ARTIFICIAL INTELLIGENCE - MAY 2015
(SEMESTER 7)

TOTAL MARKS: 80
TOTAL TIME: 3 HOURS

(1) Question 1 is compulsory.
*(2) Attempt any **three** from the remaining questions.*
(3) Assume data if required.
(4) Figures to the right indicate full marks.

---

1 (a) Discuss Belief network. (5 marks)

1 (b) Explain Heuristic function with example. (5 marks)

1 (c) Describe robot workspace. (5 marks)

1 (d) Explain Screw Transformation. (5 marks)

---

2 (a) Explain A* search with example. (10 marks)

2 (b) What is Uncertainty? Explain Bayesian network with example. (10 marks)

---

3 (a) Explain various methods of knowledge representation with example. (10 marks)

3 (b) Explain steps in problem formulation with example. (10 marks)

---

4 (a) Obtain Inverse Kinematic solution for 4-axis SCARA robot. (10 marks)

4 (b) Discuss various position sensors used in robots. (10 marks)

5 (a) Discuss partial order planning giving suitable example. (10 marks)

5 (b) Explain supervised, unsupervised and reinforcement learning with example. (10 marks)

---

6 (a) Explain the structure of learning agent. What is role critic in learning? (10 marks)

6 (b) Describe different types of environments applicable to AI agents. (10 marks)

Write short notes on:

7 (a) Properties of environment. (5 marks)

7 (b) Limitations of Hill-climbing Algorithm. (5 marks)

7 (c) PROLOG. (5 marks)

7 (d) Crypt Arithmetic. (5 marks)

1 (a) Discuss Belief network. --- *5 Marks*

ADD NOTE BOOKMARK

This question was repeated in Dec 14.

Answer:

- A belief network is a directed acyclic graph (DAG) which encodes the casual relationships between particular variables, represented in the DAG as nodes. Nodes are connected by causal links – represented by arrows – which point from parent nodes (causes) to child nodes (effects).
- As an example, a BN diagnosing diseases may have a causal link from the node "cold" to "sneezing". This encodes the causal relationship "sneezing is caused by a cold". A node can have any number of parents; the nodes "hay-fever" and "allergies" may also point to the child node "sneezing".
- Belief networks have been found to be useful in many applications related to reasoning and decision making for some time. Besides more efficient representation and computation, an additional benefit of using belief networks over a joint probability distribution is that information is represented in a more understandable and logical manner, making construction and interpretation much simpler.
- The conditional or posterior probability is used to find the probability distribution of a variable over its state, given some evidence about the state of other variables in the system.
- The syntax used for conditional probability is $P_{[query|evidence]}$;

$P_{((A=a|B=b))}$ is "the probability of variable A being in state a given the state of B is b".

- The probability distribution of A given the evidence B=b given by $P_{((A|B=b))}$, and the array of conditional probability for all states of A given each state of B is $P_{((A|B))}$ and is known as conditional probability distribution or CPD for the variable A

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

$$P(A \cap B) = P(A|B)P(B)$$

- Conditional probability can be defined in terms of standard unconditional probabilities using the product rule, as shown in equation above.

1 (b) Explain Heuristic function with example. --- 5 Marks

This question was repeated in Dec 14.

Answer:

For an 8-puzzle problem, admissible heuristic is the number of misplaced tiles.

7	2	4
5		6
8	3	1

Start State

1	2	3
4	5	6
7	8	

Goal State

h(n)=8 => all 8 tiles are misplaced from goal

The 8-puzzle was one of the earliest heuristic search problems. The objective of the puzzle us to side the tiles horizontally or vertically into empty space until configuration matched the goal configuration.

We can defince two heuristic functions for 8-puzzle problem

We can defince two heuristic functions for 8-puzzle problem

1. h_1=the number of misplaced tiles, all of the eight tiles are out of position in the diagram above, so that the start would have h_1=8. h_1 is admissible heuristic, because it is clear that any tile that is out of place must be moved at least once.
2. h_2=the sum of distances of the tiles from their goal positions. Because tiles cannot move along diagonals, the distance we will count is the sum of horizontal and vartical distances. This is something called the city of block distance or Manhattan distance. h_2 is also admissible because all any move can do is move over tile one step closer to the goal.

1 (c) Describe robot workspace. --- *5 Marks*

ADD NOTE BOOKMARK

This question was repeated in Dec 14.

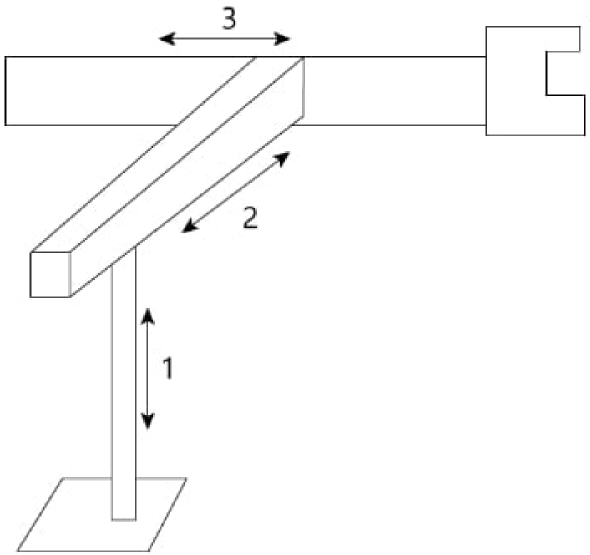
Answer:

Robot Workspace:

The workspace or configuration of a robot is defined as the locus of points in three dimensional space that can be reached by the wrist. Accordingly we get five configurations of the robot based on work envelope geometries or co-ordinate geometries which are as given as follows:

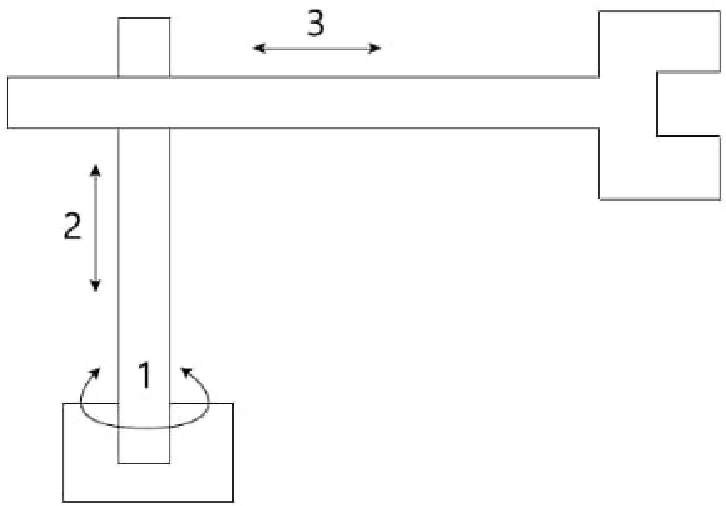
1. Cartesian robot
2. Cylindrical robot
3. Spherical robot
4. SCARA
5. Articulated robot

