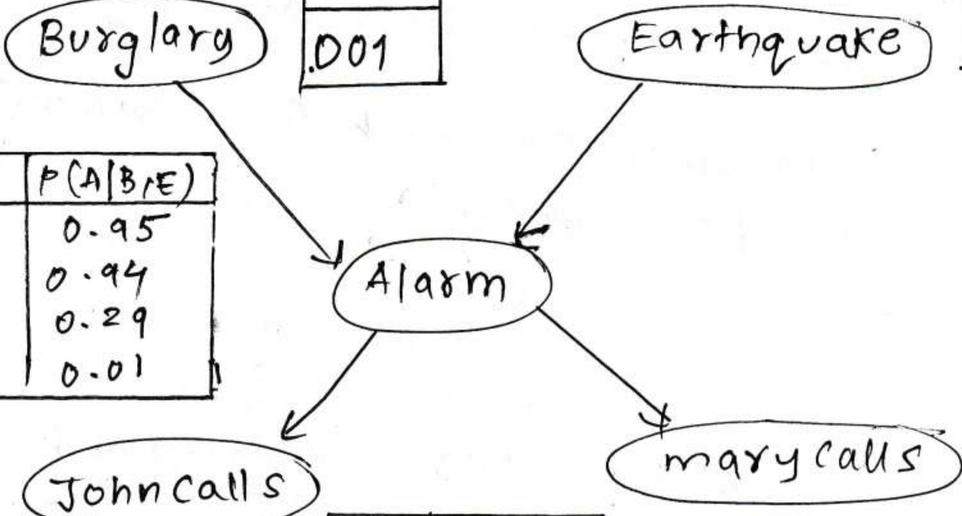
Example

"I'm at work, neighbor John calls to say my alarm is ringing, but neighbor Mary doesn't call. Sometimes it's set off by minor earthquakes. Is there a burglar"

Variables: Burglary, Earthquake, Alarm, JohnCalls, MaryCalls.

Network topology reflects "casual" knowledges:

→ A burglar can set the alarm off
→ An earthquake can set the alarm off.
→ The alarm can cause Marry to call.
→ The alarm can cause John to call.



P(B)
.001

P(E)
0.002

B	E	P(A|B,E)
T	T	0.95
T	F	0.94
F	T	0.29
F	F	0.01

A	P(J|A)
T	0.90
F	0.05

A	P(M|A)
T	0.70
F	0.01

Q1) what is the probability that the alarm has sounded but neither burglary nor an earthquake has occured and both John and marry call.

$$P(j \wedge m \wedge a \neg b \wedge \neg e) = P(j|a)P(m|a)P(a|\neg b, \neg e)P(\neg b)P(\neg e)$$

$$= 0.90 \times 0.70 + 0.001 \times 0.999 \times 0.998$$

$$= 0.00062$$

what is the probability that John call?

$= P(j/a)P(a) + P(j|\neg a)P(\neg a)$

$= P(j|a)\{P(a|b,e) * P(b,e) + P(a|\neg b,e) * P(\neg b,e) +$
$P(a|b,\neg e) * P(b,\neg e) + P(a|\neg b,\neg e) * P(\neg b,\neg e)\}$
$+ P(j|\neg a)\{P(\neg a|b,e) * P(b,e) + P(\neg a|\neg b,e) *$
$P(\neg b,e) + P(\neg a|b,\neg e) * P(b,\neg e) + P(\neg a|\neg b,\neg e) *$
$P(\neg b,\neg e)\}$

$= 0.90 * 0.00252 + 0.05 * 0.9974$

$= 0.0521$

Given :-

Let Event 'A' denote falling sick due to flu

Let Event 'B' denote falling sick due to measles and no other disease

'R' Event Rashes

$P(A) = ?$

$P(A) = 0.9$ $\quad P(R/B) = 0.95$ $\quad = \frac{0.072}{0.167}$

$P(B) = 0.1$

$P(R/A) = 0.08$ $\quad \boxed{P(A/R) = 0.43}$

$P(A/R)$ $\quad = \frac{0.08 \times 0.9}{0.95 \times 0.1 + 0.08 \times 0.9}$

$$P(A/R) = \frac{P(R/A) \cdot P(A)}{P(R/B) \cdot P(B) + P(R/A) \cdot P(A)}$$

$P(A/R) = 0.072 / 0.072 + 0.095$

All students are smart

$\forall x : ((students(x) \longrightarrow smart(x))$

There Exist a student

$\exists x : student(x)$

There Exist a smart student

$\exists (x) : student(X) \cancel{\wedge} \wedge smart(x))$

Every student Loves some student

$\exists (x) : student(x)\ \cancel{\wedge\ Loves} \longrightarrow \exists y (student(y) \wedge loves(x,y))$

