Final – ~~2022~~ 17-18

1. a) What are the learning outcomes of artificial Intelligence?

Ans! The learning outcomes of artificial Intelligence are:

1. Learning and understanding of fundamental AI ~~including~~ will gain the main theory of machine learning, neural network, algorithms.
2. AI can create such solution/software or device which can solve real-world problems very easily and accurately such as - health issues, marketing, traffic issues
2. With the help of AI, you can create your own personal assistant such as - google assistant, siri etc.
3. AI increases the skill of data processing, feature engineering and prepare data AI models.
4. AI can make or build robot that can work in a environment where survival of humans can be at risk.

5. AI opens a new path to new technologies, new devices and new applications.

6. AI ~~will~~ helps people to interact with computer vision concepts such as image recognition, object detection.

7. Learning AI will ~~b~~make people experties in designing, training, evaluating AI models.

8. AI explores techniques for processing and understanding human language, including applications such as sentimental analysis, language translation and chatbot development

b. Define in your own words the following terms: agent, agent program, rationality, autonomy, deterministic, stochastic.

Ans;

Agent: An agent is a system or entity that perceives its environment through sensors and acts upon that environment through actuators. Agents can be software programs, robots, or entity capable of making decisions and taking actions.

Agent Program: An agent program is the software or algorithmic component of an agent that determines its behavior. It processes the information received from sensors and generates actions for the actuators on predefined rules.

Rationality: Rationality refers to the ability of an agent to consistently achieve its objective or goal. An agent is considered to be rational agent if it selects actions that are expected to maximize its chances of success based on the available information.

Autonomy:
Autonomy refers to the degree of independance or self-governance exhibited by an agent. An autonomous agent can make decisions and take actions without direct external intervention.

Deterministic:
If the next state of an environment is completely determined by current state and action executed by the agent, then the environment is called deterministic. The behavior of deterministic agent is entirely predictable and there is no randomness.

Stochastic:
If the next state of environment is not determined by current state then it is called stochastic. It refers to the presence of randomness. An stochastic agent incorporates some level of randomness in its decision-making process.

c) What is PEAS in specifying the task environment? Illustrate and describe the structure of the model-based reflex agent

Ans) ~~To specify~~, the rationality of a agent, ~~we need to~~ are specified by Performance, Environment, Actuators and Sensors. They are grouped together and are called PEAS. They specify the task environment. Each letters in PEAS represent a different aspect of task environment.

Performance measure: This component specifies how the success of the agent in performing the task will be measured. It defines the criteria used to evaluate the performance of the agent.

Environment: The environment is the external context or surroundings in which the agent operates.

Actuator: Actuators are the mechanism or component through which the agent interacts with environment.

Sensors:

Sensors are the means by which the agent perceives or gathers information about its environment.

Here is a example of PEAS of the task environment for an automated taxi:

Agent Type	Performance measure	Environment	Actuators	sensors
Taxi driver	Safe, fast legal, comfortable trip, maximize profits	Roads, other traffic, pedestrians, customers	Steering, accelerator, breake, signal, horn, display.	camera, sonar, speedometer, GPS, odometer, accelerometer, engine sensors, keyboard.

Here is the model based reflex agent:

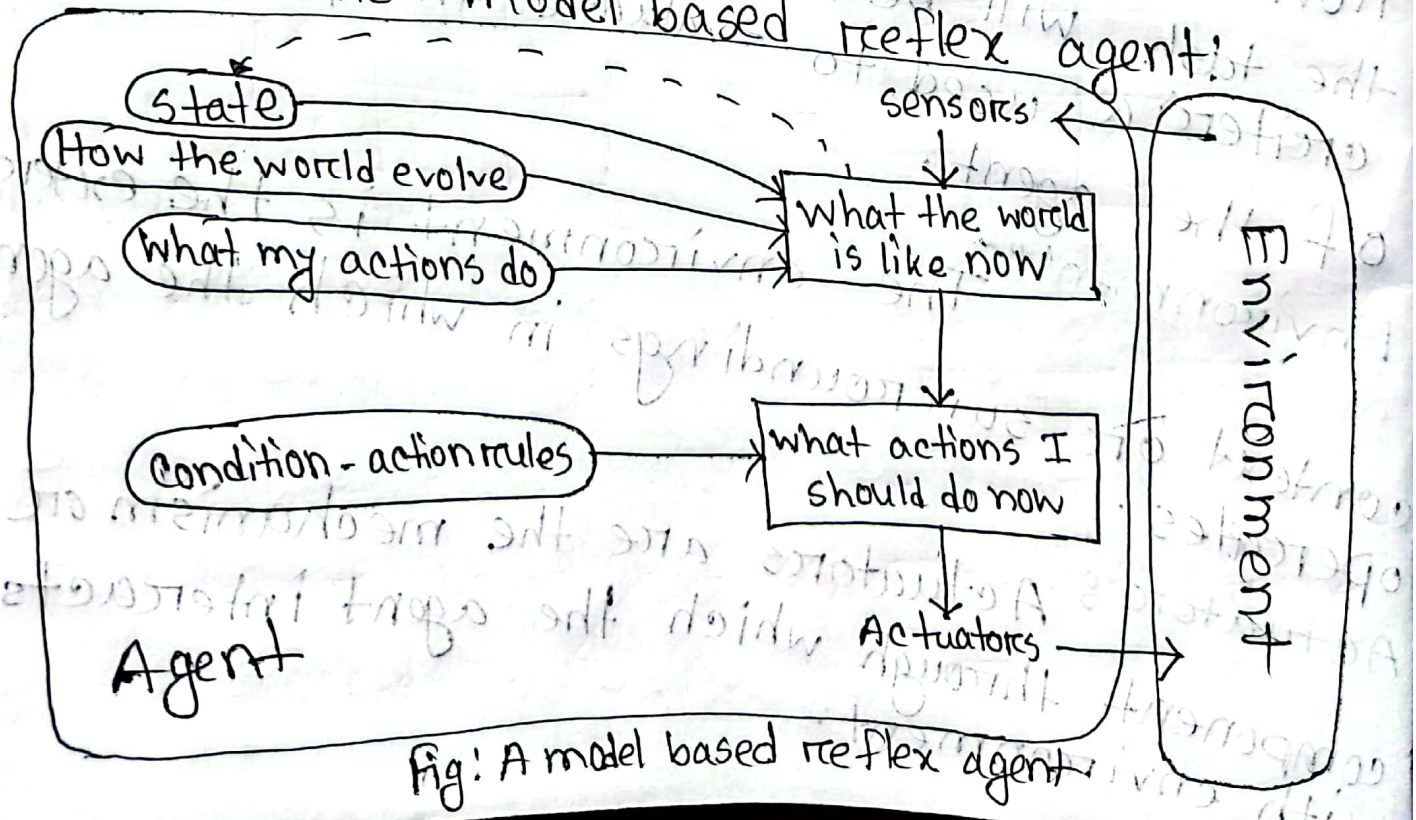


Fig: A model based reflex agent

A model based reflex agent is a type of intelligent agent that uses a model of the world to make decisions and take actions in response to stimuli from its environment. The effective way to handle partial observability is for the agent to keep track of the part of the world it can't see now. The agent should maintain some internal state that depends on the percept history and reflect the current state. Updating the internal state information requires two kinds of knowledge, one is information about how the world evolve and how the agent's own action affect the world. The actions that a agent take will affect the environment. when the environment is ~~affe~~ effected the world would look change. The box what he world is like now represents the agent's best guess.