1. Cartesian robot:



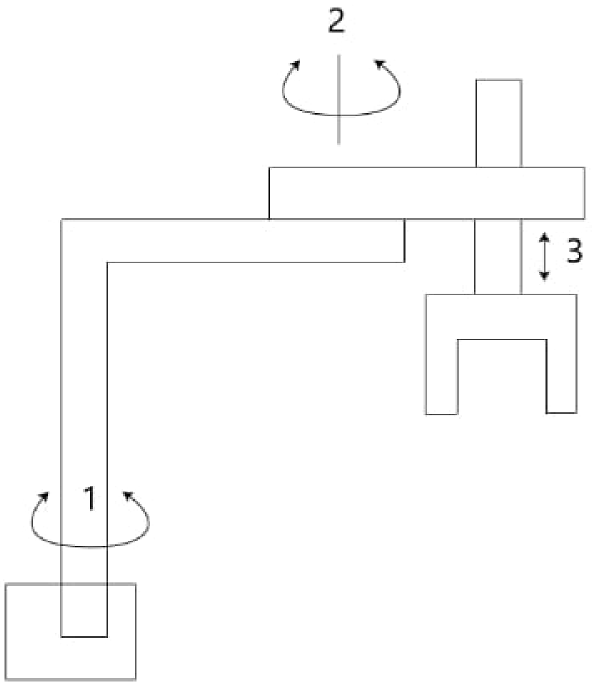
- In Cartesian robot, all three joints are prismatic joints.
- The workspace of Cartesian robot is rectangular box.

2. Cylindrical robot:



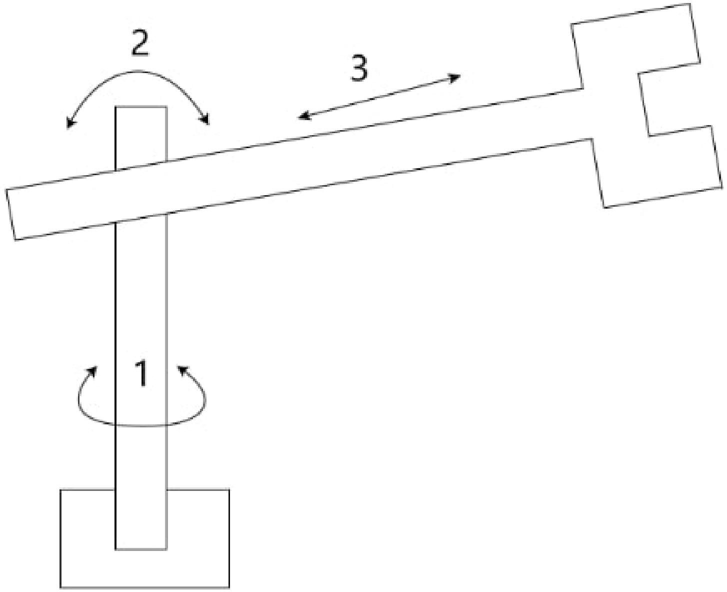
- In cylindrical robot, there is one rotary joint and two prismatic joint.
- Workspace of cylindrical robot is volume between 2 concentric cylinders.

3. Spherical robot:



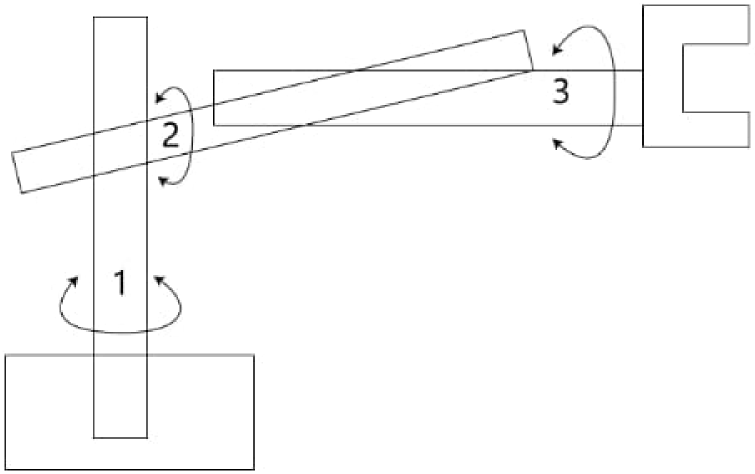
- In spherical robot, there are two rotary joints and one prismatic joint.
- Workspace of spherical robot is volume between 2 concentric cylinders.

4. SCARA robot:



- In SCARA robot, all axis are parallel, there are two rotary joints and one prismatic joint.
- Workspace of SCARA robot is complex.

5. Articulated robot:



- In articulated robot, all joints are rotary joints.
- Workspace of articulated robot is complex.

1 (d) Explain Screw Transformation. **--- 5 Marks**

ADD NOTE BOOKMARK

Answer:

Screw Transformation:

$$T = \begin{bmatrix} R & P \\ n^T & \sigma \end{bmatrix}$$

$$where R = rotation, P = translation, n^T = perspective, \sigma = scale$$

A linear displacement along an axis combined with an angular displacement about same axis is called Screw transformation. This type of motion corresponds to a threading or an unthreading operation and hence the name.

$$Screw(\lambda, \Phi, k) = Rot(\Phi, k)Tran(\lambda i^k) = Tran(\lambda i^k)Rot(\Phi, k)$$

Hence, when rotation and translations are carriied along the same axis then order does not make difference.
Properties of Screw Transformation:

1. Inverse of screw transformation is also a screw transformation

$$Screw^-1(\lambda, \Phi, k) = Screw(-\lambda, -\Phi, k)$$

2. The fundamental rotation and translation matrices associated with the unit vectors commute.

$$Rot(\Phi, k)Tran(\lambda i^k) = Tran(\lambda i^k)Rot(\Phi, k)$$

Pitch of screw $p=\emptyset/2\lambda\pi$ threads per unit length

- For pure rotation, linear displacement $\lambda = 0$, hence pure rotation is screw with infinite pitch.
- For pure translation, linear displacement $\Phi = 0$, hence pure translation is screw with zero pitch.

2 (a) Explain A* search with example. --- 10 Marks

This question was repeated in Dec 14.

Answer:

It is an uninformed search technique. It uses additional information beyond problem formulation or tree. Search is based on evaluation function f(n). Evaluation function is based on both heuristic function h(n) and g(n).

i.e. f(n)=g(n)+h(n)

$$\text{Root} \xrightarrow{g(n)} n \xrightarrow{h(n)} \text{goal}$$

For its implementation it uses two queue:

i. OPEN

ii. CLOSE

OPEN queue is priority queue which is arranged in ascending order of f(n).

