22/04/24 Date __/__/____

Artificial Intelligence

AI is concerned with design of intelligence in an artificial device.

Artificial Intelligence is a branch of computer science concerned with study and creation of computer systems that exhibit some form of intelligence: systems that learn new concepts

System that can reason and draw useful conclusions about the world around us systems that can understand a natural language or perceive and comprehend a visual scene and systems that perform other types of feats that require human type of intelligence.

6/05/24

Production System

Production system in AI is a framework that assist in developing computer programs to automate wide range of task.
Production system serves as a cognitive architecture, it encompasses rules represen-ting ~~defa~~ declarative knowledge, allowing machines to make decisions and act based on different conditions. These rules consists of condition - action statements that define how the system should behave

Date __/__/____

or react when presented with certain conditions.

The simplest form of a rule based production system consists of 3 parts.

1. A knowledge base consisting of rules
2. A working memory and
3. A rule interpreter or inference engine

The interpreter inspect LHS of each rule in the knowledge base until one that matches the content of working memory is found. This causes the rules to be activated or fired in each case the content of working memory are replaced by RHS of the rule.

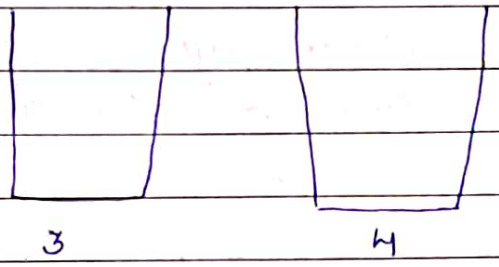
Basically, a production system consists of -

1. A set of rules · each consisting of a LHS that determine the applicability of the rule and RHS describes the operation to be performed if the rule is applied.
2. One or more knowledge base or database that contains whatever information is appropriate for particular task. Some part of database may be permanent while other part may pertain only to the solution of current problem.
3. A control strategy that specifies the order in which the rules will

rules
be applied. It compares to the database and a way of resolving conflicts that arise when several rules are matched at once.

4. A rule applier which is the computational system that implements the control strategy and applies the rules.

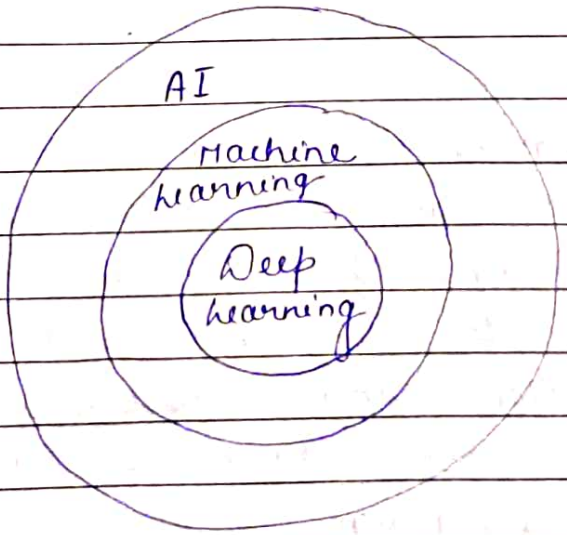
eg. Water Jug Problem.



Previous Class

1956 → John Mc Carthy

1950 - Turing Test



Supervised learning (labeled)

Unsupervised learning.

Re-enforcement learning

Date __/__/____

AI concerned with design of intelligence in an artificial device.

AI is a branch of computer science concerned with study and creation of computer system that exhibit some form of intelligence.
Systems that can learn new concept
Systems that can reason and draw useful conclusion about the world around us, the systems that can understand a natural langua-ge or perceive & comprehend a visual scene and system that performs other type of facts that require human type of intelligence

Scope of AI

1. Games : Games provides a structured task where success or failure can be measured with latest effort. Game playing shares the property that people to do them well are considered to be displaying intellig-ence.

2. Natural language Processing : It deals with analyzing, understanding and generating the languages that human uses naturally in order to interact with computers in both written and spoken content using natural human language rather than in computer language.

Date __/__/____

* Voice Recognition - It is a deep learning technique used to identify, distinguish and authenti-cate a particular person's voice. It recognises the speaker rather than what they are saying.

* Speech Recognition - It is a capability that enables a program to process human speech into written format. This technology uses machine learning and neural networks to process audio data and convert it into words that can be used in business.

3. Machine Vision - It is a concerned with hardware components and real time processing of visual information. It is a system that captures image of a given environment using special cameras. This information is then processed and used for various appli-cations such as visual inspection or object detection.

4. Computer Vision - It involves the use of algorithm and deep learning model to analyze images and video data to obtain meaning-full information. The goal of computer vision is to enable machine to perform task that would normally require human vision such as object recognition detect and track individual in a crowd and to identify cancer cell in medical image

Date __/__/____

5. Pattern Recognition - It is the process of recognizing pattern by using machine learning algorithm. Pattern recognition can be defined classification of data based an knowledge already gain on statistical information extracted from patterns and their representation. It is used in character recognition fingerprint identification, etc.

6. Theorem Proving - It has the property that people to do them well are considered to be intelligent. The logic theorist was an early attempt to prove mathematical theorem. It was able to prove several theorems.

