Q.No.1. a. *Define Precision, Accuracy and Repeatability. For which type of robot is overall precision uniform and for which robots vertical precision uniform.* *05*

Ans: **Precision:** It is the measure of spatial resolution with which the tool can be positioned within the work space envelope.

E.g. the tool tip is positioned at position A and the next closest position that it can be moved is B, then precision along that direction is the distance between, point A and B.

Accuracy: It is the measure of ability of robot to place the tool tip at an arbitrarily prescribed location in work envelope.

Repeatability is always < accuracy

Repeatability: It is the measure of ability of the robot to position the tool tip in the same place repeatedly or the ability of robot to do the same job again and again repeatedly and then come back to its original position.

For **Cartesian coordinate robot** overall precision is uniform.

For **Cartesian** coordinate robot, **cylindrical** robot and **SCARA** robot vertical precision is uniform.

b. *Explain structure of agent that keeps track of the world.* *05*

Ans: **Agents that keep track of the world:**

The simple reflex agent will work only if the correct decision can be made on the basis of the current percept. Consider an example of a driver who looks in the rear-view mirror to check on the locations of nearby vehicles.

When the driver is not looking in the mirror, the vehicle in the next lane are invisible i.e. The state in which they are present and absent are indistinguishable is because the sensors do not provide access to the complete state of the world.

In the above example the agent need to maintain some internal stare information in order to distinguish between world states that generate the same perceptual input, but different action are appropriate in the two state.

Updating this internal state information requires two kinds of knowledge to be encoded in the agent program.

First, some information about how the world evolves independently of the agent is needed. For example, that an overtaking car generally will be closer behind than it was a moment ago.

Second, some information about how the agent' own actions affect the world. For example when the agent changes lanes to the right, there is gap in the lane it was in before, or that after driving for five minutes ago.

The figure below gives the structures of the reflex agent, showing the current percept is combined with the old internal state to generate the updated description of the current state.

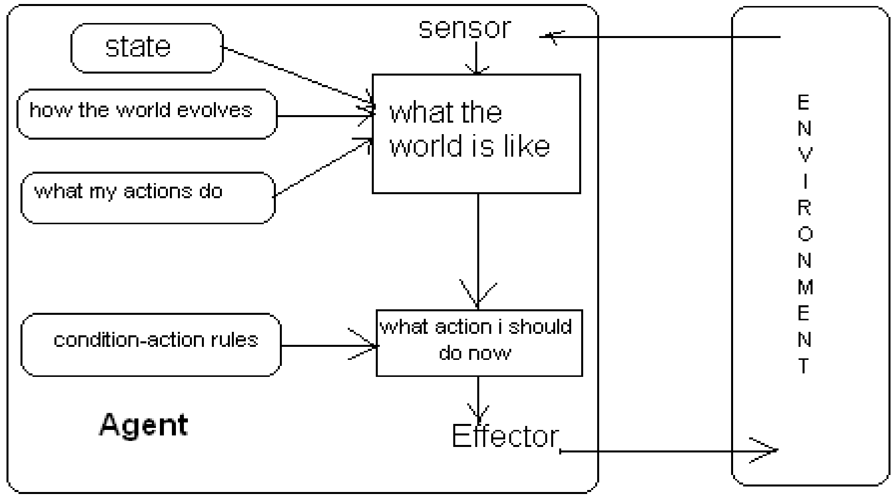


Fig : A reflex agent with internal state.

c. *Draw joint diagram, link diagram. Define joint angle, joint distance, link length and link twist angle.* *05*

Ans: **Joint angle(θ_k):-** It is rotation about Z^{k-1} needed to make axis X^{k-1} parallel with axis X^k.
Joint distance(d_k):- It is translation along Z^{k-1} needed to make axis X^{k-1} intersect with axis X^k.
Link twist angle(α_k):-It is rotation about X^{k-1} or X^k needed to make axis Z^{k-1} parallel with axis Z^k.
Link length(a_k):- It is translation along X^{k-1} or X^k needed to make axis Z^{k-1} intersect

with axis Z^k.

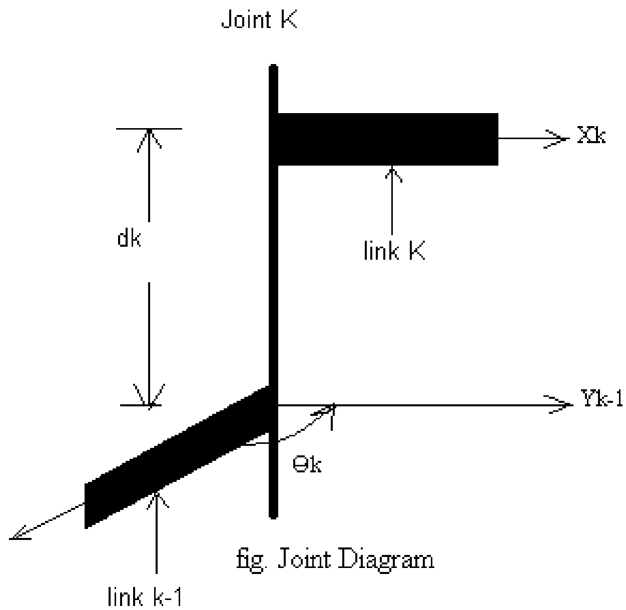


fig. Joint Diagram

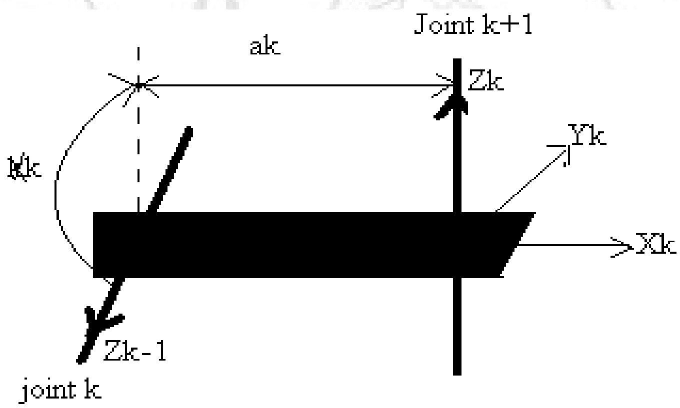


fig. Link Diagram

d. List the sensors used for reactive robot and explain the GPS system. *05*

Ans: **Sensors used for reactive robot:-**

- Active Sensors.
- Passive Sensors.
- Logical Sensors.
- Proprioceptive Sensors.
- Proximity Sensors.

GPS system:-

GPS (Global Positioning System) is becoming more common on robots, especially those used to automate farm equipment (agriculture). GPS system work by receiving signals from satellites orbiting the earth. The receiver triangulates itself relative to four GPS satellites, computing its position in terms of longitude, latitude, altitude and change in time. GPS isn't a Proprioceptive sensor since the robot must receive signals from the satellites, external to robot. However, they are not exteroceptive sensors either, since the robot isn't computing its position relative to its environment.