2. a) What is uninformed search? Show that the 8 puzzle states are divided into two disjoint sets such that no state in one set can be transformed into a state in other set by any number of moves.

Ans: নোট খাতায় লেখা আছে।

The goal state has number of certain orderings which we will measure as starting at the upper left corner, then proceeding left to right and when we reach the end of a row going down to the leftmost square in the row below.

7	2	4
5		6
8	3	1

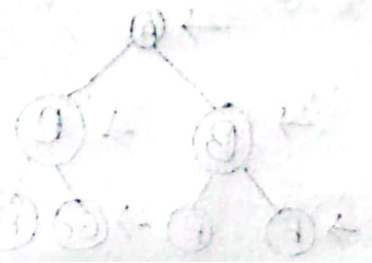
Start state

1	2	
3	4	5
6	7	8

Goal state.

Let N, denote the sum of total number of inversion First of all, sliding a tile horizontally changes neither the total number of inversions nor row number of empty square. Therefore let us considers sliding tile vertically.

Let's assume, that the tile A is located directly over the empty square. Sliding it down changes the parity of row number of empty square. Now consider total number of inversion. The move only affects relative positions of tiles A, B, C, D. If none of the B, C, D cause inversion relative to A then after sliding one get three of additional inversion! If one of the three is smaller than A, the before the move B, C, D contributed a single inversion whereas after the move they'll contributing two inversions - a change of 1 also an odd number. Tow additional cases obviously lead to the same result. Thus the change in the sum N is always even. This precisely what we have to show.



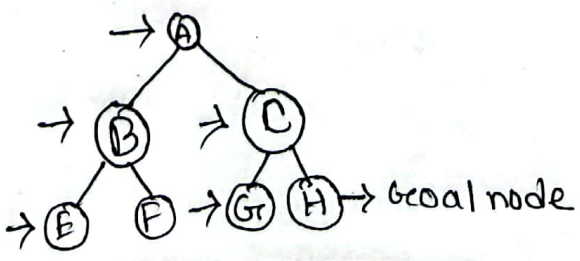
b) Prove that uniform-cost search and breadth first search with constant step costs are optimal when used with the GRAPH-SEARCH algorithm. 2018-19 (1b)

Ans:

~~Uniform cost search :~~

Breadth-first-search:

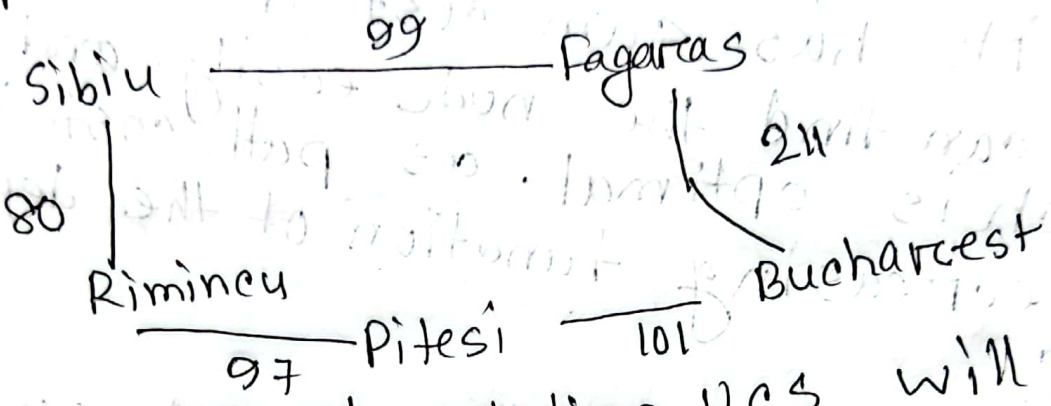
In BFS, all edges have the same cost and it explores nodes level by level. It explores all nodes at the current level before moving on to nodes at the next level. BFS explores level by level and since all step costs are constant, it ensures all nodes at a given level ~~has~~ have the same cost. BFS will find if there exists an optimal path to a node. ~~The~~ BFS explores nodes in orders of their distance from the start node and goal node ~~is found~~ is guaranteed to be found.



Here BFS ... will go through A node and then go to its next level B node. As A has two node B and C, BFS will go through also C node. Our desired node is H. After searching B and C node BFS will go E and F and then G and H. As it has sam cost in each level BFS can find the node easily. and the search is optimal. as path cost is ~~do~~ nondecreasing function of the depth of the node.

Uniform cost search: UCS explores nodes in increasing order of path cost. It use the lowest cumulative cost to find a path from the source to destination. Nodes are expanded, starting from the root, according to the minimum cumulative cost. Then it is implemented using a priority queue. It UCS in every step the path with the least cost is choosen and paths never gets shorter as nodes are added ensuring the search expand. UCS explore the optima

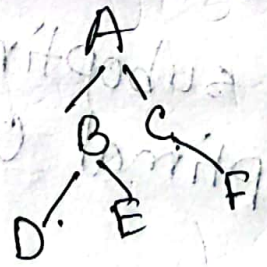
path first. So, if there is any other path with lowest cost UCS selects the lowest path to get the answere shortly. It is guranteed to find optimal solution because it explores paths in increasing order of cost suppose



To get the optimal solution UCS will go at first Sibiu to Riminicu. Then least cost expandeed next adding Pitesi 80+97 =177. which is greatere than Fagarras 99. so, UCS will go Fagaras. Then adding Bucharest 99+211 = 310. As UCS choosing and adding a path to Bucharest with cost 80+97+101 = 278 1 is less than 99+211 = 310. Now, UCS checks new path is better than 278 than 310. No goscosts is 278 and UCS will find the goal through Sibiu → Fagaras → Bucharest.