Algorithm:

Step 1: Create a single member queue comprising of root node

Step 2: If first member of queue is goal, then goto step 5

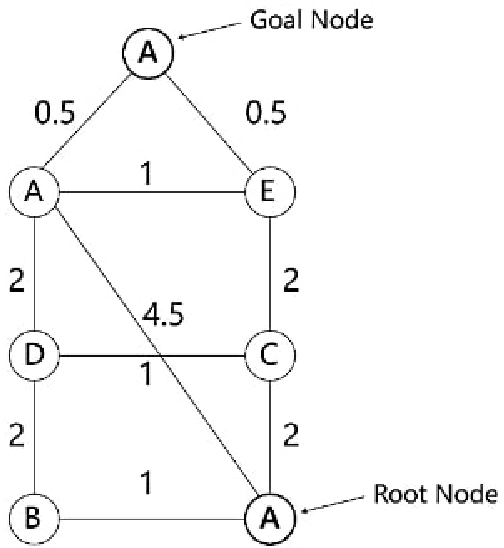
Step 3: If first member of queue is not goal, then remove t from queue and ADD to close queue. Consider it children, if any, add them into the queue, in ascending order of evaluation function f(n).

Step 4: If queue is not empty then go to step 2. If queue is empty then go to step 6

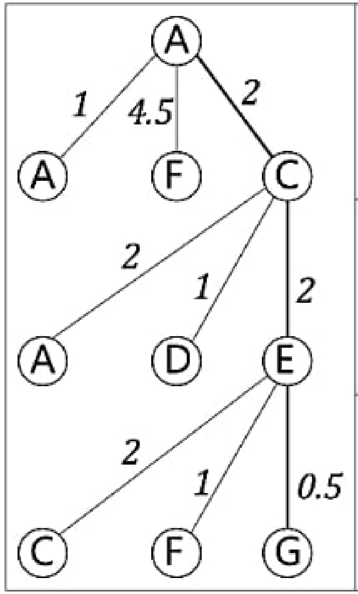
Step 5: print "SUCCESS" and stop

Step 6: print "FAILURE" and stop

Consider an example given below:



- Here we use straight line distance as heuristic h(n).
- We first convert graph into tree and then use informed search technique to solve the problem



$$Pass\ 1\ of\ algo \begin{cases} f(B) = g(B) + h(B) = 1 + 4.5 = 5.5 \\ f(F) = g(F) + h(F) = 4.5 + 0.5 = 5 \\ f(C) = g(C) + h(C) = 2 + 2.5 = 4.5 \end{cases}$$

$$Pass\ 2\ of\ algo \begin{cases} f(A) = g(A) + h(A) = 4 + 4.5 = 8.5 \\ f(D) = g(D) + h(D) = 3 + 2.5 = 5.5 \\ f(E) = g(E) + h(E) = 4 + 0.5 = 4.5 \end{cases}$$

$$Pass\ 3\ of\ algo \begin{cases} f(C) = g(C) + h(C) = 6 + 2.5 = 5.5 \\ f(F) = g(F) + h(F) = 5 + 0.5 = 5 \\ f(G) = g(G) + h(G) = 4.5 + 0 = 4.5 \end{cases}$$

OPEN	CLOSE
$[A]$	
$4.5 \quad 5 \quad 5.5$	
$[C, \quad F, \quad B]$	$[A]$
$4.5 \quad 5 \quad 5.5 \quad 5.5 \quad 8.5$	
$[E, \quad F, \quad B, \quad D, \quad A]$	$[A, \quad C]$
$4.5 \quad 5.5 \quad 5.5 \quad 5.5 \quad 8.5 \quad 8.5$	
$[G, \quad F, \quad B, \quad D, \quad A, \quad C]$	$[A, \quad C, \quad E]$

First member of queue is goal "SUCCESS"

Advantages:

- It is a complete algorithm
- It is optimum because it considers evaluation function i.e. f(n)=g(n)+h(n) where h(n) is admissible heuristic

Disadvantages:

- It generates same node again and again
- Hence large memory is needed

2 (b) What is Uncertainty? Explain Bayesian network with example. ***--- 10 Marks***

ADD NOTE

BOOKMARK

Answer:

Uncertainty:

- Unfortunately the world is an uncertain place. Any AI system that seeks to model and reasoning in such a world.
- The imperfect or incomplete knowledge causes uncertainty. In other words, we can say that an agent's incomplete or incorrect understanding of its environment causes uncertainty.
- Following are the reasons of uncertainty:
  1. **Laziness:** It is too much work to list the complete set of antecedents or consequents needed to ensure an exception less rule and too hard to use such rules.
  2. **Theoretical:** Medical science has no complete theory for the domain.
  3. **Practical Ignorance:** Even if all the rules are known, it might be uncertain about a particular patient because not all the necessary tests have been or can be run

Bayesian network:

- A bayesian or belief network is a directed acyclic graph (DAG) which encodes the casual relationships between particular variables, represented in the DAG as nodes. Nodes are connected by causal links – represented by arrows – which point from parent nodes (causes) to child nodes (effects).

- As an example, a BN diagnosing diseases may have a causal link from the node "cold" to "sneezing". This encodes the causal relationship "sneezing is caused by a cold". A node can have any number of parents; the nodes "hay-fever" and "allergies" may also point to the child node "sneezing".
- Belief networks have been found to be useful in many applications related to reasoning and decision making for some time. Besides more efficient representation and computation, an additional benefit of using belief networks over a joint probability distribution is that information is represented in a more understandable and logical manner, making construction and interpretation much simpler.
- The conditional or posterior probability is used to find the probability distribution of a variable over its state, given some evidence about the state of other variables in the system.
- The syntax used for conditional probability is P[query|evidence];

  P((A=a|B=b)) is "the probability of variable A being in state a given the state of B is b".

- The probability distribution of A given the evidence B=b given by P((A|B=b)), and the array of conditional probability for all states of A given each state of B is P((A|B)) and is known as conditional probability distribution or CPD for the variable A

$$P(A|B)=P(A\cap B)P(B)$$

$$P(A\cap B)=P(A|B)P(B)$$

- Conditional probability can be defined in terms of standard unconditional probabilities using the product rule, as shown in equation above.

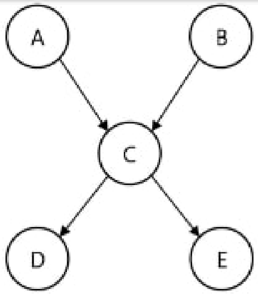


Figure: Simple example of Bayesian Network

- As shown in figure, child nodes can also be present nodes. Nodes with one or more parents require their conditional probability distribution to be provided by way of conditional probability table (CPT), which specifies the conditional probability of child node being in a particular state, given the states of all its parent: $P_{[\text{child} \mid \text{parent1, parent 2, ... partent N}]}$. A conditioning state is used to specify the states of all the parents when specifying an entry in the child node's conditional probability table. Nodes without parents only require a prior probability distribution $P_{[\text{node}]}$.
- The node C in figure would have an associated CPT specifying the conditional distribution $P_{[C|A,B]}$. Similarly, the CPT's for nodes D and E would specify $P_{[D|C]}$ and $P_{[E|C]}$ respectively. The nodes A and B have no parents, so only require the prior probability distributions P(A) and P(B).
- For Example, assuming all variables in figure are binary, where variable A can take on the states a1,a2 and so forth, the CPT for node C would be as shown in table.
- Note that in a real CPT, the probability expressions are replaced by probability values between 0 and 1.