7. Expert System - An expert system is a computer program design to act at an expert in a particular domain. Also known as knowledge based systems. An expert system typically include large knowledge based, consisting of fact about domain and for applying more facts. Expert system is design to assist experts not to replace them.

2. Searching
A general technique required when writing AI programs is search. Often there is no direct way to find a solution to problem. However, we may know to generate possibili-ties eg. in solving a puzzle we might know all possible moves but not the sequences that would lead to a solution.

Robotics Application

Date __/__/____

7/05/24

Water jug problem

You are given two jugs a 4 gallon one and a 3 gallon one. Neither has any measuring marker on it. There is a pump that can be used to fill the jugs with water how can you get exactly 2 gallons of water into the 4 gallon jug?

The state space for this problem can be described as the set of ordered pair of integers (x, y) such that $x = 0, 1, 2, 3$ or 4 and y represents the quantity of water in 3 gallon jug
The start state is $(0, 0)$.
The goal state is $(2, n)$ for any value of n.

Production rules for water jug problem.

1. Fill the 4 gallon jug.
   $(x, y) \rightarrow (4, y)$
   if $x < 4$

2. Fill the 3 gallon jug.
   $(x, y) \rightarrow (x, 3)$
   if $y < 3$

3. Pour some water out of 4 gallon jug
   $(x, y) \rightarrow (x - d, y)$.
   if $x > 0$

Date ___/___/_____

4. Pour some water out of 3 gallon jug.

$(x,y) \longrightarrow (x, y-d)$
if $y > 0$

5. Empty the 4 gallon jug on the ground.

$(x,y) \longrightarrow (0, y)$
if $x > 0$

6. Empty the 3 gallon jug on the ground

$(x,y) \longrightarrow (x, 0)$
if $y > 0$

7. Pour water from 3 gallon jug into 4 gallon jug until 4 gallon jug is full.

$(x,y) \longrightarrow (4, y-(4-x))$
if $y > 0$
and $x + y \geq 4$

8. Pour water from 4 gallon jug into 3 gallon jug until 3 gallon jug is full.

$(x,y) \longrightarrow (x-(3-y), 3)$
if $x > 0$
and $x + y \geq 3$

Date ___ / ___ / _____

9 • Pour all water from 3 gallon jug into 4 gallon jug.

$(x,y) \rightarrow (x+y, 0)$

if $y > 0$

and $x+y \leq 4$

~~and $x+y > 0$~~

10 • Pour all water from 4 gallon jug into 3 gallon jug.

$(x,y) \rightarrow (0, x+y)$

if $x > 0$

and $x+y \leq 3$

11 • Pour 2 gallons from 3 gallon jug into 4 gallon jug

$(x,y) \rightarrow (2, 0)$

if $y = 2$ &

$x = 0$

12 • Empty 2 gallons from 4 gallon jug on the ground.

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

if $x = 2$

7/05/24 Date __/__/____

Practical lisp.

(setq a 5) - setting value in a
(setq n(read)) - taking input in n
(setq x (+ a n))

Q 1. Introduction to LISP and its features

Q 2. Evaluate the following expression using LISP

i (99 + 2686 + 35) (+ 99 2686 35)
ii (5+3)*17 (* 17 (+ 5 3))
iii (104 ÷ 4)*(5+6) (* (+ 5 6) (/ 104 4))
iv (25.7 (13.5*10)) (* 25.7 (* 13.5 10))
v (169 - 32 + (222 ÷ 22)) (-(+169 (/222.0 22)) 32)

8/05/24

~~Dictionary Type~~

Case 1

	Gallons in 4 gallon jug	Gallons in 3 gallon jug	Rules Applied
{Initial}	0	0	
	0	3	2
	3	0	9
	3	3	2
	4	2	7
	0	2	5
	2	0	9