In DGPS, two GPS receivers are used. One remains stationary, while the other is put on the robot. If the two receivers are observing the same satellites, then any sudden change in position on the stationary " base" receiver is due to the induced error and can be subtracted from the readings at the robot GPS.

Q.No 2. a. What are the components of problem formulation? Hence formulate problem for vacuum cleaner and 8-puzzle game. *10*

Ans: A problem is really a collection of information that the agent will use to decide what to do. We will begin by specifying the information needed to define a single state problem. We have seen that the basic elements of a problem definition are the states and actions. The components of problem formulation are as follows:

- **Initial state:** The initial state that the agent knows itself to be in.
- **Possible actions:** The set of possible actions available to the agent. Also referred as successor function. The term operator is used to denote the description of an action in terms of which state will be reached by carrying out the action in a particular state. Together, these define the state space of the problem: the set of all states reachable from the initial state by any sequence of actions. A path in the state space is simply any sequence of actions leading from one state to another.
- **Goal state:** The goal state, which the agent can apply to a single state description to determine if it is a goal state. Sometimes there is an explicit set of possible goal states, and the test simply checks to see if we have reached one of them.
- **Path cost:** A path cost, function is a function that assigns a cost to a path. In all cases we will consider, the cost of a path is the sum of the costs of the individual actions along the path. The path cost function is often denoted by 'g.'

Problem formulation for vacuum cleaner:

1. States: The agent is one of two locations, each of which might or might not contain dirt. Thus there are $2x2^2 = 8$ possible states.

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

3. Successor function: This generates the legal states that result from trying the three actions (Left, Right and Suck or Clean).

4. Goal state: This checks whether all the squares are clean.

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

Problem formulation for 8-puzzle game:

The 8-puzzle problem consists of a 3x3 board with eight numbered tiles and a blank space. A tile adjacent to the blank space can slide into the space. The object is to reach a specified goal state. The standard formulation is as follows:

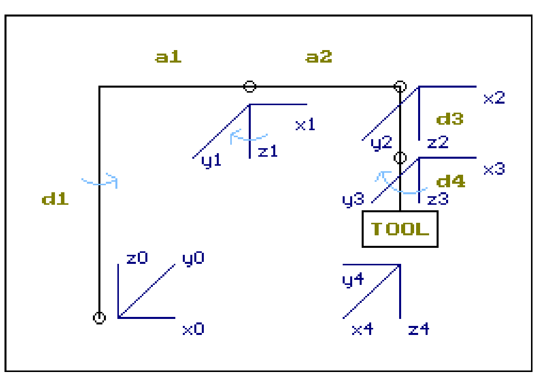
1. **States:** A state description specifies the location of each of the eight tiles and the blank in one of the nine squares.
2. **Initial state:** Any state can be designated as the initial state.
3. **Successor function:** This generates the legal states that result from trying the four actions (blank moves Left, Right, Up or Down).
4. **Goal state:** This checks whether the state matches the goal configuration.
5. **Path cost:** each step costs 1, so the path cost is the number of steps in the path.

Q.No. 2. b *Find the inverse kinematics solution of 4-axis SCARA Robot.* 10

Ans: The inverse kinematics solution of 4-axis SCARA Robot consists of following steps:

Step 1. Development of LCD:

The Link Co-ordinate Diagram for a SCARA Robot is as follows



Step 2:- Extraction of Joint Angles:-

i : - Arm Matrix Equation of 4-Axis SCARA robot (o/p of DK Analysis) is given by,

$$T_{Base}^{Tool}(q) = T_0^4(q) = \begin{pmatrix} c1\text{-}2\text{-}4 & s1\text{-}2\text{-}4 & 0 & a1c1+a2c1\text{-}2 \\ s1\text{-}2\text{-}4 & -c1\text{-}2\text{-}4 & 0 & a1s1+a2s1\text{-}2 \\ 0 & 0 & -1 & d1\text{-}q3\text{-}d4 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{——} (1)$$

$$\text{Standard HCTM Equation} = \begin{pmatrix} R11 & R12 & R13 & P1 \\ R21 & R22 & R23 & P2 \\ R31 & R32 & R33 & P3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{——} (2)$$

By Comparing Eq.1 & Eq.2

R13=0	P1=a1c1+a2c1-2
R23=0	P2=a1s1+a2s1-2
R33= -1	P3=d1-q3-d4

ii: TCV (w) of SCARA Robot :-

$$w(q) = \begin{pmatrix} w1 \\ \hline w2 \end{pmatrix} = \begin{pmatrix} w1 \\ w2 \\ w3 \\ w4 \\ w5 \\ w6 \end{pmatrix} = \begin{pmatrix} P1 \\ P2 \\ P3 \\ \exp(q4/\pi).R13 \\ \exp(q4/\pi).R23 \\ \exp(q4/\pi).R33 \end{pmatrix}$$

$$W(q) = \begin{pmatrix} a1c1+a2c1\text{-}2 \\ a1s1+a2s1\text{-}2 \\ d1\text{-}q3\text{-}d4 \\ 0 \\ 0 \\ -\exp(q4/\pi) \end{pmatrix}$$

This Equation gives TCV of SCARA Robot

Output of I.K analysis:

1) Extraction of base angle (q1) =

From the first component of TCV,

w1 =a1c1+a2c1-2

=a1c1+a2(c1c2+s1s2)

=a1c1+a2c1c2+a2s1s2

=c1(a1+a2c2)+s1(a2s2) --(3)

From the second component of TCV,

w2 =a1s1+a2s1-2

=a1s1+a2(s1c2-c1s2)

=a1s1+a2s1c2-a2c1s2

=s1(a1+a2c2)+c1(-a2s2) --(4)

Putting Eq.3 & Eq.4 in Matrix form,

$$\begin{pmatrix} w1 \\ w2 \end{pmatrix} = \begin{pmatrix} a1+a2c2 & a2s2 \\ -a2s2 & a1+a2c2 \end{pmatrix} \begin{pmatrix} c1 \\ s1 \end{pmatrix}$$

$$\begin{pmatrix} c1 \\ s1 \end{pmatrix} = \begin{pmatrix} a1+a2c2 & a2s2 \\ -a2s2 & a1+a2c2 \end{pmatrix}^{-1} \begin{pmatrix} w1 \\ w2 \end{pmatrix}$$

$$\begin{pmatrix} c1 \\ s1 \end{pmatrix} = \begin{pmatrix} a1+a2c2 / (w1^2+w2^2) & -a2s2/ (w1^2+w2^2) \\ a2s2/ (w1^2+w2^2) & a1+a2c2/ (w1^2+w2^2) \end{pmatrix} \begin{pmatrix} w1 \\ w2 \end{pmatrix}$$