9) Show a state space with constant step costs in which GRAPH-SEARCH using iterative deepening finds a suboptimal solution. Compare the four evaluation criteria set for several uninformed search strategies. 2018-19 (1 C)

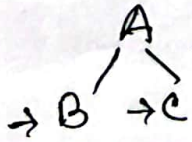
Ans:- Iterative deepening is a combination of depth first search and breadth first search. It repeatedly applies depth-first-search with increasing depth limit until solution is found. It gives suboptimal solution when constant step cost is used. Suppose,



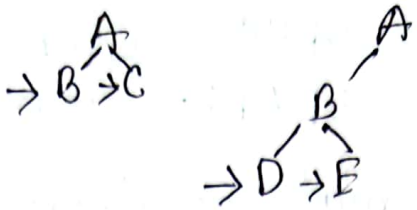
Suppose a constant step cost of 1 to all edges. The goal is to find optimal path starting from A. optimal path is A → with cost 2. Iterative deepening may find suboptimal solution:

first iteration (Depth Limit -1):

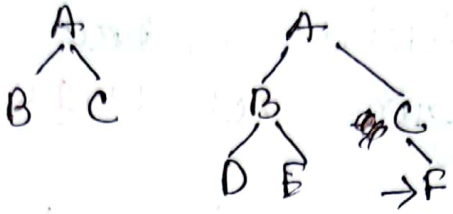
Limit-1 →A



Limit = 2 → A



Limit = 3 A



Iterative deepening encounters the goal node at depth 3 before finding A→C at depth 2. Since Iterative deepening explores nodes in a depth-first manner and increments depth limit iteratively. It finds some suboptimal solution before reaching the optimal one.

*comparing the four evalution criteria set এগুলো নোট করা।

3.a) What is the heuristic function of informed search strategies? How to minimize the total estimated solution cost using the best first search, A* search algorithm? 2018-19 (1d)

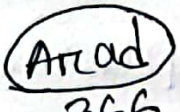
Ans: নোট থেকে।

Best first search:

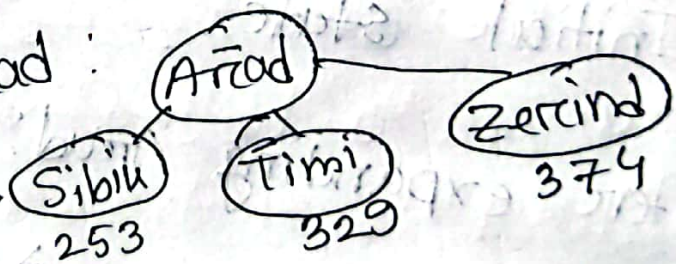
Best first search tries to expand the node that is closest to the goal. or lead to a solution quickly. It calculates the heuristic value for each node. $f(n) = h(n)$

Example:

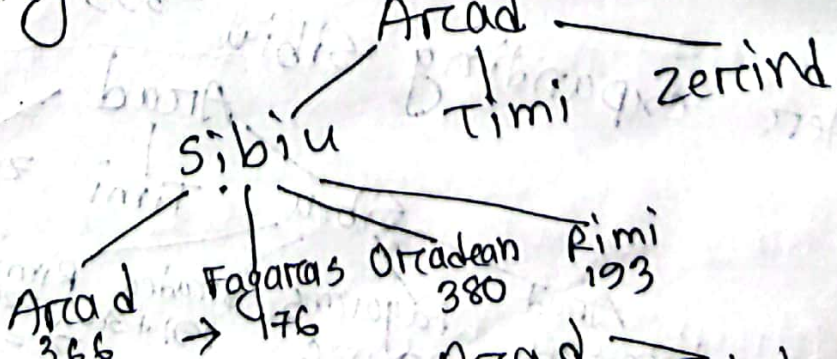
a) Initial state:



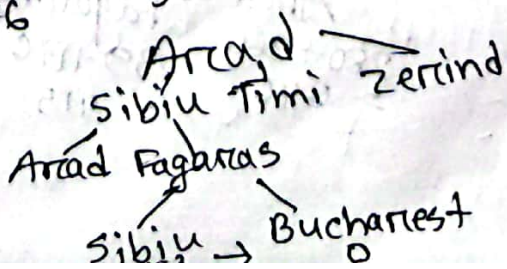
b) after expanding Arad:



c) after expanding sibiu.



d) after expanding fagaras!



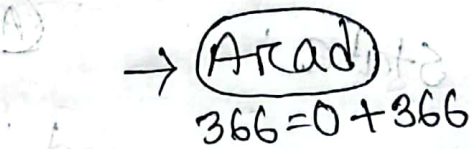
after each expandation estimated cost is counted and inlowest estimated cost field is selected.

A* search incorporates both the cost of reach a node. g(n) (path cost) and a heuristic estimated cost h(n). It evaluates nodes by combining g(n), the cost to reach the node, and h(n) the cost to get from the node to the goal.

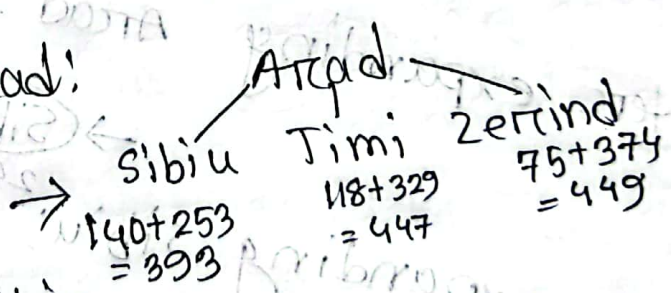
$$f(n) = g(n) + h(n)$$

Example:

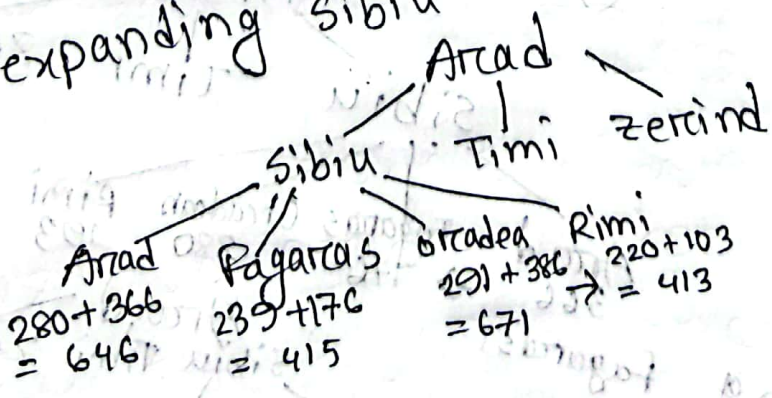
a) Initial state:



b) after expanding Arad:



c) after expanding Sibiu

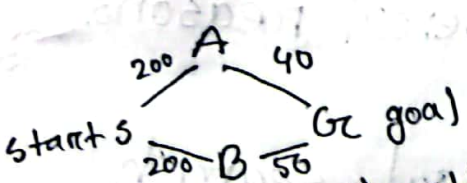


Here in each expanding step, sum of $g(n)$ and $h(n)$ is counted and lowest $f(n)$ is encountered. If the next step iteration the value is lowest than the previous step then the ~~cost is~~ goal node is converted to next step.