	C		
A	**B**	**C1**	**C2**
a1	b1	$P_{(c1\|A=a1,B=b1)}$	$P_{(c2\|A=a1,B=b1)}$
a1	b2	$P_{(c1\|A=a1,B=b2)}$	$P_{(c2\|A=a1,B=b2)}$
a2	b1	$P_{(c1\|A=a2,B=b1)}$	$P_{(c2\|A=a2,B=b1)}$
a2	b2	$P_{(c1\|A=a2,B=b2)}$	$P_{(c2\|A=a2,B=b2)}$

3 (a) Explain various methods of knowledge representation with example. *--- 10 Marks*

ADD NOTE BOOKMARK

This question was repeated in Dec 14.

Answer:

1. Predicate logic:

- Formal logic is a language with its own syntax, which defines how to make sentences, and corresponding semantics, which describe the meaning of the sentences.
- Sentences can be constructed using proposition symbols (P, Q, R) and Boolean connectives, such as conjunction (and), disjunction (or), implication (P implies Q).
- Modus Ponens: where given a rule, A implies B, if A is true, we can infer that B is also true.
- Predicate logic introduce concept of quantifiers which allow us to refer to sets of objects. Using objects, attributes, and relations, we can represent almost any type of knowledge.
- Two quantifiers:
    - Universal (all objects of this type have this attribute)
    - Existential (there exists some object that has the specified attribute)
- Example: "Minnesota is cold in the winter" can be represented in three single paramters:

  Place (Minnesota), Temperature (cold) and session (winter)

  Or it can be represented a single relation:

  Cold(Minnesota,winter),winter(Minnesota,cold)

2. Resolution:

- Resolution is an algorithm for proving facts true or false by virtue of contradiction. If we want to prove a theorem X is true, we have to show that the negation of X is not true.
- Example:

Suppose that we know the following two facts

A) not feathers (Tweety) or bird (Tweety)

B) feathers (Tweety)

C) no bird(Tweety)

D) bird (Tweety)

3. Unification:

- Unification is a technique for taking two sentences in predicate logic and finding a substitution that makes them look the same.
  - A variable can be replaced by a constant
  - A variable can be replaced by another variable
  - A variable can be replaced with a predicate, as long as the predicate does not contain that variable
- Given the following set of predicates:
  - Hates(X,Y)
  - Hates(George, broccoli)
  - Hates(Alex, spinach)

3 (b) Explain steps in problem formulation with example. ***--- 10 Marks***

ADD NOTE BOOKMARK

Answer:

A search problem is computational problem that requires indentifying a solution from some, possibly infinite, set of possible solutions.

A problem can be defined formally by five components:

1. Initial State: It is an agent starts in. For example, the initial state for Agent Smart in India might be described as In(Mumbai).

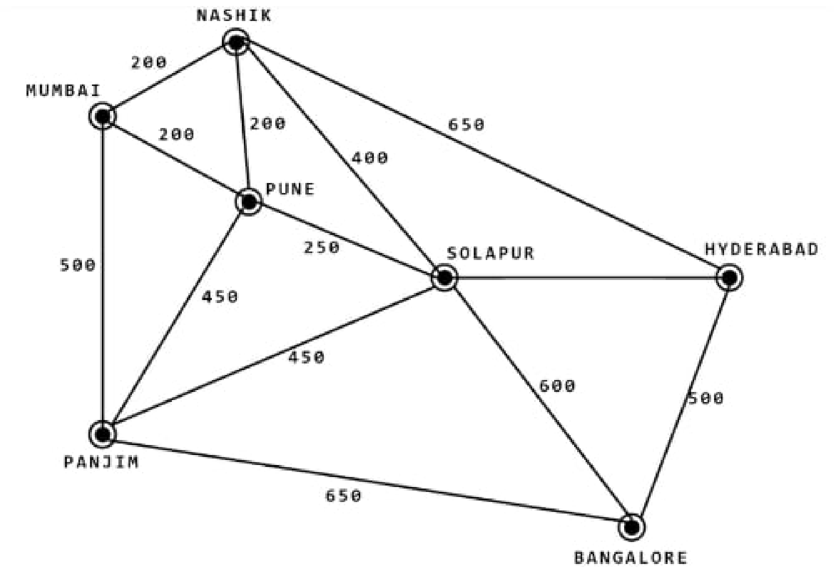
2. Actions: A description of the possible actions available to the Agent. Given a particular state 'S', ACTIONS(S) returns the set of actions that can be executed in S. For example, from the state In(Mumbai), the possible actions are: { Go(Pune), Go(Nasik), Go(Panjim)}.

3. Transition Model: This is a description of what each action does, specified by a function Result(S,A) that returns the state that results from doing action A in state S/

4. Goal Test: It determinces whether a given state is a goal state.

5. Path Cost: It assigns cost to each path. The problem-solving agent choose a cost function that reflects its own performance measure. For Agent Smart trying to get to Banglore, time is of the essence and the path cost might be measured in kilometres. The path cost can be described as:

Path Cost=Sum(Costs of all paths taken to reach goal)



Example:

Formulating the 8-puzzle problem

7	2	4
5		6
8	3	1

Start State

1	2	3
4	5	6
7	8	

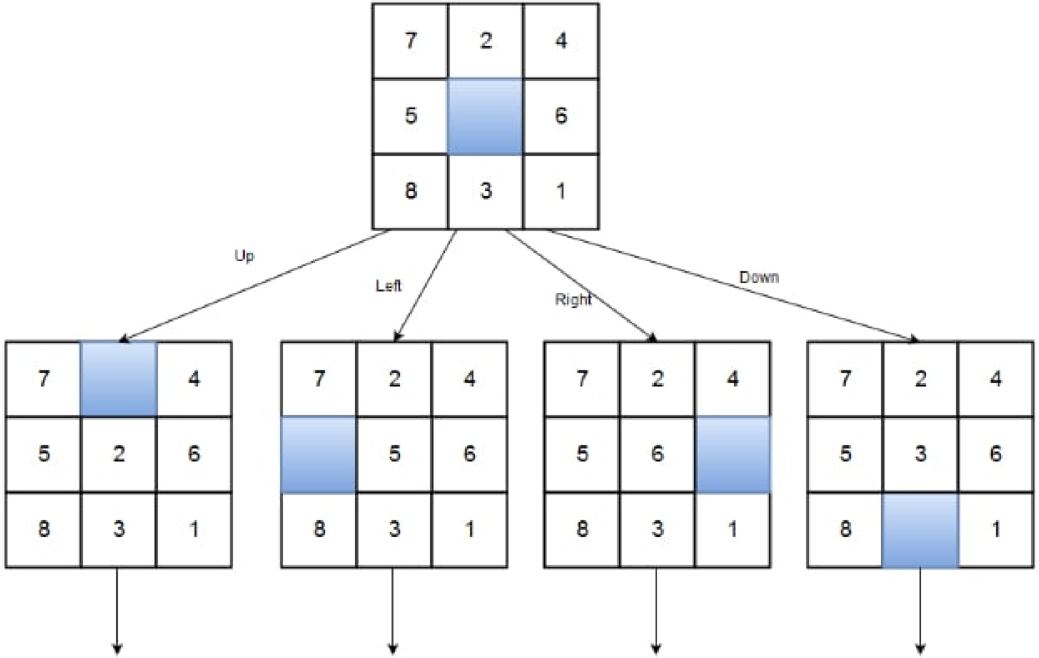
Goal State

1. **Initial State:** Specifies the location of each of the eight tiles and the blank in one of the nine squares.