$c1= (a1+a2c2)w1 + (-a2c2)w2 / (w1^2+w2^2)$

$s1= (a2s2)w1 + (a1+a2c2)w2 / (w1^2+w2^2)$

$$\mathbf{q1 = \underline{+}\,arc\ tan \frac{(a2s2)w1 + (a1+a2c2)w2}{(a1+a2c2)w1 - a2c2w2}}$$

2) Extraction of elbow angle (q2) =
From the first & second component of TCV,

$$(w1^2+w2^2) = (a1c1+a2c1\text{-}2)^2 + (a1s1+a2s1\text{-}2)^2$$

$$= a1^2c1^2 + a2^2c1\text{-}2^2 + 2a1c1a2c1\text{-}2 + a1^2s1^2 + a2^2s1\text{-}2^2 + 2a1s1a2s1\text{-}2$$

$$= a1^2 + a2^2 + 2a1a2(c1c1\text{-}2 + s1s1\text{-}2)$$

$$= a1^2 + a2^2 + 2a1a2(c2)$$

$$c2 = ((w1^2+w2^2) - a1^2 - a2^2) / (2a1a2)$$

$$q2 = \pm \arccos \frac{((w1^2+w2^2) - a1^2 - a2^2)}{(2a1a2)}$$

3) Extraction of Vertical Extension angle (q3) =
From the Third component of TCV,

$$w3 = d1 - q3 - d4$$

$$q3 = d1 - d4 - w3$$

4) Extraction of Tool Roll angle (q4) =
From the Last component of TCV,

$$w6 = -\exp(q4/\pi)$$

taking natural log on both side,

$$\ln|w6| = -\ln\{\exp(q4/\pi)\}$$

$$= -\ln\{e^{(q4/\pi)}\}$$

$$q4 = -\pi \ln|w6|$$

$$q4 = \pi \ln|-w6|$$

$$q4 = \pi \ln|w6|$$

Q.No. 3.a. Explain with example Baye's Belief network & simple inference in belief network. *10*

Ans: **Inference in Belief Networks:**

The basic task for any probabilistic inference system is to compute the posterior Probability distribution for a set of query variables, given exact values for some evidence variables.
In general, an agent gets values for evidence variables form its percepts and asks about the possible values of other ver5ables so that it can decided what action to take.

The nature of probabilistic inferences :

Belief networks are not limited to diagnostic reasoning and in fact can make four distinct kinds of interference :

1. Diagnostic inferences : From effects to causes
   Given that John Calls , infer that P(Burglary John Calls) = 0.016

2.Causal inference : From causes to effects
   Given Burglary, P(Burglary John Calls) = 0.86 and P(Mary Calls Burglary) =0.67

3. Inter causal inferences : Between causes of a common effect

   Given Alarm ,P(Burglary Alarm) =0.376

But if the evidence that earthquake is true is added then P(Burglary Alarm Earthquake) goes down to 0.003.

4. Mixed inferences: Combining two or more of the above inferences :
   Setting the effect johnCalls to truth and the cause Earthquake to false gives
   P(Alarm johnCalls ¬ Earthquake) = 0.03

that is , it is the simultaneous use of diagnostic and causal inference.

P(Burglary Alarm ¬ Earthquake) = 0.017

This is the combination of Inter causal and diagnostic inference. Belief network can also be used for the following:

Making decisions based on probabilities in the network and on the agents utilities.

Deciding which additional evidences variables should be observed in order to gain useful information.

Performing sensitivity analysis to understand which aspects of the model have the Greatest impact on the query variables.
Explaining the results of probabilistic inference to the user.

Q.No. 3. b. Explain WUMPUS World environment giving its PEAS Description. Explain How percept sequence is generated. *10*

Ans:

Problem Definition:-A Wumpus World problem comprises of a 4*4 grid with the heap of gold in one of the squares. An agent enters the grid and the aim is to find the heap of gold.

The Hurdles:-The grid has live wumpus in one of the squares. The agent is eaten up by the wumpus if he enters the wumpus square also there are pits in some square and agent dies falling into pits if the agent enters any of these squares. Also there exist walls in between some of the cells.

Performance Measures:-

-1 for each action taken

-10 for using the arrow

-1000 for falling in the pit or being eaten by the wumpus

+1000 for picking up gold

Actuators:-The agent can turn 90 degrees left or 90 degrees to right. An action grab can be taken for picking up the object in the square. An action shoot can be taken for using the arrow.

Sensors:-

Stench: In squares adjacent to the wumpus.

Breeze: In squares adjacent to pits.

Glitter: In squares where there is gold agent will perceive glitter.

Bump: If the agent walks into the wall.

Scream: If the wumpus is killed.

Now consider the environment as shown

4,1	4,2	4,3	4,4
Wumpu 3,1	Gold 3,2	Pit 3,3	3,4
2,1	2,2	2,3	2,4
Start 1,1	1,2	Pit 1,3	1,4
1	2	3	4

For the grid shown above, we explorer it till the agent reaches the square containing gold & give the final solution & steps.

- **Step 1**:

The agent is at the start position 1, 1.

Sensors {none, none , none ,none ,none}

Action – move to (1, 2)

Score –-1

4,1	4,2	4,3	4,4
Wumpus 3,1	Gold 3,2	Pit 3,3	3,4
2,1	2,2	2,3	2,4
Start 1,1 Agent	1,2	Pit 1,3	1,4

1 2 3 4

- Step 2:

Now after step 1 the agent is at the position 1, 2

Sensors { none, breeze ,none, none ,none}

Action – move to (1, 1)

Score –1—1= - 2

4,1	4,2	4,3	4,4
Wumpus 3,1	Gold 3,2	Pit 3,3	3,4
2,1	2,2	2,3	2,4
Start 1,1	1,2 Agent	Pit 1,3	1,4

- Step 3:

Now after step 2 the agent is at the position 1, 1

Sensors {none, none, none, none, none}

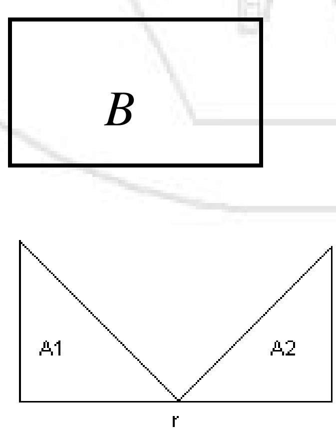
Action – move to (2, 1)

Score –2—1= - 3

4,1	4,2	4,3	4,4
Wumpus 3,1	Gold 3,2	Pit 3,3	3,4
2,1	2,2	2,3	2,4
Start 1,1 Agent	1,2	Pit 1,3	1,4
1	2	3	4

Similarly, find out the position of wumpus & glitter by visiting other adjacent squares.