Date __/__/____

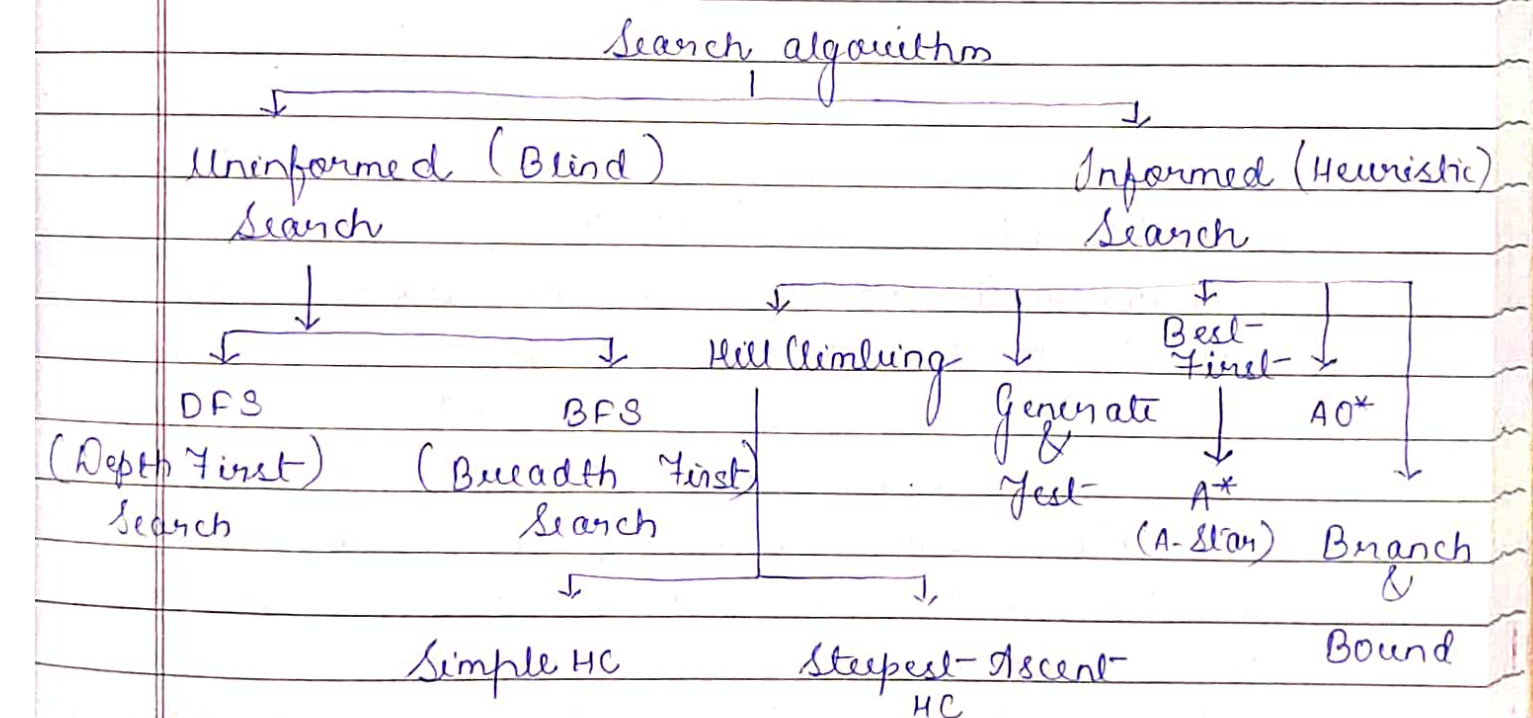
Searching.

Search is a systematic examination of states to find path from start state to the goal state. It is a search for the solution in the problem space. Search is fundamental to problem solving process.

Search process explores state space. A set of all space/states which can be reached from initial state is a search space.

Worst case the search explores all possible paths between initial state and goal state

Types of searching algorithm.



Date __/__/____

Case 2.

	Gallons in 4 gallon jug	Gallons in 3 gallon jug	Applicable Rule
initial	0	0	
	4	0	1
	1	3	8
	1	0	6
	0	1	10
	4	1	1
	2	3	8
	2	0	6
	0	2	10

14/05/24

1. Uninformed Search (Blind Search)

Uninformed search uses no information about a problem to ~~guide~~ guide the search and therefore may not be very efficient. In this search, no preference is given to the order of successor node generation and selection.

The path selected is blindly or mechanically followed. No info. is used to determine preference of one child over another.

Date __/__/____

a. BFS

Breadth first-search are performed by exploring all nodes at a given depth before proceeding to the next level.

BFS systemalically proceeds testing each node that is reachable from a parent node before it expands to any child of those nodes.

BFS explores the space in a level by level manner.

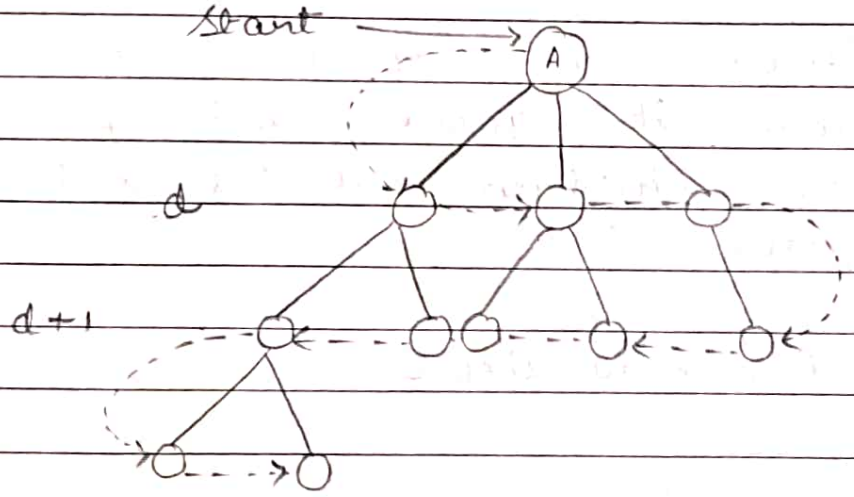


fig. BFS

All the nodes at depth 'd' in search tree are expanded before those at depth 'd+1'

BFS uses queue as a datastructure to hold all generated but still unexplored nodes.