2. **Operators:** Blank tiles moves left, right up or down.

3. **Goal Test:** The current state matches a certain state

4. **Path Cost:** Each move of the blank costs 1



4 (a) Obtain Inverse Kinematic solution for 4-axis SCARA robot. *--- 10 Marks*

ADD NOTE BOOKMARK

This question was repeated in Dec 14.

Answer:

A robotic manipulator can be modeled as a chain of rigid links interconnected by either revolute or prismatic joints. The relative position and orientation of two successive links can be specified by two joint parameters.

- θ_k - joint angle. It is the rotation about z^{k-1} needed to make axis x^{k-1} parallel with axiz x^k
- d_k - joint distance. It is the translation along z^{k-1} needed to make axix x^{k-1} intersects with axis x^k.
- The relative potition and orientation of the axis of two successive joints are specified by two link parameters.
- α_k - Link twist angle. It is the rotation about x^k needed to make axis z^{k-1} parallel axis z^k.
- a_k - Link lenght. It is the translation along x^k needed to mek axis z^{k-1} intersect with axis z^k

Kinematic Parameters:

Arm Parameter	Symbol	Revolute Joint (R)	Prismatic Joint (P)
Joint angle	θ	Variable	Fixed
Joint distance	d	Fixed	Variable
Link length	a	Fixed	Fixed
Link twist angle	α	Fixed	Fixed

Given:

The final position of the robot $P=(P_x,P_y,P_z)$

To find

ϕ_1, ϕ_2, ϕ_3 and ϕ_4 for the SCARA robot

The final T matrix is given below:

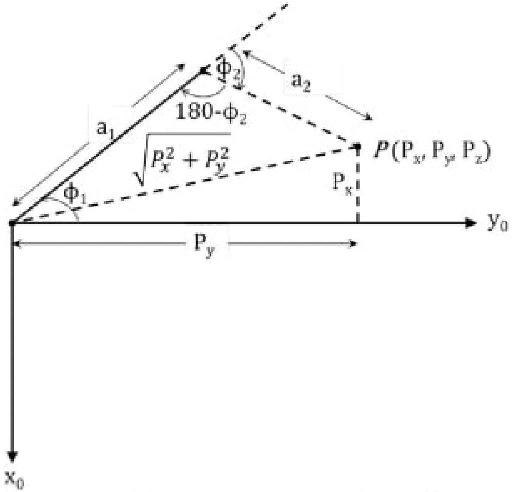
$$T_4^0 = \begin{bmatrix} C_{1-2-4} & S_{1-2-4} & 0 & a_1C_1 + a_2C_{1-2} \\ S_{1-2-4} & -C_{1-2-4} & 0 & a_1S_1 + a_2S_{1-2} \\ 0 & 0 & -1 & d_1 - q_3 - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- To find $Ø_2$: if we squre and sum P_x and P_y, we can get an expression in $Ø_2$

$$P_x^2+P_y^2=(a_1C_1+a_2C_{1-2})^2+(a_1S_1+a_2S_{1-2})^2$$

$$P_x^2+P_y^2=a_1^2+a_2^2+2a_1a_2C_1(C_1C_2+S_1S_2)+2a_1a_2S_1(S_1C_2-S_2C_1)$$

$$P_x^2+P_y^2=a_1^2+a_2^2+2a_1a_2C_1^2C_2+2a_1a_2S_1^2C_2$$



Solution to ϕ_2 of Adept, as seen from above (along Z-axis)

$P_x^2+P_y^2=a_1^2+a_2^2+2a_1a_2C_2$

$$C_2 = \frac{P_x^2 + P_y^2 - a_1^2 + a_2^2}{2a_1a_2}$$

$$S_2 = \sqrt{1 - C_2^2}$$

2. This is really just the derivation of the Law of Cosines which we can also use to find $\emptyset_2$ (see figure above):

$a_1^2 + a_2^2 - 2a_1a_2cos(180 - \phi_2) = p_x^2 + p_y^2$ (Law of Cosines)

$$cos(180 - \phi_2) = \frac{p_x^2 + p_y^2 - a_1^2 - a_2^2}{-2a_1a_2}$$

$$-cos(\phi_2) = \frac{p_x^2 + p_y^2 - a_1^2 - a_2^2}{-2a_1a_2}$$

$$cos(\phi_2) = \frac{p_x^2 + p_y^2 - a_1^2 - a_2^2}{-2a_1a_2}$$

- To solve equations in two unknowns (C_1,S_1)

$$a_1C_1 + a_2C_1C_2 + a_2S_1S_2 = P_x$$
$$a_1S_1 + a_2S_1C_2 + a_2S_2C_1 = P_y$$
$$(a_1 + a_2C_2)C_1 + (a_2S_2)S_1 = P_x$$
$$(-a_2 + S_2)C_1 + (a_1 + a_2C_2)S_1 = P_y$$
$$S_1 = \frac{a_2S_2P_x + (a_1 + a_2C_2)P_y}{(a_2S_2)^2 + (a_1 + a_2C_2)^2}$$
$$\theta_2 = atan_2(a_2S_2P_x + (a_1 + a_2C_2)P_y, (a_1 + a_2C_2)P_x - a_2S_2P_x)$$

- To solve for $\emptyset_3$:

$$P_z = d_1 - \phi_3 - d_4;$$
$$\phi_3 = d_1 - d_4 - P_z$$

- 5. To solve for $\emptyset_4$: The final roll angle cannot be determined from the position vector [px,py,pz]

If we are given the orientation matrix, then we can use the ratio of N_x , N_y to find $\emptyset_4$

$$\tan_{1-2-4} = \frac{S_{1-2-4}}{C_{1-2-4}} = \frac{N_y}{N_z}$$
$$\phi_1 - \phi_2 - \phi_4 = a\tan_2(Ny, Nz)$$
$$\phi_4 = -a\tan_2(N_y, N_z) + \phi_1 - \phi_2$$

4 (b) Discuss various position sensors used in robots. --- *10 Marks*

ADD NOTE BOOKMARK

Answer:

The various position sensors used in robots are as follows:

1. Encoders:

- Encoders are used for converting the angular or linear displacement into digital signals. Some types of encoders are:

    1. Linear encoders: It is used to calculate the directions and positions that are in the linear form
    2. Rotary encoders: It helps to calculate the directions and positions that are angular form
    3. Incremental encoders: It is used in robots for sensing the position from its last position
    4. Absolute encoders: It brings a definite position tha·t is proportional to a fixed reference position

2. Potentiometers:

- Potentiometers produce an output voltage that is proportional to the position of wiper.
- They are the analog devices used for calculating the linear or rotary movements based on the design.
- It consists of a rotating wiper, which is in contact with a resistive element.
- This wiper is connected with an object that is motion. An AC or DC voltage is supplied to the resistive element.
- The wiper and ground voltage is proportional to the ratio of the wiper's one side resistance to the resistive element's total resistance.