Q.No. 4. *a. Explain configuration space method used in metric path planning. Hence draw configuration space for non-convex object A and obstacle B shown below.* 10



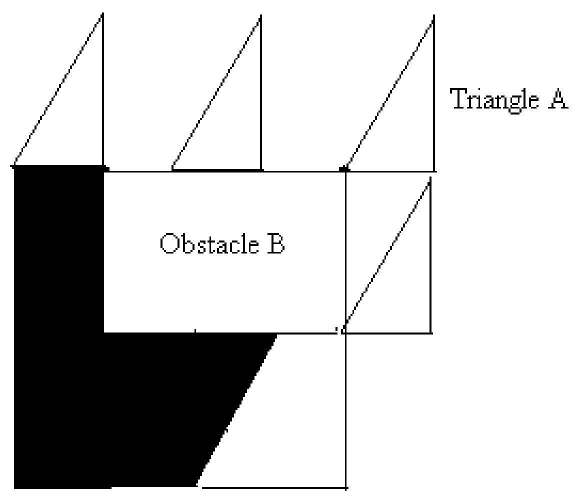
Ans:

Metric path planning is the opposite of topological navigation. Metric path planners have two components: the representation and the algorithm. Metric path planners first partition the world into a structure amenable for path planning.

The physical space robots and obstacles exist in can be thought of as the world space. The configuration space or Cspace is a data structure which allows the robot to specify the position of any objects and the robot. A good Cspace representation reduces the number of dimensions that a planner has to contend with. Consider that it takes six dimensions to represent precisely where an object is. A person may specify the location of an object as a (x, y,z) coordinates in some frame of reference. But an object is 3-D; it has a front and a back, top and bottom. Three more degrees are needed to represent where the front of chair is facing, whether it is tilted or not, or even upside down. Those are the Euler angles also called as yaw, pitch and roll.

Meadow maps: Meadow maps transform free space into convex polygon. Convex polygons have an important property: if the robot starts on the perimeter and goes in a straight line to any other point on the perimeter, it will not go out of the polygon. The polygon represents a safe region for the robot to traverse. Meadow maps are not that common in robotics, but serves to illustrate the principles of effecting a configuration space and then planning a path over it.

The configuration space is defined as “The set of all possible configurations of the part that is obtained around the obstacle”. When the Cspace of two obstacles overlaps; then there is no path for the part to move and when mere is an enlarged Cspace of obstacles then a path can be planned.



Fig, configuration space

Q.No. 4. *b. Draw and Explain general model of learning agent. Hence explain various methods of learning.* *10*

Ans: **A General Model of Learning Agents:**

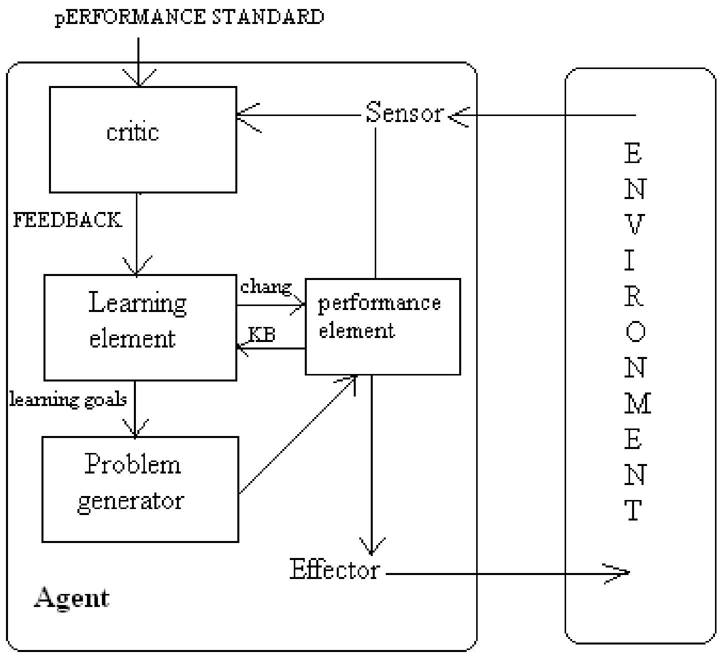


Fig. A General Model of Learning Agents

A learning agent is divided into four conceptual components i.e. learning element performance element, problem generator, critic as shown in below . The learning element is responsible for making improvement. The performance element is responsible for selecting external action. It takes in percepts and decides on actions.

The learning element takes some knowledge about the learning element and some feedback on how the agent is doing , and to tell determine how the performance element should be modified to do better in future. The design of learning element depends on the design of the performance element. The critic is designed to tell the learning element how well the agent is doing. The critic employ's fixed standard of performance.

Since the percepts do not provide indication of the agents success therefore the critic is necessary. For example, a chess program may receive a percept indicating that it has check mated its opponent, but it needs a performance standard to know that this is a good thing the percept itself does not say so.

It is important that the performance standard is a fixed measure that is outside the `agent; otherwise the agent could adjust its performance standards to meet its behavior, The last component of the learning agent is the problem generator. Problem generator is responsible for suggestion action that will lead to new and informative experiences. The point is that if the performance element had its way, it would keep doing the action that are best, given what it knows. But if the agent is willing to explore a little, and do some perhaps suboptimal action in the short run it might discover much better action for the long run.

Consider an example of the automated taxi.

a) The performance element consists of whatever collection of knowledge and procedures the taxi has for selecting its driving action i.e. Turning, accelerating, braking, honking etc. The taxi runs and drives, using this performance element.

b) The learning element formulate goals i.e. To learn better rules describing the effects of braking and accelerating, to learn the geography of the area, to learn how the taxi bravest on different conditions of the road the road is wet etc.

c) The critic observes the world and passes information along to the learning element.

For example, after the taxi makes a quick left turn across three lanes of traffic, the critic observes shocking language used by other drivers, the learning element is able to formulate a rule saying this was a bad action, and th performance elements is modified by installing the new rule.

The problem generator gives suggestion for eg. It tells which route is faster and better. The learning element is also responsible for improving the efficiency of the performance element. For example, when asked to make a trip to a new location , the taxi driver may take some time to consult its map and plan the best route. Next time a similar trip will be faster. This is called seed up learning.

Various methods of Learning are as follows:

1. Supervised Learning.
2. Unsupervised Learning.
3. Reinforcement Learning.
4. Inductive Learning.

Q.No. 5. *a. Describe Hill climbing Algorithm. What are its limitations?* *10*

Ans: **Hill Climbing Algorithm:**

The Hill Climbing Algorithm always moves towards the goal. Using Heuristic it finds which direction will take it closest to the goal. The name "Hill Climbing" comes from the analogy: A Hiker is lost half way up/ down a mountain at night. His camp is at the top of the mountain. Even though it is dark, the hiker knows that every step he takes up the mountain is the step towards his goal. So a Hill Climbing search always goes to the node closest to the goal.

A Hill climbing search algorithm is simply a loop that continuously moves in the direction of increasing value.