both Best first search and A* search minimize the total estimated solution cost. A* incorporates both actual cost and estimated cost, making it an informed and optimal search algorithm. while Best first search uses only the heuristic value to estimate solution cost.

when, $h(n) \geq$ actual cost, then it is overestimate

$h(n) \leq$ actual, " " " underestimate



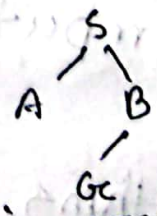
here, $g(A) = 200$
$g(B) = 200$

actual cost, $A \to Gr = 90$
" " , $B \to Gr = 50$

case 1: Overestimation

$h(A) = 80$, $h(B) = 70$ } > actual cost

$f(A) = 200 + 80 = 280$
$f(B) = 200 + 70 = 270$
$f(Gr) = g(Gr) + h(Gr) = 250 + 0 = 250$



As A is greatest so the search stopped

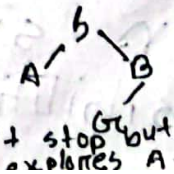
case 2: Underestimation

Let, $h(A) = 30$, $h(B) = 20$ } < actual cost

$f(A) = 200 + 30 = 230$
$f(B) = 200 + 20 = 220$ $f(Gr) = 250$

Underestimation gives optimal solution

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



A* doesn't stop Gr but A is less than Gr so it explores A then Gr

b) Why do we use local strategy to address optimization problem? Show how the last configuration of 4 queens on a 4x4 board has fewer conflicts than the first configuration using local search strategy in where conflicts means there are no two queens on the same row, column or diagonal.

2018-19 {2(a)}

Ans:

We use local search strategies ~~in~~ to address optimization problem in which the aim is to find the state according to objective function. We can use local search strategies to address optimization problem for several reasons:

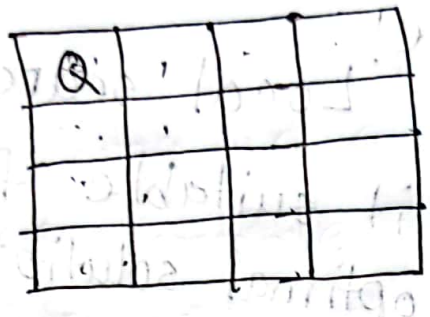
1. Computational efficiency: Local search algorithms often provide efficient solutions for optimization problems, especially when solution space is large and exploring all possibilities is impractical.

2. Complex objective function:
Local search methods can be effective in optimizing complex and non-linear spaces.
3. Heuristic nature:
Local search is heuristic in nature, making it suitable for problems where finding the globally optimal solution is challenging or even impossible.
4. Memory efficiency:
5. Search space exploration:
6. Adaptability to Dynamic environments:
7. Incremental improvement.

Let's consider N-Queen problem with N=4 on a 4x4 chessboard. The goal is to place four queen on the board in such a way that no two queen share the same row, column or diagonal.

As local search algorith operate using single current node and generally move

only to neighbours of that node. It is not systematic but takes little memory. Its aim is to find best state according to an objective function.



At first, ~~initialize~~ Q will be placed in first column. The conflicts are:

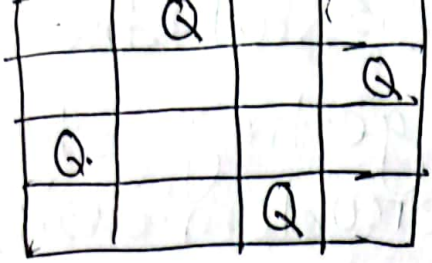
⇒ Queen 1 at position (1,1) has 0 conflict

⇒ Queen 2 at " (2,1) has 0 conflict

⇒ Queen 3 at position (3,1) has 0 conflict.

⇒ Queen 4 at position (4,1) has 0 conflict.

After local search algorithm the queens are rearranged. ~~aft~~



The conflicts can be calculated!

⇒ Queen 1 at position (1,2) has 0 conflict

⇒ Queen 2 " " (2,4) " 0 "

⇒ " 3 " " (3,1) " 0 "

⇒ " 4 " " (4,3) " 0 "

Local search has successfully found a solution with zero conflicts. They efficiently navigate the solution space to improve the current solution and reduce conflict in this case.

c) what are the problems of the hill climbing algorithm for getting stuck? How to escape this problem using simulated search algorithm?

Ans:

Hill climbing is a local search algorithm that makes iterative improvements by moving toward the direction of increasing elevation in the search space. ~~Unfort~~ The problems of hill climbing gets stuck are:

1. Local maxima: a local maxima is a peak that is higher than each of its ~~nea~~ neighbouring states but lower than global maimum. Hill climbing algorithm that reach the vicinity of a local maximum will be drawn upward toward the peak but will then be stuck with nowhere else to go.

2. Ridges: Ridges result in a sequence of Local maxima that is very difficult for greedy algorithm to navigate.

3. plateux: A plateu is a flat area of the state space landscape. It can be a flat local maximum. from which no uphill exit exists, or a shoulder from which progress is possible. A hill climbing search might get lost on the plateu.

Simulated annealing is metaheurcistic than can help overccome the limitations of hill climbing. It is a probabilistic optimization algorcithm inspired by the annealing proces It introduces controlled randomness to escape local optima and explorce the solution space morce effectively.

1. Accepting worcse moves: simulated annealing introduces a probability of accepting worcse moves durcing search. This allows to explorce regions with higherc cost which helps escape local optima.

2. Divercsification: This algorcithm incorcporates diverc sification mechanism by allowing moves

that increase the objective function value.

3. Avoiding Premature convergence:

By introducing randomness and allowing for moves that increases the objective function values, simulated annealing avoids premature convergence and higher likelihood of finding the global optimum.

chapter - 13

4. a) Define in your own words the following terms: decision theory, random variable conditional probability and marginal probabil

Ans:

Decision theory: The combination of utility theory and probability theory is called decision theory.