3. Resolvers:

- A resolver is also an analog device as like potentiometers and it is a rotary electrical transformer basically implemented for calculating the degress of rotation.
- It requires only AC signal for excitation because the use of DC current will not produce any output signal.
- The out put signal of a resolver is proportional to the angle of rotating element with respect to the fixed element.

5 (a) Discuss partial order planning giving suitable example. **_--- 10 Marks_**

ADD NOTE BOOKMARK

This question was repeated in Dec 15.

Answer:

- Planning is "the task of coming up with a sequence of actions that will achieve a goal."
- There are several types of algorithms that allow us to construct a plan such as progression planning, regression planning and partial-order planning.
- Partial order planning is not totally ordered.
- Partial order planning enables us to take advantage of problem decomposition.
- The algorithm works on several subgoals independently, solves them with severals subplans, then combines the subplans.
- In addition, such an approach also has the advantage of flexibility.
- That is, the planner can work on 'obious' or 'important' decisions first, rather than being forced to work on steps in chronological order.
- Partially order plans are created by a search through the space of plans rather than through the state space.
- ***Example:*** the simple flat tire probelm

Init(At(Spare,Truck) ^ At(Flat,Axle))

Goal (At(Spare,Axle))

Action (Remove(Spare,Trunk),

PRECOND: At(Spare,Truck)

EFFECT: At(Spare,Ground) ^ ¬ At(Spare,Trunk))

Action (Remove(Flat,Axle),

PRECOND: At(Flat,Axle)

EFFECT: At(Flat,Ground) ^ ¬ At(Flat,Axle))

Action (PutOn(Spare,Axle),

PRECOND: At(Spare,Ground) ^ ¬ At(Flat,Axle)

EFFECT: At(Spare,Axle) ^ ¬ At(Spare,Ground))

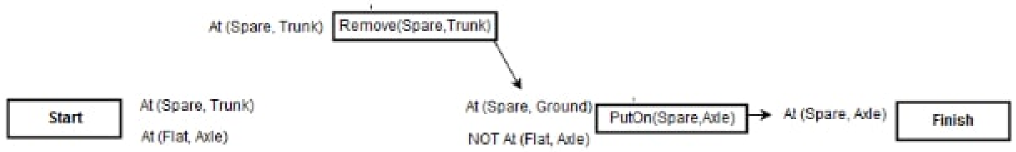
Action (LeaveOvernight,

PRECOND:

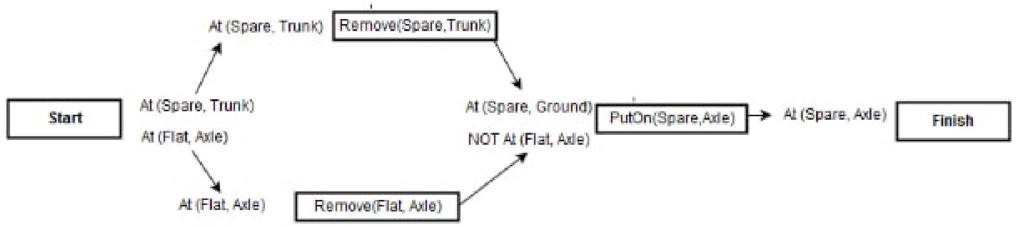
EFFECT: ¬ At(Spare,Ground) ^ ¬ At(Spare,Trunk) ^ ¬ At(Spare,Axle) ^ ¬ At(Flat,Axle) ^ ¬ At(Flat,Ground))

- We start with the empty plan consisting of just initial state and goal with no action in between, as in figure (a)
- The search procedure then looks for a flaw in the plan, and makes an addition to the plan to correct the flaw (or if no correction can be made the search backtracks and tries something else).
- For example one flaw in the empty plan is that no action achieves At(Spare,Axle). One way to correct the flaw is to insert into the plan the action PutOn(Spare,Axle).

(a)



(b)



(c)

(a) the tire problem expressed as an empty plan

(b) an incomplete partially ordered plan for the tire problem

(c) a complete partially-ordered solution

5 (b) Explain supervised, unsupervised and reinforcement learning with example.

--- 10 Marks

ADD NOTE

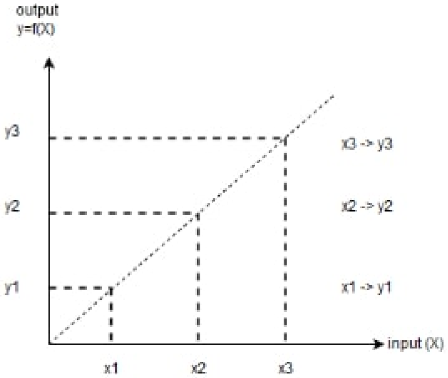
BOOKMARK

This question was repeated in Dec 14.

Answer:

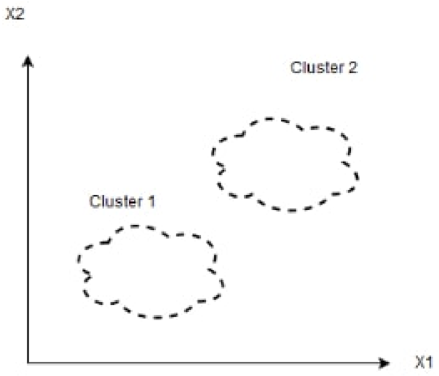
1. Supervised Learning:

- In this learning, critic provides input as well as desired output as an ordered pair
- During the training phase agent learns the relation between input (X) and output y=f(X)
- Once the relation f(X) is learned then testing phase starts.
- Example: Perceptron learning rule.