Date __/__/____

Algorithm

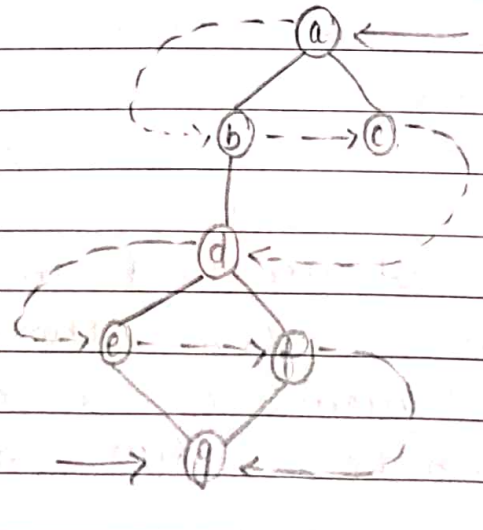
Step 1 - Place the starting node 'S' on the queue

Step 2 - If the queue is empty, return failure and stop

Step 3 - If the 1st element on the queue is a goal node 'G' return success and stop. Otherwise,

Step 4 - Remove and expand the first element from the queue and place all the children at the end of the queue.

Step 5 - Return to step 2



a c b d f e g
a b c d f e g
a b c d e f g
a c b d e f g

Queue | a |

b	c

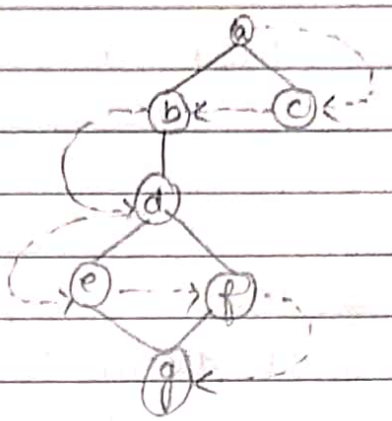
c	d

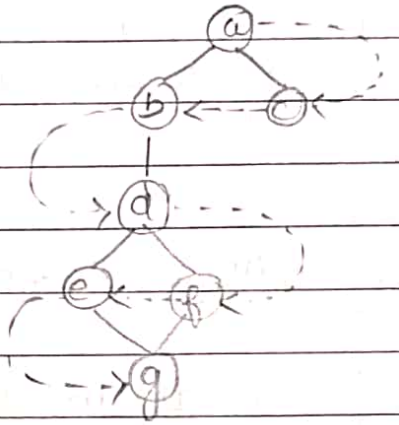
d	

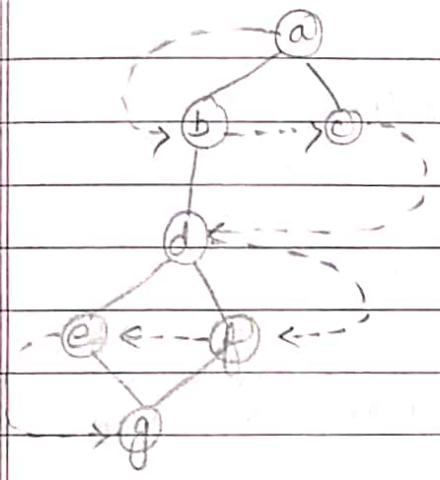
e	f

f	g

g







Practical

(1025 ÷ 5) - 20 + 60 × 10

(1025 ÷ 5) - 20 + (60 × 10)

205 - 20 + 600

805 - 20

(+ (- (/ 1025 5) 20 (* 60 10))

print- " " (Format + "label in a" (+ 4 5))

printc " "

printl " "

Date __/__/____

Q1. Write a program to find °C -to °F.

$F = \left(\frac{9}{5} \times C\right) + 32$

```
( defun temperature ()
( print " Enter the centigrade value" )
( setq c (read) )
( print (+ (* (/ 9.0 5) c) 32))
)
```

Q2. WAP to find area of circle.

```
( defun area ()
( print- " Enter the radius ")
( setq r (read))
( print- (* (/ 22.0 7) (* r r)))
)
```

15/05/24

Depth first search

A search strategy that extends the current path as far as possible before backtracking to the choice point and trying the next alternative path is called depth first search.

DFS is performed by diving downwards as quickly as possible in DFS one has to examine all children of a particular state and their descendants before

Date __/__/____

coming to any of its siblings.
DFS always expands one node till the deepest level of the tree. Only when the search hits a dead end, it goes back to the most recently expanded node and generate another of its children.
This process continues until a goal node is found or failure occurs

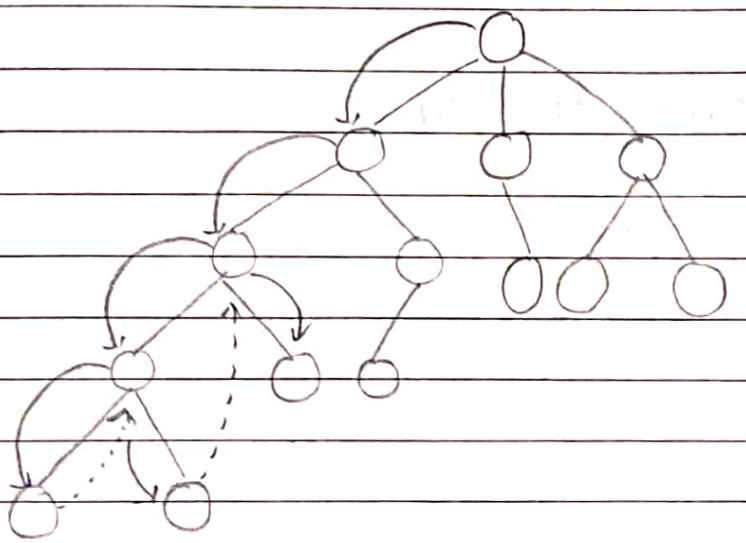


fig. DFS

DFS uses stack as a datastructure to hold all the generated nodes.