Decision theory = probability theory + utility theory.

Random variable: Variables in probability theory are called random variable. It represents a quantity whose value is not certain and can

vary due to chance. In dice example Total and Die, are random variable.

Conditional probability: It refers to the probability of an event occuring given that another event has already occured.

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

Marginal Probability: It refers to the probability to sum up the probabilities for each possible value of the other variables, thereby taking them out of the equation.

$$P(Y) = \sum_{z \in Z} P(Y, z)$$

↓

sum over all possible combinations of values of the set of variables z

Chapter 6

b) What is Bayer's Rule? Compute the patient's probability of having the liver disease if they are an alcoholic. "Being an alcoholic" is the test for liver disease. Past data tells you that 10% of patients entering your clinic have liver disease and 5% of the clinic's patients are alcoholic. You might also known that among those patients diagnosed with liver disease 7% are alcoholic.

Ans;

Bayer's Rule:

We know Product rule;

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

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

Equating two right hand side

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

$$\Rightarrow P(b|a) = \frac{P(a|b)\,P(b)}{P(a)}$$ [Dividing P(a) in both side]

This equation is known as Bayer's rule.

To compute the probability that a patient has liver disease and ~~have~~ are alcoholic, we can use Baye's theorem.

$$P(\text{Liver disease} \mid \text{Alcoholic}) = \frac{P(\text{Alcoholic} \mid \text{Liver disease}) \times P(\text{Liver disease})}{P(\text{Alcoholic})}$$

Here given:

10% of patients have liver disease,

$$P(\text{Liver disease}) = 10\% = \frac{10}{100} = 0.10$$

5% of the patients are alcoholic.

$$P(\text{Alcoholic}) = 5\% = \frac{5}{100} = 0.05$$

~~$P(\text{Liver disease} \mid \text{Alcoholic}$~~

$P(\text{Alcoholic} \mid \text{Liver disease}) = 7\% = \frac{7}{100} = 0.07$

$$P(\text{Liver disease} \mid \text{Alcoholic}) = \frac{0.07 \times 0.10}{0.05}$$

$$= 0.14$$

$$= 14\%$$

c) Design a naive Bayes model, Bayesian classifier based on the dentistry example.

Ans:

	toothache		¬toothache	
	catch	¬catch	catch	¬catch
cavity	0.108	0.012	0.072	0.08
¬cavity	0.016	0.064	0.144	0.576

From the full joint distribution:

$$P(cavity \mid toothache \wedge catch) = \alpha \langle 0.108, 0.016 \rangle \approx \langle 0.871, 0.129 \rangle$$

we can use Bayes rule to reformulate the problem

$$P(cavity \mid toothache \wedge catch) = \alpha P(toothache \wedge catch \mid cavity) P(cavity \quad -(1)$$

Each variable toothache and catch are directly caused by cavity.

$$P(toothache \wedge catch \mid cavity) = P(toothache \mid cavity) P(catch \mid cavity)$$

This equation expresses conditional independence of toothache and catch given cavity.
We can write eqn ① to obtain the probability of cavity.

$$P(\text{cavity} \mid \text{toothach} \wedge \text{catch}) = \alpha P(\text{toothache} \mid \text{cavity}) P(\text{catch} \mid \text{cavity}) P(\text{cavity})$$

~~In th~~ The general definition of conditional independence of two variable X and Y given a third variable Z is

$$P(X, Y \mid Z) = P(X \mid Z) P(Y \mid Z)$$

In the dentist domain, conditional independance of variable toothache, catch, cavity;

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

we can derive a decomposition as follows.

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

$$= P(\text{toothache}, \text{catch} \mid \text{cavity}) P(\text{cavity}) \quad [\text{Product rule}]$$

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

This example illustrates a commonly occurring pattern in which a single cause directly influences a number of effects, all of which are conditionally independent, given the cause. The full joint distribution can be written as:

$$P(Cause, Effect_1, Effect_n) = P(Cause) \prod P(Effect_i | Cause)$$

This probability distribution is called a naive Bayes model.

5. a) what is supervised learning? How to learn decision tree using entropy and information gain of attributes?

Ans:

Supervised learning:

Supervised learning is a learning in which the agent observes some example input-output pairs and learns a function that maps from input to output. For example an agent training to become taxi driver. Every time the instructor shouts "Brake!" the agent learn a conditional action rule for when to brake. In this example the inputs are percept and outputs are provided by teacher who says "Brake!"

Entropy is a measure of uncertainty of random variable; acquisition of information corresponds to a reduction in entropy. In general entropy of a random variable V with values v_k, each with probability $P(v_k)$ is defined as:

Entropy: $H(V) = \sum_k P(v_k) \log_2 \frac{1}{P(v_k)} = -\sum_k P(v_k) \log_2 P(v_k)$

In decision tree learning, If a training set contains p positive examples an n negative example then entropy of the goal attribute on the whole set is

$$H(\text{Goal}) = B\left(\frac{p}{p+n}\right)$$

Here B is the entropy of Boolean random variable that is true with probability q.

$$B(q) = -(q \log_2 q + (1-q) \log_2 (1-q))$$

Information gain is a measure of the effectiveness of a particular feature in classifying data. The goal is to find features that best separate the data into different classes.

suppose an attribute A with d distinct values divides the training set E into subsets. Each subsets E_k has positive examples p_k and n_k negative examples.

we need additional $B(p_k/(p_k+n_k))$ bits of information to answer the question.

expected entropy remaining after testing attribute A is:

$$\text{Remainder}(A) = \sum_{k=1}^{d} \frac{p_k + n_k}{p+n} B\left(\frac{p_k}{p_k+n_k}\right)$$

The information gain from the attribute test on A:

$$\text{Gain}(A) = B\left(\frac{p}{p+n}\right) - \text{Remainder}(A).$$

b) Distinguish generalization loss and empirical loss. show the model selection using error rates of training and validation data for different size decision trees.

Ans:

Generalization loss	Empirical Loss
1. It is a measures the performance of a model on a dataset that it has not seen during training.	1. It measures the performance of a model on the training dataset.
2. It also known as test loss or out of sample loss.	2. It is known as training loss or in-sample loss.
3. It reflects how well the model generalizes to new, unseen data.	3. It compares the model's predictions to the actual target value.