There is a student who is loved by Ever other student

$\exists (x) : student(x) \Rightarrow \exists (y) : (student(y) \wedge \neg (x = y) \rightarrow Loves(y, x))$

Bill is a student

$\cancel{\exists (x)}$: student (Bill)

Bill takes either Analysis or Geometry (not both)

takes (Bill, Analysis) $\xleftrightarrow{}$ $\neg$ takes (Bill, Geometry)

Bill takes Analysis or Geometry or both

takes (Bill, Analysis) $\vee$ takes (Bill, Geometry)

Bill takes Analysis and Geometry

~~takes (Analysis~~ takes (Bill, Analysis) $\wedge$ takes(Bill, Geometry)

FIRST ORDER PREDICATE LOGIC (FOPL)

1. All boys like cricket

$\forall(x): boys(x) \longrightarrow like(x, criket)$

2. some girls hate Football

$\cancel{\exists(x): girls(x) \longrightarrow hate(x, Football)}$ (wrong)

$\exists(x): girls(x) \wedge hate(x, Football)$

2. Some boys like football

$\exists(x): boys(x) \wedge like(x, football)$

3. All girls love pink

$\forall(x): girls(x) \longrightarrow love(x, pink)$

4. Every person who buys a policy is smart

$\cancel{\exists(x)}\ \forall x \forall y: Person(x) \wedge policy(y) \wedge \cancel{by}\ boys(x, y) \rightarrow smart(x)$

$\exists \cancel{(x, y): person(x}$

Imp. Example

6. No person buys expensive policy

$\cancel{\exists}\ \forall x \forall y: person(x) \wedge policy(y) \wedge \underset{expensive}{\cancel{buys}}(x, y) \longrightarrow$

$\neg buys(x, y$

Imp example

if it is winter, then it will be cool

$$\underset{\text{Action}}{}$$

~~$\exists x,y$: winter(x) $\wedge$ cool(y) $\rightarrow$ willbe(x,y)~~

$\exists x,$~~y~~ $\boxed{: \text{winter}(x) \longrightarrow \text{willbe}(\text{cool}, x)}$

winter(x) $\wedge$ cool(y) $\longrightarrow$ willbe(x,y)

" the car is either at John's house or at Fred's house if car is not at John house, it must be at Freds house

~~$\forall x,$~~ $\exists (x), y/$~~z~~ : Johnshouse(x), fredshouse(z),

~~car $\rightarrow$ isAt(x) $\leftrightarrow$ car $\rightarrow$~~,

isat(car, x) $\veebar$ isat(car, y)

$\neg$(car, x) $\rightarrow$ isat(car, y)

$\exists x, y$: JohnHouse(x), Fredhouse(y): ~~isAt(c~~

[isAt(~~car~~ car, x) $\vee$ isat(car, y)] $\wedge$

$\neg$isat(car, x) $\longrightarrow$ isat(car, y))

Bill does not take Analysis

$\neg take(Bill, Analysis)$

No student loves Bill

$\neg \exists x (student(x) \rightarrow loves(\cancel{Bill}\ x, Bill))$

~~of~~ Bill has at least one sister

$\boxed{\exists x\ sister(x) : sisterof(x, Bill)}$

Bill has no sister.

~~$\forall \exists$~~

$\neg \exists x : sister(x) : sisterof(x, Bill)$

Bill has~~t~~ at most one sister

~~$\forall\ \exists x : Bill\ sister(x)$~~ :

$\forall (x)$: Bill has at the most one sister

$\forall (x), \forall (y) : (sisterof(x, Bill) \wedge sisterof(y, Bill) \Rightarrow x = y)$

Bill has exactly one sister

$\cancel{\forall}(x\ \exists x\ (sisterof(x, Bill) \wedge \forall y (sisterof(y, Bill\ x = y))$

if winter

if

NAME: RUTVIK AVINASH BARBHAI
CSE - CORE - A

REG NO. 225805222
ROLL NO: 039

ARTIFICIAL INTELLIGENCE (AI) ASSIGNMENT

Q1) outline the important milestone for the evolution time line of artificial intelligence. What are the latest introduction into the domain of AI?