Algorithm

Step 1 - Place the starting node 'S' on the stack

Step 2 - If the stack is empty return failure

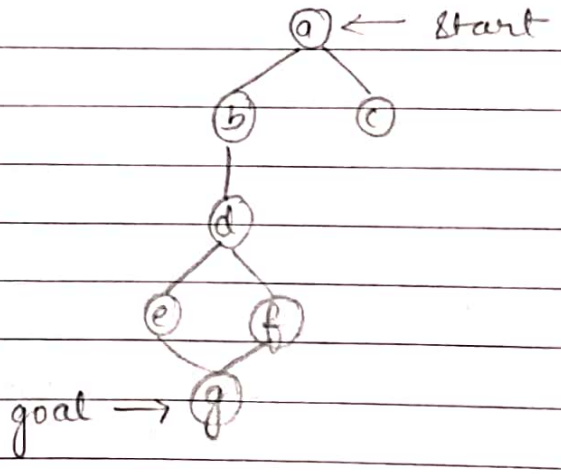
Date __/__/____

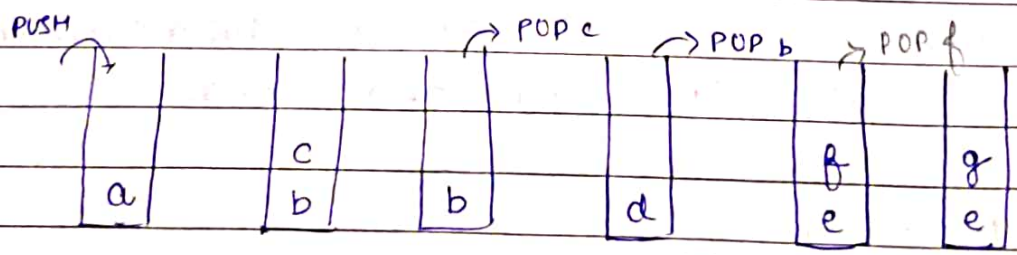
and stop

Step 3: If the 1st element on the stack is a goal node 'G' return success and stop, otherwise,

Step 4 Remove and expand the 1st element and place the children at the top of the stack.

Step 5 Return to step 2.



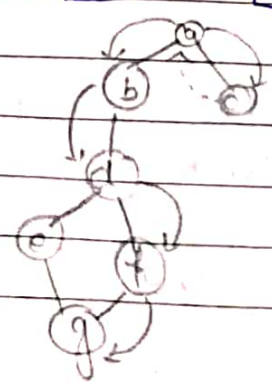


a c b d f g

a b d e g

a b d f g

a c b d e g



Date __/__/____

20/05/24

2 Heuristic Search (Informed)

Heuristic is a technique designed for solving a problem more quickly than classical methods are too slow or for finding an approximate solution when classical methods fail to find any exact solution. A heuristic is only an informed guess of the next step to be taken for solving a problem. It is often based on past experience or intuition.
It do not guarantee optimal solution but it offers solutions which are good enough most of the time

1. Hill Climbing

Hill Climbing is a variant of generate & test in which feedback from the test procedure is used to help the generator decide in which direction to move in search space.
In a pure generate & test procedure the test function responds with only yes or no but if test function is augmented with a heuristic function that provides an estimate of how close a given state is to a goal state, the generator procedure can use it to reach the goal state.
Hill climbing is often used when good heuristic function is available for evaluat

Date __/__/____

ing states but no other useful knowledg is available.

① Simple Hill Climbing

Step 1 : Evaluate the initial state if it is also a goal state then return it and quit. Otherwise, continue with initial state as the current state.

Step 2 : loop until a solution is found or until there are no new operators left to be applied in the current state.

a) Select an operator that has not yet been applied to the current state and apply it to produce a new state.

b) Evaluate the new state.

(i) If it is a goal state then return it and quit

(ii) If it is not goal state but better than the current state, then make it current state.

(iii) If it is not better than the current state, then continue in the loop.

21/05/24
Date __/__/____

② Steepest Ascent hill climbing

A useful variation of simple hill climbing which considers all the moves from the current state and selects the best one as the next state. This method is called steepest-ascent hill climbing or gradient search.

Algo

Step1:

Evaluate the initial state. If it is also a goal state then return it and quit. Otherwise continue with initial state as the current state.

Step2.

Loop until a solution is found or until a complete iteration produces no change to the current state.

a. SUCC be a state, such that any possible successor of a current state will be better than SUCC.

b. For each operator that applies to the current state to

(i) apply the operator & generate a new state

(ii) evaluate the new state, if it is a goal state then return it & quit if not, compare it to SUCC if it is better then set SUCC to this state, if it is not better leave SUCC alone.