Generalization Loss	Empirical Loss
4. It indicates how well the model is expected to perform on new, unseen example	4. It indicates how well the model learning from the training data.
5. It represents how well the model fits the training data and also generalize new, previously seen data	5. It represents how well the model fits the training data

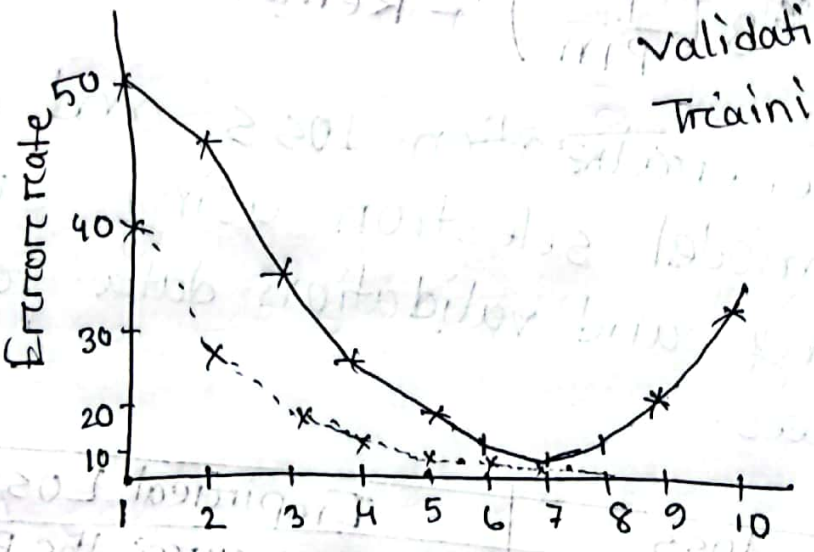


Fig: Error rates on training data and validation data for different size decision tree.

In this curve, the training set error decreases monotonically, while the validation set error decreases at first and then increases when the model begins to overfit. The cross validation procedure picks the value of size with the lowest

validation set error, the bottom of the U shaped curve. We stop when the training set error rate asymptotes, and then choose the tree with minimal error validation set, in this case the of size 7 node's.

c) What is univariate linear regression? How to minimize the loss using gradient descent for fitting linear regression?

Ans:

Univariate linear regression: A univariate linear regression function with input x and output y has the form $y = w_1 x + w_0$, where w_0 and w_1 are real-valued coefficients to be learned. W represent coefficients as weight. W can be define as vector $[w_0, w_1]$.

$$h_w(x) = w_1 x + w_0$$

The task of finding the hw that best fits these data is called linear regression.

- To minimize the loss using gradient desce we choose starting point in weight space, a point in the (w_0, w_1) plane, and then move to a neighbour point that is downhill repeating until we converge on the minimu possible loss.

$w \leftarrow$ any point in the parameter space
loop until convergence do
for each w_i in w do

$$w_i \leftarrow w_i - \alpha \frac{\partial}{\partial w_i} Loss(w)$$

α is called step size, usually called the learning rate.

The slopes in the simplified case of only one training example (x, y):

$$\frac{\partial}{\partial w_i} Loss(w) = \frac{\partial}{\partial w_i} (y - h_w(x))^2$$

$$= 2(y - h_w(x)) \times \frac{\partial}{\partial w_i} (y - h_w(x))$$

$$= 2(y - h_w(x)) \times \frac{\partial}{\partial w_i} (y - (w_1 x + w_0))$$

applying this to both w_0 and w_1.

$$\frac{\partial}{\partial w_0} Loss(w) = -2(y - h_w(x))$$

$$\frac{\partial}{\partial w_1} Loss(w) = -2(y - h_w(x)) \times x$$

6. a) What do artificial neural network mean? How do the human brain work?

Ans: An artificial neural network is a computational model inspired by the structure and functioning of the human brain. It consists of interconnected nodes organized into layers. The basic building blocks are neuron, which receives inputs, apply weights to them and procedure output through an activation function. Neural network are used in AI for tasks such as pattern recognition, classification, regression.

The human brain is an incredibly complex and intricate organ responsible for various cognitive functions. It consists of 86 billion neurons, each connected to thousands of other through synapses

. Neuron communicate through electrical and chemical signals. The brain's functionality involves processes such as learning, memory, perception, language and decision making

b) Illustrate and describe the standared activation functions?

Ans:

The activation function

$$g(in_j) = g\left(\sum_{i=0}^{n} w_{i,j} a_i\right)$$

Neural networks are composed of nodes or units connected by directed links. A link from unit i to unit j serves to propagate the activation a_i from i to j. Each link has weight $w_{i,j}$ associated with it, which determines the strength and sign of the connection. Just as in linear regression model, each unit has a dummy input $a_0 = 1$ with an associated weight $w_{0,j}$. Each unit j first computes a weighted sum of its input.

$$in_j = \sum_{i=0}^{n} w_{i,j} a_i$$

then applies an activation function g to this sum to derive the output.

$$a_j = g(in_j) = g\left(\sum_{i=0} w_{i,j} a_i\right)$$

c) How to adjust the weights of perceptron in neural network?

Ans; A perceptron is the simplest form of a neuron that takes multiple input values, applies weight to them, sums them up, add a bias, and then passes the result through an activation function. The adjustment of weights can be done by a process called back propagation. Here are the back propagation algorithm to adjust weight:

function BACK-PROP-LEARNING (example, network)
returns a neural network
inputs: examples, ~~networks~~ a set of examples
networks, a multilayer network with L layer.
local variables: Δ, a vector of errors.
repeat

for each weight $w_{i,j}$ in network do
$w_{i,j} \leftarrow$ a small random number
for each example (x, y) in example do
for each node i in the input layer do
$a_i \leftarrow x_i$
for $l = 2$ to L do
for each node j in layer l do
$in_j \leftarrow \sum_i w_{i,j} a_i$
$a_j \leftarrow g(in_j)$
for each node j in the output layer do
$\Delta[j] \leftarrow g'(in_j) \times (y_i - a_j)$
for $l = L-1$ to 1 do
for each node i in layer l do
$\Delta[i] \leftarrow g'(in_j) \sum_j w_{i,j} \Delta[j]$
for each weight $w_{i,j}$ in network do
$w_{i,j} \leftarrow w_{i,j} + \alpha \times a_i \times \Delta[j]$
until some stopping criterion is satisfied
return network

d) what is the necessity of k-fold cross validation technique?

Ans: we can get accurate estimate of data using k-fold cross validation technique. In this technique we split the data into k equal subsets. we then perform k rounds of learning on each round 1/k of the data is held out as a test set and the remaining example are used as training data. The average test set score of k rounds are better than single score.

The necessity of k-fold cross technique is:
K-fold cross techniquess provides more accurate estimate of a model's performance compared to a single split.