A1) MILESTONES IN AI EVOLUTION

1. 1950s-1960s: Birth of AI - Alan Turning proposed the Turning Test (1950), First AI progroms developed (eg: Logic Theorist 1956).

2. 1980s-90s: Rise of expert systems, introduction of neural networks and advacements in machine learning.

3. 2000s-2010s: Deep learning revolution with algorithms like CNNs and RNNs, rise of AI powered applications (Siri, Google Assistant), Alphago defeating human players.

4. 2020s-Present: Generative AI break through (chatGPT, DALLE), advancements in AGI research, AI ethics and regulations gaining importance.

LATEST AI Advancement

- 2024 Highlights
  - open AI's Sora: AI powered video generation.
  - Google's Gemini 1.5: Advonced multimodel AI model.
  - XAI's Grok: Elon Musk's AI chatbot

- Emerging trends
  - Multimodel AI (text, image, video integration), AI driven agents, self improving models, advancement in robotics and automation.

Q② your AI agent is provided with 2 water jugs
Q2) of capacity 6L and 8L respectively. Design the steps for your agent so that it can successfully fill another jug with 4L of water.

→ <u>Given</u>

Jug A : 6L capacity
Jug B: 8L capacity
Goal: measure exactly 4L of water

<u>Rules for AI agent</u>

1. Fill jug A completley:
$(x, y) \longrightarrow (6, y)$ if $x < 6$

2. Fill jug B completley
$(x, y) \longrightarrow (x, 8)$ if $y < 8$

3. Empty the 6L jug on ground:
$(x, y) \longrightarrow (0, y)$ if $x > 0$

4. Empty the 8L jug on ground
$(x, y) \longrightarrow (x, 0)$ if $y > 0$

5. Pour from jug B to jug A until it is Full.
$(x, y) \longrightarrow (6, y - (6 - x))$ if $x + y \geq 6$ and $y > 0$

6. pour water from jug A to jug B until it is Full.
$(x, y) \longrightarrow (x - (8 - y), 8)$ if $x + y \geq 8$, $x > 0$

7. Pour all water from jug A to jug B
$(x, y) \longrightarrow (0, x + y)$ if $x + y \leq 8$ and $x > 0$

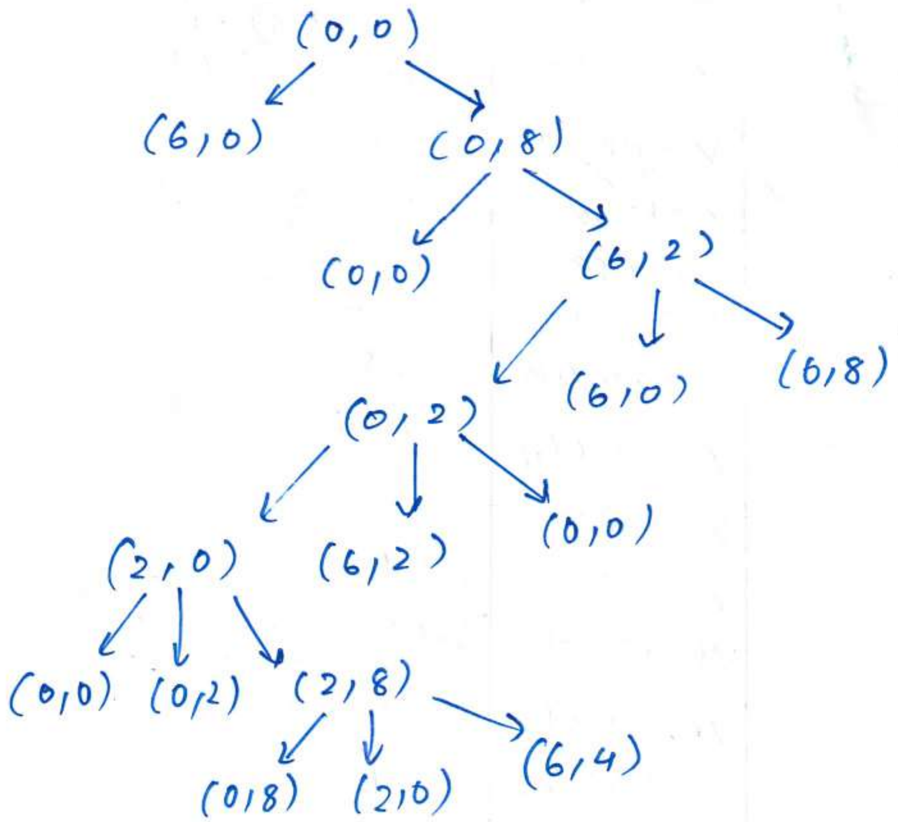
8. pour all water from jug B to jug A.
$(x, y) \longrightarrow (x + y, 0)$ if $x + y \leq 6$ and $y > 0$