Date __/__/____

c. If succ is greater / better than the current state to succ

Drawback of Hill Climbing.

1. Local Maxima / Minima
2. Plateau
3. Ridge

1. local Maxima

A local maxima is a state that is better than all its neighbours but it is not better than some other states that is further away. That a local maxima all moves appear to make things worse.

Local maxima's are frustrating because they often occur almost within the site of the solution. In this case they are called foothills ~~backtrack to some earlier~~

2. Solution of local maxima

Backtrack to some earlier node and try going to a different direction.

2. Plateau

A plateau is a flat area of the search space in which a goal set of neighbouring states have the same value.

On a plateau it is not possible to determine the best direction in which

Date __/__/____

to move by making local comparisons.

Solution

Make a big jump in some direction to try to make a new section of the search space.

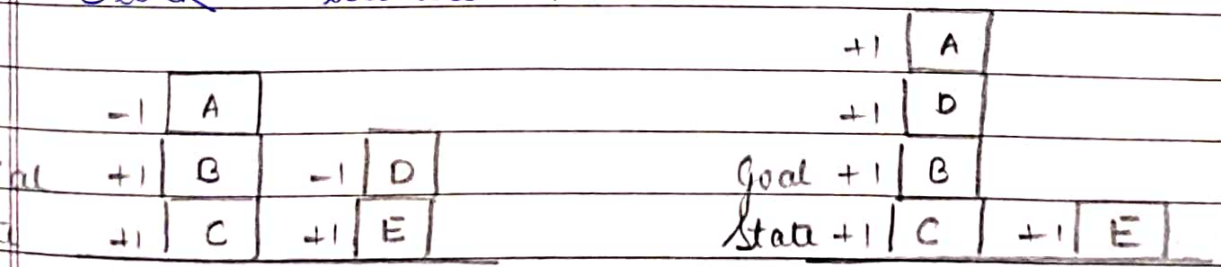
3. Ridge

A ridge is a special kind of local maxima. It is an area of search space that is higher than the surrounding areas and that itself has a slope

Solution

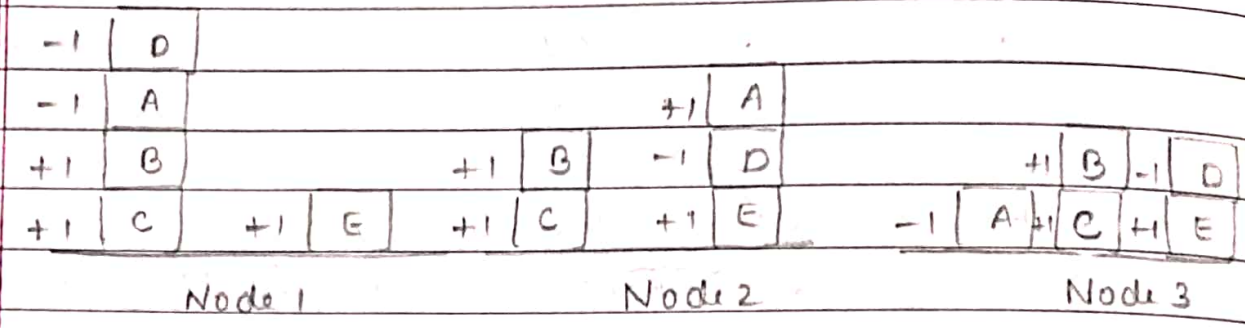
Apply two or more rule before doing a test. this corresponds to moving in several directions at once.

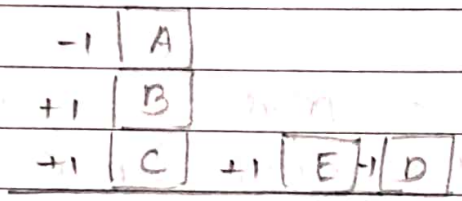
Block World Problem



$h_1(n)$ = add 1 if the block is on correct block or table and subtract 1 if it is in wrong location

Date ___ / ___ / _____


-1 D
-1 A
+1 B
+1 C
+1 E
+1 A
+1 B
-1 D
+1 C
+1 E
+1 B
-1 D
-1 A
+1 C
+1 E
Node 1
Node 2
Node 3


-1 A
+1 B
+1 C
+1 E
+1 D
Node 4

22/05/24

Date __/__/____

Best first search

Best first search also depends on the use of heuristic to search and select the most promising path to the goal node. Unlike hill climbing this algorithm retains all estimates computed for previously generated nodes and makes its selection based on the best among them all. Thus, at any point in the search process, best first moves forward from the most promising of all the nodes generated so far. In doing so it avoids the potential traps encountered in the hill climbing

Algorithm of Best First Search

Step 1:
Place the starting node S on the queue

Step 2:
If the queue is empty return failure and stop

Step 3:
If the first element on the queue is a goal node G return success and stop. otherwise,

Step 4:
Remove the first element from the queue, expand it and compute the estimated goal distances for each child. Place the children on the queue and arrange all queue elements in ascending order corresponding to goal distance

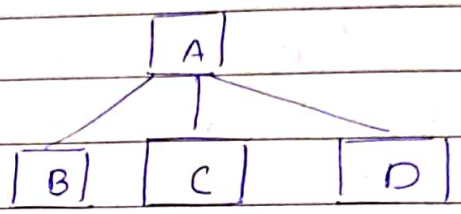
Date __/__/____

from front of the queue.