Algorithm:

Begin /* initially OPEN contains the root node and CLOSE is empty*/

OPEN= [start]

CLOSE= []

While OPEN ≠ [] **do**

Begin

Remove the leftmost state from OPEN and call it X

If X=GOAL **then**

Return SUCCESS

Else

Begin

1. Generate children of X
2. Put X on close
3. Discard the children of X if already on OPEN or CLOSE
4. Sort the remaining children according to the heuristic value of each state
5. Put the children on left side of OPEN in sorted order.

End

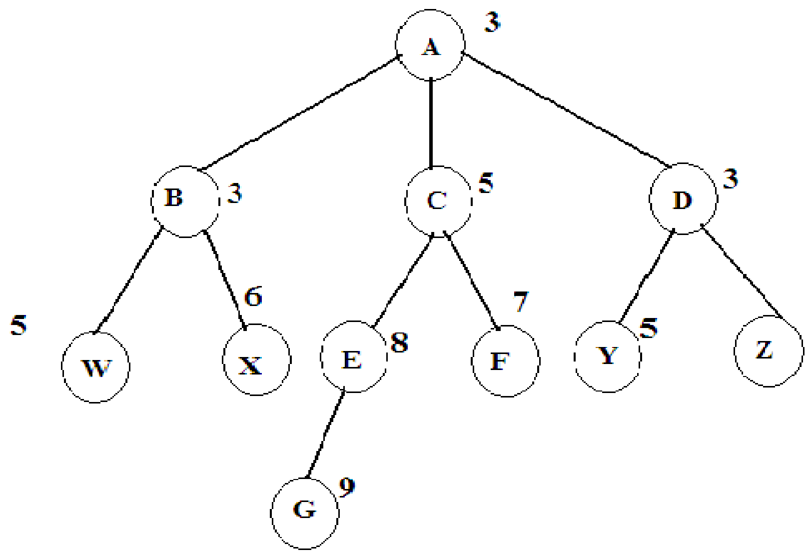
End

Return Fail

End

The algorithm described above is called as Steepest Ascent Hill Climbing, and we will always apply the same algorithm for all our problem solving. In simple hill climbing algorithm, the first closer node is chosen, whereas in steepest ascent hill climbing all successors are compared and the closest to the solution is chosen. Both forms fail if there is no closer node, which may happen if there are local maxima in the search space which are not solutions.

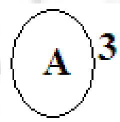
Example: Apply Hill Climbing to the following tree considering G is the goal state and node A as the initial state.



Let us apply Hill climbing algorithm to the tree. Remember, OPEN list contains all the elements which are generated but not expanded, but the CLOSE list contains all the elements which are generated as well as expanded.

Graphically a Hill Climbing Algorithm can be simulated as follows:

Step 1) The Hill Climbing algorithm starts with the initial state.

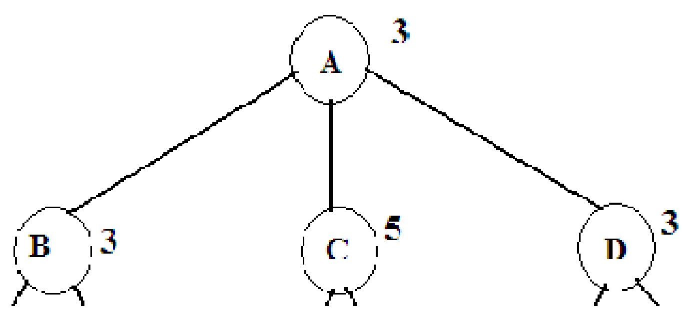


Since there is only one node so far that is node A, whose children are not yet generated it is OPEN and the CLOSE list is EMPTY at this stage.

OPEN= [A3]

CLOSE= []

Step 2) All the children of node A are now generated.

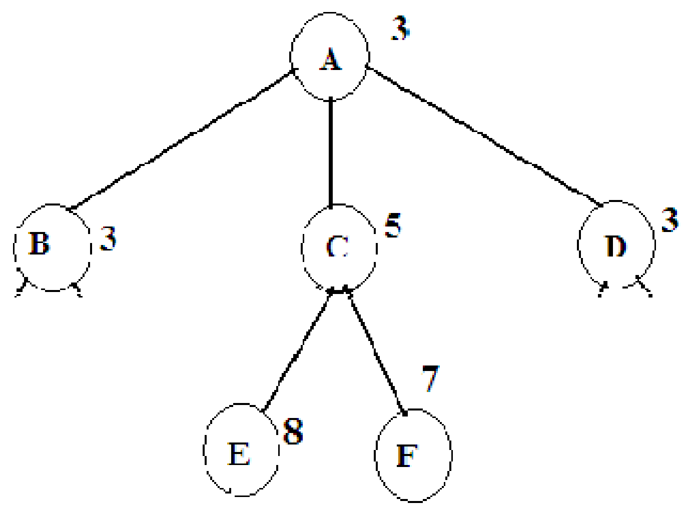


Note that the OPEN list will contain nodes B, C, D sorted from higher heuristic to lower heuristic And hence node C will be first than B and D as shown in figure. CLOSE list will contain only node A is the node whose children are generated.

OPEN= [c5, B3, D3]

CLOSE= [A]

Step 3) Amongst all the children node root node A node C has the highest heuristic merit and hence all the children nodes of C are now generated.



Now since children of Node C are generated, Node C is put into CLOSE.

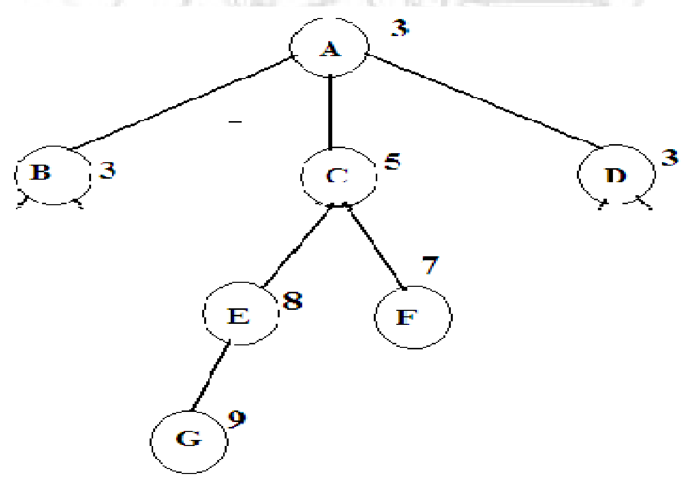
Hence, OPEN= [E8, F3, B3, D3]

CLOSE= [A, C]

Note that the leftmost node in OPEN list is node E and hence we expand node E in the next step.

Step 4)

Now the algorithm proceeds towards node E and among all the children node of node E, node G has a highest heuristic merit and hence node G is generated.