9. if jug B has 4L stop
Goal state : $(x, 4)$ where $0 \leq x \leq 6$

roduction for this problem!

jug A (6L)	jug B (8L)	Rule Applied.
0	0	
0	8	2
6	2	5
0	2	3
2	0	8
2	8	2
6	4	5
	Goal state Achieved	

Solution Tree:-



c) Determine which species does a flower with sepal length, 5.2, sepal width 3.1 using KNN algorithm. Assume K=3

$$d = \sqrt{(5.3-5.2)^2 + (3.7-3.1)^2}$$
$$d = \sqrt{(0.01)^2 + (0.36)^2} = 0$$
$$d = \sqrt{0.0001 + 0.1296}$$
$$d = \sqrt{0.1297}$$
$$=$$

sepal length	sepal width
5.2	3.1

sepal length	sepal width	species	Euclidian distance
5.3	3.7	setosa	$\sqrt{(5.2-5.3)^2+(3.7-3.1)^2} = 0.61$
5.1	3.8	setosa	0.71
7.2	3.0	virginica	2.00
5.4	3.4	setosa	~~3.00~~ 0.36
5.1	3.3	setosa	0.22
5.4	3.9	setosa	0.82
7.4	2.8	virginica	2.25
6.1	2.8	versicolor	0.95
7.3	2.9	virginica	2.11
6.0	2.7	versicolor	0.89
5.8	2.8	virginica	0.67
6.3	2.3	versicolor	1.41
5.1	2.5	versicolor	0.61
6.3	2.5	versicolor	1.28
5.5	2.4	versicolor	0.74

(5.1, 3.3) – setosa – 0.22

(5.4, 3.4) – setosa – 0.36

(5.3, 3.7) – setosa – 0.61

All 3 nearest neighbors are setosa.

) Given below are the inital and Final states of an 8 puzzle problem. Find the path and path cost to obtain the solution for this.

g(n) = Depth of ~~mo~~ node

and h(n) = Number of misplaced tiles.)

~~gf~~ f(n) = g(n) + h(n)

5	1	3
2		4
7	6	8

Initial state

4	1	3
5	–	2
7	6	8

Final state

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

5	1	3
2	–	4
7	6	8

g = 0 h = 3 f = 3

INITIAL STATE

Depth 1

5	1	3
–	2	4
7	6	8

g = 1 h = 3
f = 4

5	1	3
2	4	–
7	6	8

g = 1, h = 3
f = 4

5	–	3
2	1	4
7	6	8

g = 1 h = 4
f = 5

Depth 2

5	1	–
2	4	3
7	6	8

g = 2, h = 4
f = 6

5	–	1
2	4	3
7	6	8

g = 3, h = 5
f = 8 Depth 3

5	4	1
2	–	3
7	6	8

g = 4, h = 5 ~~8~~
f = 9 Depth 4

5	4	1
–	2	3
7	6	8

g = 5, h = 5
f = 10

–	4	1
5	2	3
7	6	8

g = 6, h = 4
f = 10

4	–	1
5	2	3
7	6	8

g = 7, h = 3
F = 10

4	1	–
5	2	3
7	6	8

g = 8,
h = 2 f = 10

GOAL STATE

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

$g=8,\ h=2$ $\quad$ $g=9,\ h=1$ $\quad$ $g=10,\ h=0$

$f=10$ $\quad$ $f=10$ $\quad$ $\boxed{f=10}$

$\boxed{\text{Final path cost} = 10\ [\text{Ans}]}$

Q4) Justify the need for conversion to ~~clasu~~ clausal from in KR. convert the following expression into ~~clas~~ clausal from along with details step

$$\exists x \forall y \forall u (\exists v P(f(x), v) \Longleftrightarrow Q(g(y, u) \,\&\, R(x, v)))$$

A4) In KR (Knowledge representation), it is often necessary to convert logical expression into clausal form, because:

1. Standardization.
2. Resolution based Reasoning
3. Eliminating ambiguity
4. Efficiency in computation.
5. Simplifies logical processing.