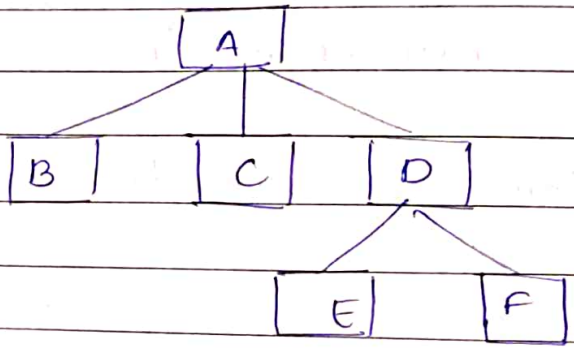
Step 5:
Return to Step 2.

eg Step 1.

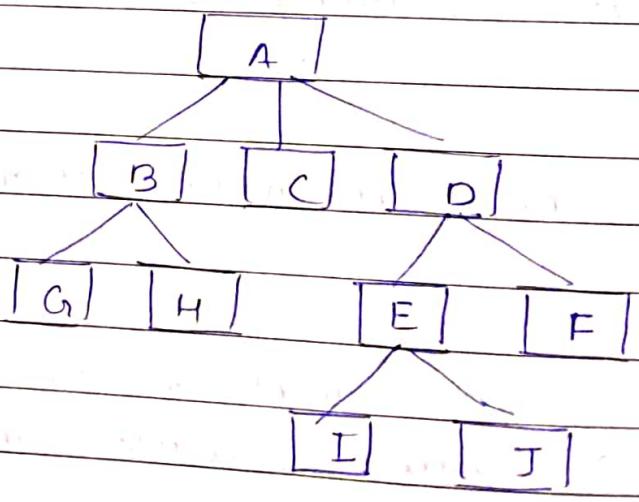
Step 2.



Step 3



Step 4



Date __/__/____

Minimax search

Minimax is a concept of game theory. Game theory allows decision makers to cope with other players who have different purposes in mind. In other words players determine their own strategies in terms of strategy and goals of their opponent. In minimax ~~their~~ players adopt those strategies which will maximize their gains while minimizing their process. Therefore the solution is, the best each player can do for himself in the face of opposition of the other player.

Steps for picking up the next move.

Step 1 - Since, it's done to move for player one the start node is max node with current board configuration.

Step 2 - Expand nodes down to some depth of look ahead in the game.

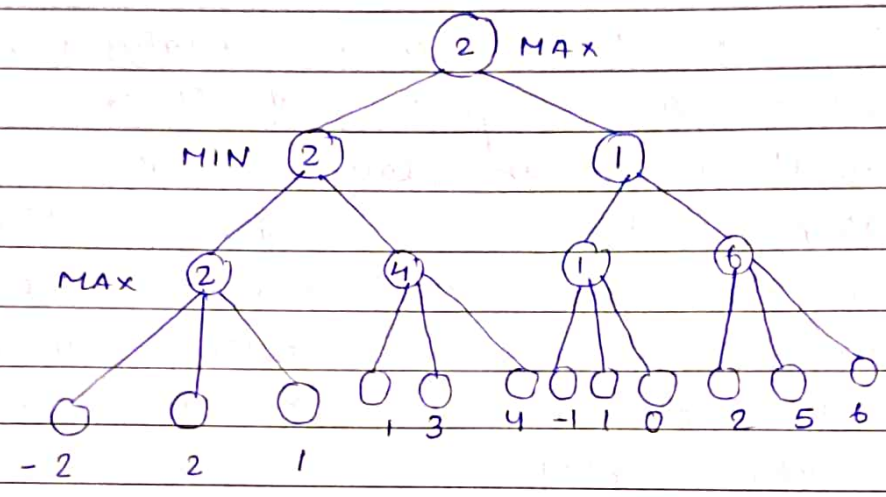
Step 3 - Apply evaluation function at each leaf node.

Step 4 - Backup values for each non-leaf nodes untill computed for the root node.

Step 5 - At min node the backed up values is the min of the values associated with its children.

Date ___/___/_____

Step 6 - At max node the backed up value is the max of values associated with its children.



max $(-\infty, -2) = -2$
max $(-2, 2) = 2$
max $(2, 1) = 2$

max $(-\infty, 1) = 1$
max $(1, 3) = 3$
max $(3, 4) = 4$

max $(-\infty, -1) = -1$
max $(-1, 1) = 1$
max $(1, 0) = 1$

max $(-\infty, 2) = 2$
max $(2, 5) = 5$
max $(5, 6) = 6$

min $(\infty, 2) = 2$
min $(2, 4) = 2$

min $(\infty, 1) = 1$
min $(1, 6) = 1$

max $(-\infty, 2) = 2$
max $(2, 1) = 2$

Date ___ / ___ / _____

Alpha-beta Pruning

The problem with minimax search is that the no. of states it has to examine is exponential in the number of moves. The 'α-β' pruning helps to arrive at correct minimax algorithm decision without looking at every node of the game tree.