Hence,

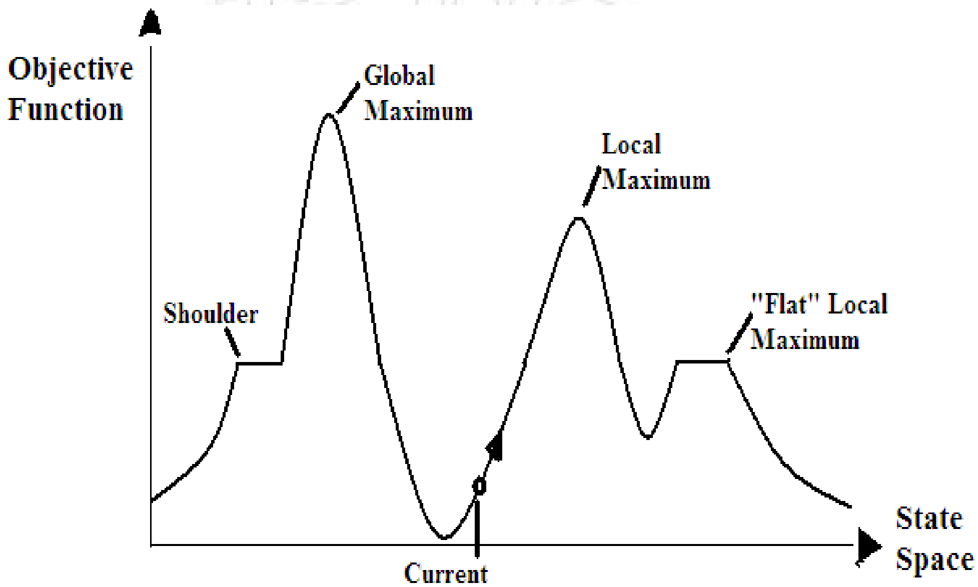
OPEN= [G9, F3, B3, D3]

CLOSE= [A, C, E]

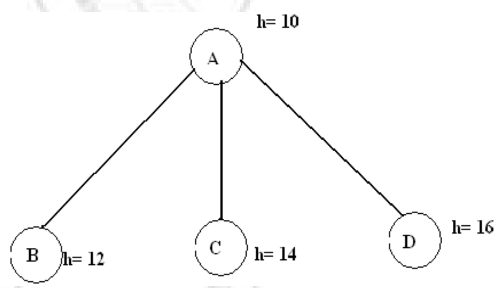
Now that the leftmost node in OPEN list is node G and node G is a goal and hence we **STOP**.

Demerits of Hill Climbing:

Waveform:

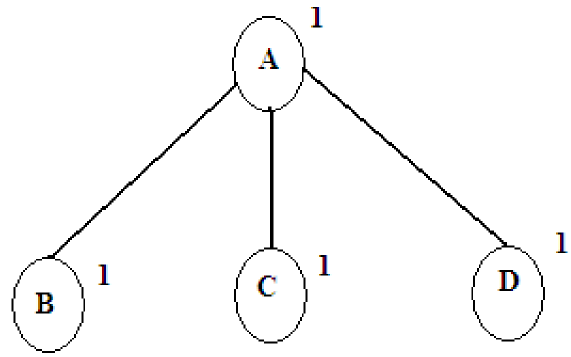


1. Local Maxima: A local maximum is a state that is better than all its neighbors but is not better than some others states farther away, at a local maximum, all moves appear to make things worse. Local maxima are particularly frustrating because they often occur almost within sight of a solution. In this case, they are called "foothills".



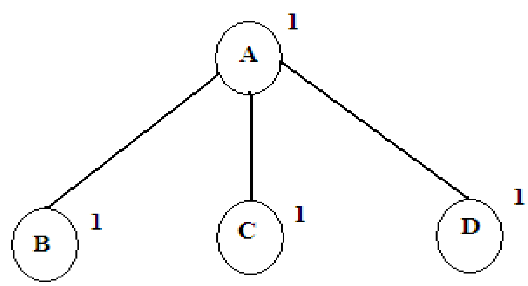
As shown all the children of state A are at less heuristic value than the heuristic value of state A. Hence we call state A as Local Maxima. Note that in this case we assume that heuristic value, and our initial state is node A with a heuristic value of 10. Hence we need to move from heuristic value of 10 to heuristic value of 0, therefore at each state we need numerical value less than the current heuristic value and not greater than the current heuristic value, any state with greater heuristic value than current state is hence worst than the current state and any state with lesser heuristic value than the current state heuristic value is hence superior than the current state.

Plateaux: A "plateau" is a flat area of the search space in which a whole set of neighbouring states have the same value. On a plateau, it is not possible to determine the best direction in which to move by making local comparisons.



As shown all the children of state A are at same heuristic value as that of the heuristic value of state A. Hence we call state A as Plateaux.

Ridges: A Ridge is a special kind of Local maximum. It is an area of the search space that is higher than surrounding areas and that itself has a slope. But the orientation of the high region, compared to the set of available moves and the directions in which they move, makes it impossible to traverse a ridge by single moves.



As shown all the children of state A are at same heuristic value as that of the heuristic value of state A. Hence we call state A as Ridge.

Q.No. 5. *b. Compute the joint variable vector* $q=[q1,q2,q3,q4]^T$ *for the following tool configuration vector of SCARA.*

$W(q)=[203.4,662.7,557,0,0,-1.649]^T$

$a=[425,375,0,0]^T$ *mm*

$d=[877,0,q3,200]^T$ *mm* *10*

Ans: **1. Compute Elbow Angle q_2:**

$$q2 = \underline{+}\arccos \frac{((w1^2+w2^2) - a1^2 - a2^2)}{(2a1a2)}$$

$q_2 = + \arccos[(203.4)^2 +(662.7)^2 -(425)^2-(325)^2/2(425)(325)]$

$q_2 = +60^o$

2. Compute base angle (q1) =

$$q1 = \underline{+}\arctan \frac{(a2s2)w1 + (a1+a2c2)w2}{(a1+a2c2)w1 - a2c2w2}$$

= arctan2[(375sin60)203.4+(425+375cos60)662.7/(425+375cos60)203.4-(375sin60)662.7]
.

$q_1 = -79.13^o$

3. **Extraction of Vertical Extension angle (q3) =**

$$q3=d1-d4-w3$$

q3 = 877-200-557 = 120 mm

4. Extraction of Tool Roll angle (q4) =

$$q4 = \pi \ln |w6|$$

$= \pi \ln |-1.649|$
$= 90^o$

Q.No. 6. *a. Consider the following statements. Rimi is Hungry. If Rimi is Hungry she barks.*

If Rimi is barking then Raja is angry.

Explain the sentences in Prepositional Logic. 03

Convert them into CNF form. 03

Prove that 'Raja is angry' Using Resolution. 04

Ans: We can use the following prepositional symbols in above set of facts.