2. Unsupervised Learning:

- In this method critic provides only input. The desired output is not provided.
- Agent forms clusters from the given input training samples
- The unknown sample is classified into one of the clusters depending on its closeness
- This method is mainly useful for classification type of problems
- Example: Clustering



3. Reinforcement based learning:

- This method of learning is based on bonus/reward or penalty
- If agent action is along the performance measure then critic gives reward or bonus. Agent then learns that the action is to be repeated in future
- If critci gives penalty then agent learns that the action was not proper and should be avoided in future.
- Example: Training a dog

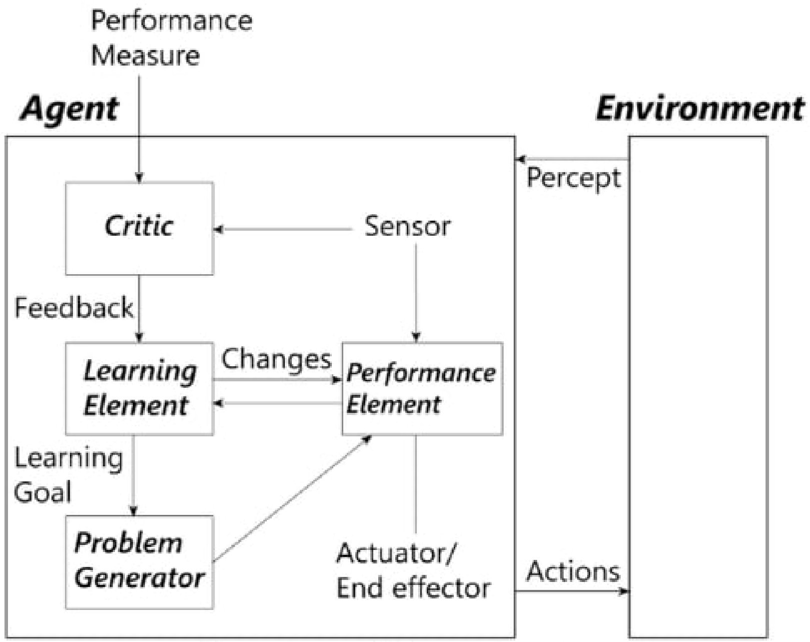
6 (a) Explain the structure of learning agent. What is role critic in learning? *--- 10 Marks*

ADD NOTE

BOOKMARK

Answer:

Structure of learning agent:



Structure of Learning Agent

An agent is anything that can be viewed as perceriving its environment through sensors and acting upon that environment through effectors.

A learning agent takes advantage of "learning" allowing the agent to initially operate in unknown enviroments and to become more competent that its initial knowledge alone might allow.

A general learning agent has four basic components:

1. **The performace element :-** Which takes in percepts and decides on appropriate actions in the same way as a non-learning agent.
2. **The critic :-** Which uses a fixed standard of performnce to tell the learning element how well the agent is doing.
3. **The learning element :-** That receives information from the critc and makes appropriate improvements to the performance element.
4. **The problem generator :-** That suggests actions that will lead to new and informative experiences.

The design of learning element depends very much on the design of the performance element. The critic tells the learning elemt how well the agent is doing with repect to a fixed performance standard. The critic is necessary because the percepts themselves provide no indication of the agent;s success.

Learning agent is also responsible for improving the efficiency of performance element for example when asked to make a trip to new destination, the taxi might take a while to consult its map and plan the best route. But next time, on a similar trip, the planning process should be much faste. This is called speed up learning request.

Role of critic in learning:

It compares performance measure and the percept from sensor about the enviroment and guides learning element. Critic plays a very important role in the structure of learning agent. Critic has access to performance measure and gets the percepts from the sensor about the environment, on comparision between both performance measure and percept, it provides a feedback or actually guides the learning element on receiving the feedback from the critic. Learning element can change the rules.

6 (b) Describe different types of environments applicable to AI agents. ***--- 10 Marks***

ADD NOTE BOOKMARK

Answer:

The different types of environment applicable to AI agents are as follows:

1. Fully observable vs. Partially observable: A task environment is effectively fully observable, if the agent's sensors are able to detect all the aspects that are relevant to its choice of action.

2. Deterministic vs. stochastic: If a next state of the environment is completely determined by an agent, and any variations are excluded, then the environment is deterministic. Otherwise, it is stochastic.

3. Episodic vs. sequential: Episodic environment is divided into atomic episodes, each of which consist of agent perceiving and performing a single action. Next episode is independent from actions taken in the previous episode. In contrast, in sequential environment, each decision can affect all the future decisions.

4. Static vs. Dynamic: If an environment is changing while an agent is deliberating, then it is dynamic. Static environments does not change over time. Semidynamic environments does not change, but an agent's performance score does.

5. Discrete vs. Continuous: Describes a state of the environment, the way time is being handled, and to the percepts and action of an agent. Chess game is discrete (finite number of states, discrete set of actions). Taxi driving is continuous.

6. Single agent vs. multiagent: Either an agent is acting in the environment solely, or engage into certain relationships with other agents, distinguishing them from other objects of the environment (by identifying that its own perormance depends on other agent's performance). Multiagent environment can be competitive, cooperative, or partially both.

7 (a) Properties of environment. *--- 5 Marks*

ADD NOTE BOOKMARK

Answer:

Properties of environment:

1. Fully observable vs. Partially observable: A task environment is effectively fully observable, if the agent's sensors are able to detect all the aspects that are relevant to its choice of action.

2. Deterministic vs. stochastic: If a next state of the environment is completely determined by an agent, and any variations are excluded, then the environment is deterministic. Otherwise, it is stochastic.

3. Episodic vs. sequential: Episodic environment is divided into atomic episodes, each of which consist of agent perceiving and performing a single action. Next episode is independent from actions taken in the previous episode. In contrast, in sequential environment, each decision can affect all the future decisions.

4. Static vs. Dynamic: If an environment is changing while an agent is deliberating, then it is dynamic. Static environments does not change over time. Semidynamic environments does not change, but an agent's performance score does.

5. Discrete vs. Continuous: Describes a state of the environment, the way time is being handled, and to the percepts and action of an agent. Chess game is discrete (finite number of states, discrete set of actions). Taxi driving is continuous.

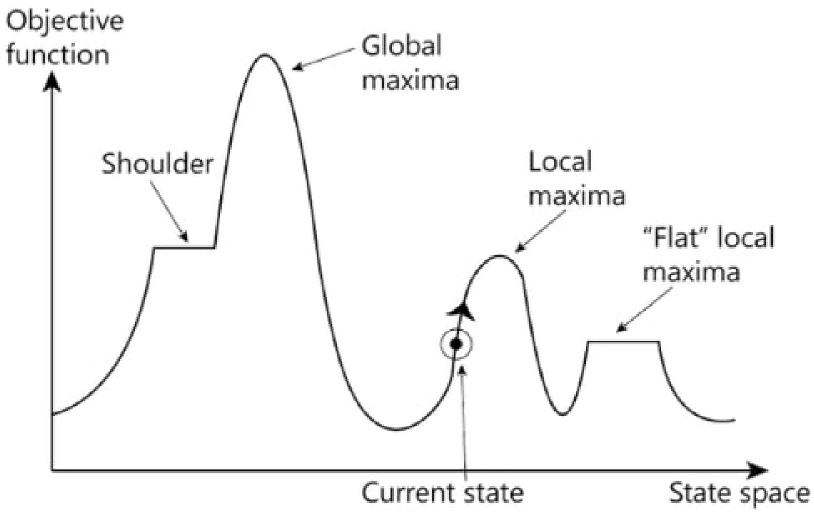
6. Single agent vs. multiagent: Either an agent is acting in the environment solely, or engage into certain relationships with other agents, distinguishing them from other objects of the environment (by identifying that its own perormance depends on other agent's performance). Multiagent environment can be competitive, cooperative, or partially both.