2. It ensures that every data point is used for both training and testing, maximizing the use of available data.

3. K-fold-cross validation helps detect overfitting by evaluating the model.

4. It is valuable when tuning hyperparameters

5- When comparing different models or algorithm, cross-validation provides more unbiased performance comparison.

6. It helps ensuring that each fold has a representative mix of examples from all classes.

7. It allows practitioners to assess the stability of the model by observing how performance metrics vary accross different folds.

2018-19

1.a) what does artificial Intelligence mean, AI?

Ans: Artificial Intelligence is the art of creating machines that perform functions that require intelligence when performed by people. It refers to the development of computer systems or software that can perform tasks that typically require human intelligence. These tasks include learning, reasoning, problem-solving perception, natural Language understanding and even creative tasks.

b) 2017-18-2(b)

c) 2017-18-2(c)

d) 2017-18-3(a) Show the heuristic must be consistent for the optimal solution in the A* search algorithm.

Ans: The consistency condition for A* search algorithm:

$$h(n) \leq c(n, a, n') + h(n')$$

$h(n) \rightarrow$ heuristic cost estimate from state n to the goal.

$c(n, a, n')$ → ~~is~~ the cost of action a from n to state n'

$h(n')$ → heuristic cost estimate from state n' to the goal.

consistency ensures that the heuristic is admissible and does not overestimate the true cost to reach the goal. If the heuristic is consistent, the A* algorithm is guranteed to find optimal solution.

2.a) 2017-18-3(b) What are the key advantage of local search algorithm?

Ans: The key advantage of local search algorithm are:

1. Local algorithm are simple and easy to implement.
2. It does not require the storage of an entire search tree. They only need to remember current state and move to neighbouring state.

3. It is effective for optimization problems with complex, nonlinear objective functions.

4. This algorithms are versatile and can be adapted to different types of problems.

5. It is suitable for problems where the goal is to find a local optimum rather than global optimum.

6. This algorithms iteratively improves solutions by exploring the neighbourhood of the current solution.

2.b) 2017-18 - 3(b)

c) 2017-18 - 3(c)

d) Illustrate and explain genetic algorithm using ~~(05)~~ digit strings representation of 8 queen states.

Ans:

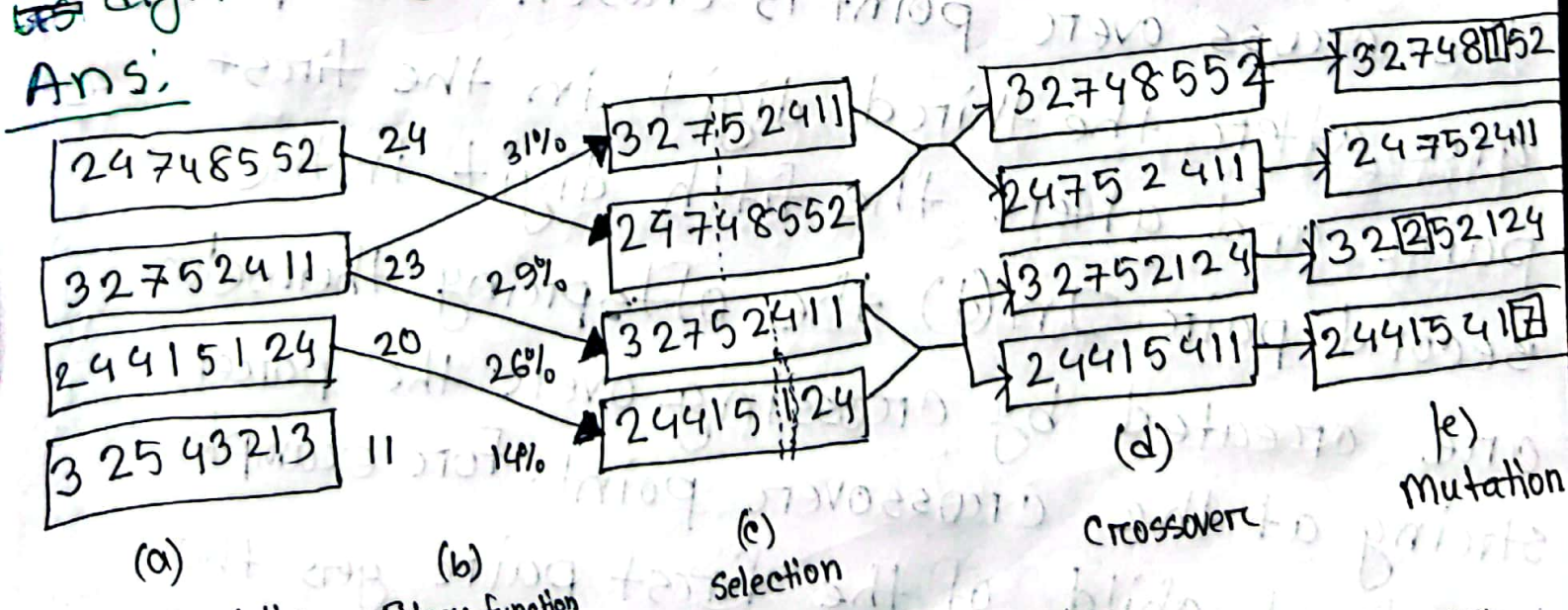
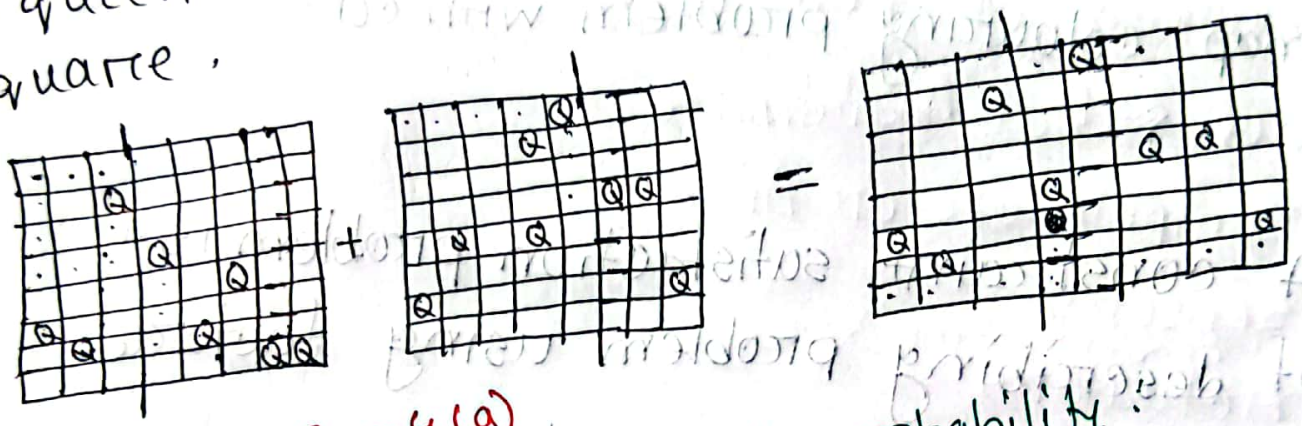


Fig: Genetic algorithm, illustrated for digit strings representing 8 queen states.