P : Rimi is Hungry
Q : Rimi barks
R : Raja is angry

i) Now represent sentences in Prepositional Logic:

1. P
2. $P \rightarrow Q$
3. $Q \rightarrow R$

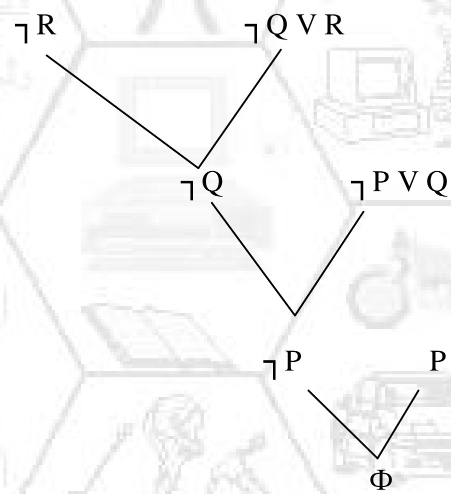
ii) Now Convert above sentences into CNF form:

1. P
2. $\neg P \vee Q$
3. $\neg Q \vee R$

Now we want to prove: Raja is angry. i.e R

Hence, negate the conclusion. i.e. $\neg R$

{**P**, $\neg P \vee Q$, $\neg Q \vee R$, $\neg R$ }



Φ indicates that $\neg R$ is false. i.e negation of conclusion is false. Hence assertion is true.

Q.No. 6. *b. Derive the general link co-ordinate transformation matrix, T^{k}_{k-1}.* *10*

Ans: There are four steps involved in constructing the homogenous transformation matrix which maps frame k coordinates into frame k-1 coordinates.

Operation	Description
1	Rotate Lk-1 about Z^{k-1} by $\boldsymbol{\theta_k}$
2	Translate Lk-1 along Z^{k-1} by $\mathbf{d_k}$
3	Translate Lk-1 along X^{k-1} by $\mathbf{a_k}$
4	Rotate Lk-1 about X^{k-1} by $\boldsymbol{\alpha_k}$

T^{k}_{k-1} is given by,

$T^{k}_{k-1}(\boldsymbol{\theta_k}, \mathbf{d_k}\ \mathbf{a_k}, \boldsymbol{\alpha_k}) =$ screw $(\boldsymbol{\theta_k}, \mathbf{d_k}, \mathbf{3})$ screw $(\mathbf{a_k}, \boldsymbol{\alpha_k}, \mathbf{1})$

$$\begin{bmatrix} C\theta k & -S\theta k & 0 & 0 \\ S\theta k & C\theta k & 0 & 0 \\ 0 & 0 & 1 & dk \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & ak \\ 0 & C\alpha k & -S\alpha k & 0 \\ 0 & S\alpha k & C\alpha k & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T^{k}_{k-1} = \begin{bmatrix} C\theta k & -C\alpha kS\theta k & S\alpha kS\theta k & aKC \\ S\theta k & C\alpha kC\theta k & -S\alpha kC\theta k & akS\theta k \\ 0 & S\alpha k & C\alpha k & dk \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Q.No. 7. *Solve any two.* *20*

a. Compare and contrast BFS & DFS. And explain the search strategy developed to overcome the drawbacks of both.

Ans:

BFS	DFS
Searches the nodes level by level.	Searches the nodes depth wise.
It requires more memory space.	It doesn't require more memory space.
Backtracking is not possible.	Backtracking is possible.
Time complexity of BFS is $O(b^d +1)$	Time complexity of DFS is O(bm)
Can't be implemented recursively	Can be implemented recursively
It is complete and optimal	Neither complete nor optimal
Solution is always guaranteed	Solution is not always guaranteed

Uniform Cost Search:

It modifies the Breadth first strategy by always expanding the lowest cost node on the fringe, rather than the lowest depth node. Breadth first search is uniform cost search with g(n)=DEPTH(n).

consider the example shown below. The problem is to get from S to G, and the cost of each operator is marked. It first expands the initial state, yielding path to A,B & C. because the path to A is cheapest, it is expanded next, generating the path SAG, which in fact a solution, though not the optimal one. The algorithm does not recognize this as a solution, because it has a cost 11, and thus is buried in the queue below the path SB, which has the cost 5. the next step is to expand SB, generating SBG, which is now the cheapest path remaining in queue, so it is goal checked and returned as a solution. Uniform cost search finds the cheapest solution provided a simple requirement is met: the cost of a path must never decrease as it goes along a path.

i.e. g(SUCESSOR(n))>=g(n) for every node n.

Depth-limited Search:

Depth limited search avoids the pit falls of depth first search by imposing a cutoff on the maximum depth of a path. This cutoff can be implemented with a special depth-limited search algorithm, or by using the general search algorithm with operators that keep track of the depth.

For example: if a person in city A travels a path of less than 20 steps, then a new state in city B is generated with a path length that is 1 greater. With this new operator set, we are guaranteed to find the solution if it exists. But we are still not guaranteed to find the shortest solution first: Depth limited search is complete but not optimal. If a depth limit is two small then depth limited search is not even complete.

The time and space complexity of Depth-limited search is similar to Depth-first search. It takes $O(b^l)$ time and O(bl) space, where l is the depth limit.

Q.No. 7. *b. Give steps in designing the Reactive Behavioral system.*

Ans:

Steps in designing the Reactive Behavioral system.

- Describe the Task.-----Specification & Analysis: ecological niche
- Describe the Robot.-----Specification & Analysis: ecological niche
- Describe the environment.-----Specification & Analysis: ecological niche
- Describe how the Robot should act in response to its environment -----Design
- Implement and refine each behavior----- Implementation & unit testing
- Test each behavior independently-----Implementation & unit testing
- Test behaviors together.-----System Testing

The methodology assumes that a designer is given a task for the robot to do, and a robot platform.

The goal is to design a robot as a situated agent. Therefore the first three steps serve to remind the designer to specify the ecological niche of the robot.

The fourth step begins the iterative process of identifying & refining the set of behaviors o the task. Defining the ecological niche defines constraints and opportunities but doesn't introduce major insights into the situatedness of the Robot: how it acts and reacts to the range of variability in its ecological niche.

Steps 5-7 are less abstract. Once the candidate set of behaviors has been proposed, the designer works on designing each individual behavior, specifying its motor perceptual schemas. This is where the designer has to write the algorithm for finding red blobs in a camera image for the random search until find red and move to red behaviors.

The designer usually programs each schema independently, then integrates them into a behavior and test the behavior thoroughly in isolation before integrating all behaviors.

This style of testing is consistent with good software engineering principals, and emphasizes the practical advantages of the reactive Paradigm.

Q.No. 7. *c. Give and explain A* Algorithm.*

Ans: **A * search :**

Greedy search minimizes the estimated cost to the goal, **h(n)** and hereby cuts the Search cost. But it is neither optimal nor complete. Uniform-cost search minimizes the cost of the path, **g(n)**; it is optimal and complete, but can be very inefficient.

Combining the above two strategies is advantageous.

i.e. **F(n) = g (n) + h (n)**

g(n) gives the path cost from the start node to node **n,** and **h(n)**is the estimated cost of the cheapest path from n to the goal.