7 (b) *Limitations of Hill-climbing Algorithm.* **--- 5 Marks**

ADD NOTE BOOKMARK

Answer:

Consider following state-space landscape



Local Maxima:

- A problem with hill climbing is that it will find only local maxima. Unless the heuristic is convex, it may not reach a global maximum. Other local search algorithms try to overcome this problem such as stochastic hill climbing, random walks and simulated appealing. This problem of hill climbing can be solved by using random hill climbing search technique.

Ridges:

- A ridge is a curve in the search space that leads to a maximum but the orientation of the ridge compared to the available moves that are used to climb is such that each move will lead to a smaller point. In other words, each point on a ridge looks to the algorithm like a local maximum, eve, through the point is part of a curve leading to a better optimum

Plateau:

- Another problem with hill climbing is that of a plateau, which occurs when we get to a "flat" part of the search space i.e. we have a path where the heuristics are all very close together. This kind of flatness can cause the algorithm to cease progress wander aimlessly.

7 (c) PROLOG. --- 5 Marks

ADD NOTE BOOKMARK

Answer:

PROLOG:

- PROLOG (PROGramming LOGic) is one of the most widely used programming languages in artificial intelligence research.
- PROLOG is a declarative programming language.
- That means, when implementing the solution to a problem, instead of specifying how to achieve a certain goal in a certain situation, we specify what the situation (rules and facts) and the goal are and let the Prolog interpreter derive the solution for us.
- In Prolog, program logic is expressed in terms of relations, and a computation is initiated by running a query over these relations.
- Relations are defined by clauses.

Syntax of Prolog:

(a) Clauses:

- A prolog program consists of clauses
- Each clause terminates with a full stop
- Prolog clauses are of three types: rules, facts and questions
- Example 1: rule = offspring(X,Y):-parent(Y,X).
- Example 2: facts = parent(tom,ann).

- Example 3: question = parent(X,ann).

(b) Procedures:

- A group of clauses which the head of each clause are the same is called a procedure
- Example:

connects(san_francisco,oakland,bart_tan).

connects(san_francisco,framont,bart_tan).

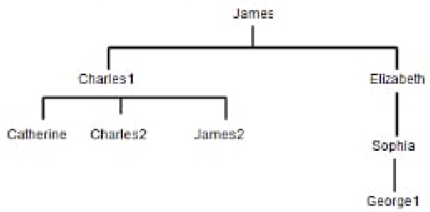
connects(concord,daly_city,bart_tan).

(c) Rules:

- A rule is a clause with one or more conditions. For a rule to be true, all of its conditions must also be true
- The right-hand side of the rule is a condition and the left-hand side of the rule is the conclusion
- The ':-' sign means 'if' and ',' are understood as conjunctions
- Example:

offspring(X,Y):-parent(Y,X).

Example of PROLOG:



male(James1).

male(James2).

male(Charles1).

male(Charles2).

male(George1).

female(Elizabeth).

female(Catherine).

female(Sophia).

parent(Charles1,James1).

parent(Elizabeth,James1).

parent(Catherine,Charles1).

parent(James2,Charles1).

parent(Charles2,Charles1).

parent(Sophia,Elizabeth).

parent(George1,Sophia).

7 (d) Crypt Arithmetic. *--- 5 Marks*

ADD NOTE BOOKMARK

Answer:

Crypt Arithmetic:

- Cryptarithmetic is a mathematical puzzle in which each letter represents a digit (for example, if X=3, then XX=33).
- The objective is to find the value of each letter. Here, no two letters represent the same digit (i.e., if X=3, Y cannot be 3).
- And the first letter cannot be 0 (i.e., given the value ZW, Z cannot be 0).
- Example illustrating how to solve a cryptarithmetic puzzle:

	S	E	N	D
+	M	O	R	E
-	-	-	-	-
M	O	N	E	Y

STEP 1:

- This is an addition problem. Now we know that the sum of two four digit numbers can't be more than 19,998. But, M can't be O according to the rules since it's the first letter. This means that the sum is more than 10,000 and M must be 1.
- Now the problem is simplified to:

	S	E	N	D
+	1	O	R	E
-	-	-	-	-
1	O	N	E	Y

STEP 2:

Now in the column, S-1-0, S+ 1>=10. So, either S must be 8 (with 1 carried over from the column E-0-N) or 9 (with or without 1 carried over from the column E-0-N.)

Case I:

S=9; and there's 1 carried over from the column E-0-N:
In this case, 0 will come out to be 1. But since, M is also 1, 0 cannot be 1.

Case II:

S=9; and there's no carried over from the column E-0-N:
In this case, 0 will come out to be 0.

Case III:

S=8; and there's 1 carried over from the column E-0-N:
In this case also, 0 will come out to be 0.
Thus, 0 must be equal to 0.

Simplifying the expression, we get:

	S	E	N	D
+	1	O	R	E
-	-	-	-	-
1	O	N	E	Y

STEP 3:

- Consider the column E-0 (zero) -N. There could be 1 carried over from this column only in one case, i.e., E=9; and there's a 1 carried over from the column N-R-E. However, in that case, N would come out to be O as:
- E+O+I (carried) = 10 - 1 carried over and N=O.
- But N cannot be 0, as O has already been found out to be 0. So E <9 and there is no carry from this column. Therefore, from Case III above, S=9.

	9	E	N	D
+	1	O	R	E
-	-	-	-	-
1	O	N	E	Y

STEP 4:

In the column E-0-N, E cannot be equal to N, so there must be a carry from the column N-R-E and E+I N.
Now consider the column N-R-E;

- We know that E+I N.
- We also know that there is a carry from tlhis column, so:

<u>*Case I:*</u>

N+R=IE (if there is no carry from the column D-E-Y)

N+R=1E and E+1= N (i.e., E=N-1)
:. N+R - 10+ (N-1)
:. R = 9
But S is also 9. So this assumption must be wrong.

<u>*Case II:*</u>

N+R+I=IE (if there is a carry from the column D-E-Y).

N+R+1=1E and E+1=N (i.e., E=N-1) :. N+R+I = 10+ (N-1)
:. R = 8

The original problem now simplifies to:

	9	E	N	D
+	1	0	8	E
-	-	-	-	-
1	O	N	E	Y

STEP 5:

The digits left are 7, 6, 5, 4, 3, and 2.
Possible values for E are 5 and 6 because:

- There must be a carry from the column D-E-Y, which means D+E>10. D is at most 7, so E cannot be 2 because then D+E<1O. Also, E cannot be 3 or 4 because then D+E=1O or D+E=11 implying Y=O or Y=1, which conflicts with O=O and M=1.
- N=E+1, so E can't be 7 because then N would be 8 which conflicts with R=8.

Case I:

E=6

N=E+l gives N=7. Also, D+E>lO, Y!=0 and Y!=1 gives D=7 But this is contradictory.

Case II:

E=5

N=E+1 gives N=6. Again, D+E>=10, Y!=O and Y!=1 gives D=7 and Y=2. This is not contradictory and must be the correct result.

Thus the final solution is:

	9	**5**	**7**	**6**
+	1	O	8	5
-	-	-	-	-
1	O	6	5	2