Here, fig. (a) shows a population of four 8 digit strings representing 8-queen states. In fig (b) each state is rated by objective function or fitness function. We use the number of non attacking pairs of queens, which has a value of 28 for a solution. The values for the four states are 24, 23, 20, 11. The variant of the genetic algorithm, the probability of being chosen for reproducing is directly proportional to the fitness score, the percentage are shown next to the raw scores. In (c) two pairs are selected for reproduction, in accordance with the probabilities in (b). The cross over point is chosen and points are after the third digit in the first pair and after the fifth digit in the second pair. In (d) the offspring themselves are created by crossing over the parent string at the crossover point. For example the first child of the first pair gets the

first three digits from the first parent and the remaining digit from the second parent. Finally in (e) each location is subject to random mutation with small independant probability. One digit was mutated in the first, thited, fourth offspring.

In 8 queen problem this corresponds to choosing a queen at random and moving it to a random square.



3.a) 2017-18-4(a)

Normalization, independent probability

Ans:

Normalization refers to the process of scaling and transforming the features of a dataset to a standard range.

In probability theory, events are considered to be independent if the occurance of one event does not affect the occurance of another. Independence between propositions a and b: $p(a \wedge b) = P(a)\,P(b)$, $P(a|b) = P(a)$, $P(b|a) = P(b)$

b) 2017-18 - 4(b)

c) 2017-18 - 4(c)

4.a) " - 5(a)

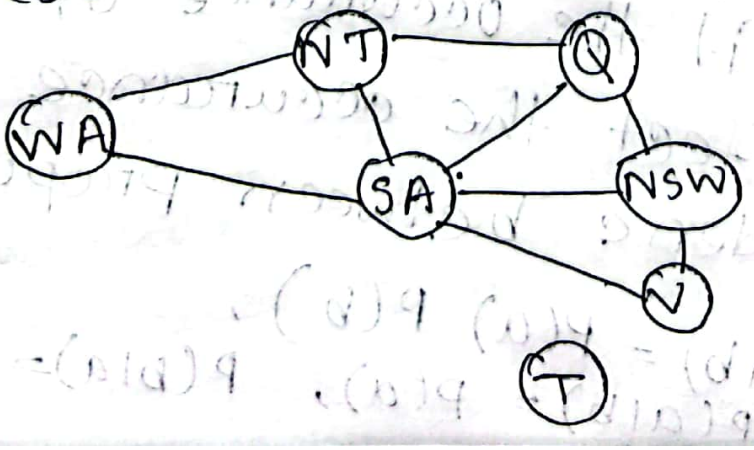
b) " - 5(c)

c) Define constraint satisfaction problem. Represent the map colouring problem with constraint graph.

Ans:

A constraint satisfaction problem is a way of describing problem using factored representation for each state; a set of variables, each of which has value that satisfies all the constraints on the variable.

Here is the map colouring problem represented as a constraint graph:



o formulate a CSP, we define the variables be the regions.

$X = \{WA, NT, Q, NSW, V, SA, T\}$

e domain of each variable is set $D_i = \{$red, green, lue$\}$

nce there are nine places where region rder, there are nine constraints!

$C = \{SA \neq WA, SA \neq NT, SA \neq Q, SA \neq NSW, SA \neq V, WA \neq NT, NT \neq Q, Q \neq NSW, NSW \neq V\}$

A ≠ WA can be fully enumerated in turn as {(red, green), (red, blue), (green, red), (green, blue), blue, red), (blue, green)}

ere are possible solution to this problem,

WA = red, NT = green, Q = red, NSW = green, V = red, SA = blue, T = red}

d) Define a robot, Briefly describe different type of robot hardware.

Ans: A robot is a programmable, multifunctional machine designed to carry out tasks autonomously. Robot can be controlled by computer programs or operate under the guidance of human operators.

Types of Robot Hardware:

1. Manipulator or Arm
2. Sensors
3. Actuators
4. control systems
5. Power supply
6. Communication device
7. Chassis or Mobile Base.

5. a) বইয়ের exercise 7.3 (a, b, c, d) (280 Page)

Ans:

i) বাদ

ii) The sentence

False $\wedge P$

True $\vee \neg P$

$P \wedge \neg P$

can be determined to be true or false in a partial model.

iii) function PL-TRUE?(s, m) returns true or false

if s = True then return true

else if s = False then return false

else if SYMBOL?(s) then return LOOKUP(s, m)

else branch on the operator of s

$\neg$: return not PL-TRUE?(ARG 1(s), m)

$\vee$: return PL-TRUE?(ARG 1(s), m) or PL-TRUE?(ARG 2(s), m)

$\wedge$: return PL-TRUE?(ARG 1(s), m) and PL-TRUE?(ARG 2(s), m)

$\Rightarrow$: (not PL-TRUE?(ARG 1(s), m) or PL-TRUE?(ARG 2(s), m)

$\Leftrightarrow$: PL-TRUE?(ARG 1(s), m) iff PL-TRUE?(ARG 2(s), m)

iv) $Q \vee \neg Q$

$Q \vee True$

$0 \wedge \neg Q$

are not detected.

b) i) $(A \wedge B) \models (A \Leftrightarrow B)$

$\Rightarrow$ True

ii) $A \Leftrightarrow B \models A \vee B$

$\Rightarrow$ False

iii) $A \Leftrightarrow B \models \neg A \vee B$

$\Rightarrow$ True.

iv) $(A \vee B) \wedge (\neg C \vee \neg D \vee E) \models (A \vee B \vee C) \wedge (B \wedge C \wedge D \Rightarrow E)$

$\Rightarrow$ True;

v) $(A \vee B) \wedge (\neg C \vee \neg D \vee E) \models (A \vee B) \wedge (\neg D \vee E)$

$\Rightarrow$ False.

vi) $(A \vee B) \wedge \neg (A \Rightarrow B)$ is satisfiable

$\Rightarrow$ model has A and $\neg B$

vii) $(A \wedge B) \Rightarrow C \models (A \Rightarrow C) \vee (B \Rightarrow C)$

= True.