F(n) = estimated cost of the cheapest solution through **n**. To find the cheapest solution, first try the node with the lowest value of **f.** It is complete and optimal, given a simple restriction on the **h** function.

The restriction is to choose an **h** that never overestimates the cost to reach the goal. Such **h** is called and admissible heuristic. Admissible heuristics are by nature optimistic, because the cost of solving the problem is less than it actualizes. If **n** is admissible, **f(n)** never overestimate the actual cost of the best solution through **n.** Best-first search using **f** as the evaluation function and an admissible h function is known as **A***search.

- **The behavior of A* search :**

The **A*** algorithm is a specialization of best first search. It provides general guidelines with which to estimate goal distances for general search graphs. At each node along a path to the goal, the **A*** algorithm generates all successor nodes and computes an estimate of the distance (cost) from the start node to goal node through each of the successors. It then chooses the successor with the shortest estimated distance for expansion. The successors for this node are then generated, their distances estimated, and the process continues until a good is found or the search ends in failure.

The form of the heuristic estimation function for.

A*is f*(**n) = g* (n) + h* (n)**

g* (n) = estimate of the cost (or distance) from the start node to node n .

h* (n) = the cost from node n to goal node.

are used to designate estimates of the corresponding true values f(n) = g(n) + h (n). For state space three problems g*(n) = g(n) since there is only one path and the distance g*(n) will be know to be the true minimum from the start to the current node n. This is not true in general for graphs, since alternate paths from the start node to n may exist. For this problem, it is convenient to maintain two list of node types designated as open and closed. Nodes on he open list are nodes that have been generated but not yet expanded while nodes on the closed list are nodes that have been expanded and whose children are, therefore, available to the search program.

The A* algorithm is given below :

1. place the starting node s on OPEN.
2. If OPEN is empty, stop and return failure.
3. Remove from open the node n that has the smallest value of f*(n). If the node is a goal node, return success and stop otherwise.
4. Expand n, generating all of its successor n' and place n on closed. For every successor n', if n' is not already on open or closed attach a back- pointer was changed, remove it and place it one open.
5. Each n' that is already on open or closed should be attached to back-pointers which reflect the lowest g*(n) path. If n' was on closed and its pointer was changed, remove it and place it one open. Return to step 2.

In blind or uniformed search, no preference is given to the order of successor node generation and selection. The path selected is blindly or mechanically followed. No information is used to determine the preference of one child over another. In informed or directed search, some information about the problem space is used to computer a preference among the children for exploration and expansion.

◆ **Admissibility of A* Algorithm :**

Algorithm 'A' is admissible if it is guaranteed to return an optimal solution when one

exists. The admissibility condition for an algorithm has led to corresponding definition for a heuristic function h*,h* is said to admissible if h*<= h for all n. It can be shown that if A_1 and A_2 are admissible, and is more informed than A_2, then A_1 never expands a node not expanded by A_2. In general then, it is desirable to find admissible heuristic function that approximate h as closely as possible. This will insure that few if any nodes off the optimal path are expanded. If h* = h only nodes on the optimal path will be expanded. The cost of computing such a function should also be taken into account, however. It may not be cost effective if the computation cost is too high. It has been shown that A* algorithm is both complete and admissible. Thus A* will always find an optimal path if one exist.

The cost of an A* algorithm depends on how closely h* approximates h and the cost of the computing f*.

Q.No. 7.d. Write short note on proximity sensors & representing knowledge in an uncertain domain.

Ans: **Proximity sensors:**

Proximity sensors measure the relative distance (range) between the sensors and objects in the environment. Since the sensor is mounted on robot, it is a straightforward computation to translate a range relative to the sensors to a range relative to the robot at large. Most proximity sensors are active.

Types of Proximity sensors:

- **SONAR:** Sonar refers to any system for using sound to measure range. Sonars for different applications operate at different frequencies; e.g. a sonar for underwater vehicles would use a frequency appropriate for traveling through water, while a ground vehicle would use a frequency more suited for air.

  Ultrasonics is possibly the most common sensor on commercial robots operating indoor and on research robots. They are active sensors which emits a sound and measure the time it takes for the sound to bounce back.

  The time of flight (time from emission to bounce back) along with the speed of sound in that environment is sufficient to compute the range of the object.

- **Infra-Red (IR) Sensors:** Infra-Red (IR) Sensors are another type of active proximity sensor. They emit near infrared energy and measure whether any significant amount of the IR light is returned. If so, there is an obstacle present, giving a binary signal.

  IR sensors have a range of inches to several feet, depending on what frequency of light is used and the sensitivity of receiver. The simplest IR proximity sensors can be constructed from LEDs, which emit light into the environment and have a range of 3-5 inches.

- **Bump and feeler Sensors:** Bump sensors are usually a protruding ring around the robot consisting of two layers. Contact with an object causes the two layers to touch, creating an electrical signal. The bump sensors on a Nomad 200 base protect the robot only from low obstacles not perceivable by sonar.

  Feeler sensors are whiskers or antennae, only not as sensitive as those on animals. Contact of a whisker and an object will trip a binary switch.

  Bump and feeler Sensors are actually tactile sensors since they require the robot to be touching something in order to generate a reading.

Representing Knowledge In An Uncertain Domain:

A belief network is a graph in which the following holds:

1. A set of random variables makes up the nodes of the network.
2. A set of directed links or arrows connects pairs of nodes. The inductive meaning of an arrow form node x to node y is that has a direct influence on y.
3. Each bode has a conditional probability table that quantifies the effects that the parents have on the node. The parents of a node are all those nodes that have rrows

pointing to it.
4. The graph has no directed cycles.

Consider the situation where a new burglar alarm installed, it is reliable at detecting a burglary, but it also responds to minor earthquakes. John always calls when he hears the alarm, but also gets confuse with the telephone ringing with the alarm and calls then too. Mary likes loud music and misses the alarm together.

The topology of the network can be thought of as an abstract knowledge base that holds in a wide variety of different settings, because it represents the general structure if the casual processes in the domain rather than any details of the population of individuals. In the case of the burglary network, the topology shows that burglary and earthquakes directly affect the probability of the alarm going off, but whether or not John and Mary call depends only on the alarm-the network thus represents out . assumption that they do not perceive any burglaries directly, and they do not feel the minor earthquakes.

The network does not have nodes corresponding to May currently listening to loud music, or to the telephone ringing and confusing John. These factors are uncertainty associated with the links from alarm to johnCalls and MaryCalls. The probabilities summarize a potentially infinite set of possible circumstances in which the alarm might fail to go off (high humidity, power failure, dead batte4ry, cut wires, dead mouse stuck inside bell,...) or John or Mary might fail to call and report it (out to lunch, on vacation, temporarily deaf, passing helicopter,...). In this way, a small agent can cope with a very large world (at least approximately) . The degree of approximation can be improved if additional relevant information is introduced.