Given expression:-

$$\exists x \forall y \forall u (\exists v P(F(x), v) \Longleftrightarrow Q(g(y, u) \,\&\, R(x, v))$$

$$\cancel{Q(g(y,u)) \longleftrightarrow R(x,v))}$$

Step 1:- Eliminate Biconditional

$$\exists x \forall y \forall u (\exists v P(f(x), v) \longrightarrow (Q(g(y, u)) \wedge R(x, v)) \wedge$$
$$Q(g(y, u)) \wedge R(x, v) \longrightarrow \exists v P(f(x, v)))$$

tep 2 : Eliminate Implications:

$\exists x \forall y \forall u (\neg \exists u P(f(x), v) \vee (Q(g(y,u)) \wedge R(x,v)) \wedge$
$(\neg (Q(g(y,u)) \wedge R(x,v)) \vee \exists u P(f(x), v)))$

Step 3:- move quantifiers outward.

$\exists x \forall y \forall u \forall v ((\neg (f(x), v) \vee Q(g(y,u)) \wedge \neg P(f(x), v)$
$\vee R(x,v)) \wedge (\neg Q(g(y,u)) \vee \neg R(x,v) \vee P(f(x),v)))$

Step 4: Distribute quantifiers

$\exists x \forall y \forall u \forall v ((\neg P(f(x), v) \vee Q(g(y,u)) \wedge \neg P(f(x),v)$
$\vee R(x,v)) \wedge (\neg Q(g(y,u)) \vee \neg R(x,v) \vee P(f(x),v)))$

Step 5: skolemigation

$\forall y \forall u \forall v ((\neg P(F(x), v) \vee Q(g(y,u)) \wedge \neg P(f(x),$
$v) \vee R(c,v)) \wedge (\neg Q(g(y,v) \vee \neg R(c,u) \vee$
$P(f(c), v)))$

Step 6: Distribute conjunction over disjunctions

$\forall y \forall u \forall v (\neg P(f(c), v) \vee Q(g(y,u))) \forall y \forall u \forall v$
$(\neg P(f(c), v) \vee R(c,v)) \forall y \forall u \forall v (\neg Q(g(y,u))$
$\vee \neg R(c,v) \vee P(f(c), v))$

Step 7 : convert to clausal form

1. $\neg P(f(c), v) \vee Q(g(y,u))$
2. $\neg P(f(c), v) \vee R(c,v)$
3. $\neg Q(g(y,u)) \vee \neg R(c,v) \vee P(f(c),v)$

$\forall x, y \; (Add(x, y) \rightarrow \text{greater than}(x, 0))$

For Every X and Y, ~~Add x.~~ if $x+y > 0$ then ~~X is greater than~~ / add $x+y$

$(\neg A \vee \neg B \vee C) \wedge (\neg A \vee B \vee \neg C)$

step 1 Eliminate biconditionals and implications

Eliminate $\rightarrow$ replacing $\alpha \rightarrow \beta$ with $\neg\alpha \vee \beta$.

Eliminate $\leftrightarrow$ replacing $\alpha \leftrightarrow \beta$ with $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$

$\neg(\forall x . p) \equiv \exists x \neg p$

$\neg(\exists x \; p) \equiv \forall x \; \neg p$

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

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

$\neg\neg\alpha \equiv \alpha$

step 3 Skolemization

$\forall x \; person(x) \Rightarrow \exists y \; Heart(y) \wedge Has(x, y)$

$\forall x \; Person(x) \Rightarrow Heart(H(x)) \wedge Has(x, H(x))$

$H \rightarrow$ skolem fn

$\forall x \; person(x)$ becomes $person(x)$

$(\alpha \wedge \beta) \vee \gamma \equiv (\alpha \vee \gamma) \wedge (\beta \vee \gamma)$