MAX player cut off search when he knows MIN player can force probably bad outcome.
MIN player cut off search when he knows MAX player can force probably good outcome.

Applying alpha cutoff means, we stop search of a particular branch because we see that we already have a better opportunity else where.
Applying beta cut off means, we stop search of a particular branch because we see that a opponent already has a better opportunity elsewhere.
Applying both forms is alpha-beta pruning.
It can be applied to any depth of the tree sometimes it is not only prune the Tree leaves but also entire sub tree.
The main condition required for alpha-beta pruning is " $\alpha \geq \beta$ "

$MAX = \alpha$ $MIN = \beta$

Date __/__/____

A* search

A* algorithm combines the feature of uninformed first-search and pure heuristic search to efficiently compute optimal solution.

A* algorithm is a best first search algorithm in which the cost associated with a node is $f(n) = g(n) + h(n)$

where

$g(n)$ - the cost of the path from initial state to node n

$h(n)$ - is the heuristic estimate on the cost of a path from node 'n' to goal node.

Thus $f(n)$ estimates the lowest total cost of any solution for going through node n. At each point a node with lowest f-value is chosen for expansion.

Algorithm

1. Create a search graph 'G' consisting of start node 'n_0' put 'n_0' on a list called OPEN.
2. Create a list CLOSED that is initially empty.
3. If OPEN is empty, exeed with failure.

Date ___ / ___ / _____

4. Select the first node on OPEN, remove it from OPEN & put it on CLOSED, call this node 'n'.

5. If 'n' is a goal node then exit successfully with the solution obtained by tracing a path along the pointers from n to no in Gr.

6. Expand node 'n' generating the set M of its successor that are not already ancestors of 'n' in 'Gr'.
Install these members of M as successors of 'n' in 'Gr'.

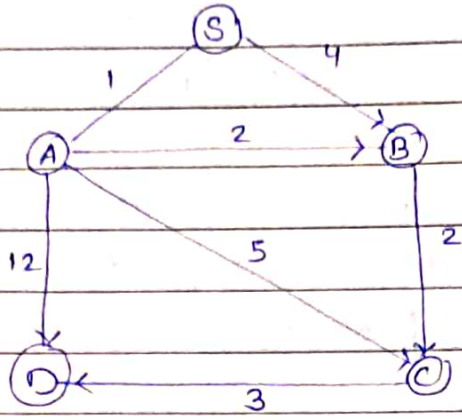
7. Establish a pointer to 'n' from each of those members of M that were not already in Gr. Add these members a M to OPEN for each member m of M that were already in OPEN or CLOSED, redirect its pointer to n if the best path to 'm' found so far is to n.
For each member of M already on CLOSED, redirect the pointers of each of its descend-ants in Gr so that they point backward along the best path found so far.

8. Reorder the list OPEN in order of increasing f' value.

9. Goto step 3.

Date __/__/____

Q9



S	7
A	6
B	2
C	1
D	0

goal

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

1. $S = \underset{g(n)}{0} + \underset{h(n)}{7} = 7$

2. $S \to A = 1 + 6 = 7$
   $S \to B = 4 + 2 = 6$

3. Choose $S \to B$ for expansion
   $(S \to B) \to C = (4+2)+1 = 7$
   Now in OPEN ; $S \to A = 7$
   $(S \to B) \to C = 7$
   Choose any one of them for expansion.
   expanding $S \to A$

4. $(S \to A) \to B = (1+2)+2 = 5$
   $(S \to A) \to C = (1+5)+1 = 7$
   $(S \to A) \to D = (1+12)+0 = 13$

   Expanding $(S \to A) \to B$

5. $S \to A \to B \to C = 6$

Date ___ / ___ / _____

1. Expanding $S \to A \to B \to C$

$S \to A \to B \to C \to D = (1+2+2+3)+0 = 8$

	OPEN	CLOSED
1.	$S \to A = 7$ $S \to B = 6$	
2.	$S \to A = 7$ $(S \to B) \to C = 7$	$S \to B = 6$
3	$S \to B \to C = 7$ $S \to A \to B = 6$ $S \to A \to C = 7$ $S \to A \to D = 13$	$S \to A = 7$
4	$S \to B \to C = 7$ $S \to A \to C = 7$ $S \to A \to D = 13$ $S \to A \to B \to C = 6$	$S \to A \to B = 5$
5	$S \to B \to C = 7$ $S \to A \to C = 7$ $S \to A \to D = 13$ $S \to A \to B \to C \to D = 8$	$S \to A \to B \to C = 6$
6	$S \to B \to C = 7$ $S \to A \to C = 7$ $S \to A \to D = 13$	$S \to A \to B \to C \to D = 8$
7	$S \to B \to C \to D = 9$ $S \to A \to C = 7$ $S \to A \to D = 8$	$S \to B \to C = 7$

24/06/24 Date __/__/____

Constraint Satisfaction Problem

Constraint- satisfaction is a search procedure that operates in a space of constraint- set.

The initial state contains the constraints that are originally given in the problem description

A goal state is any state that has been constrained enough, where constrained enough must be difined for each problem.

Constraint- satisfaction is a 2 step process first-, constraints are discovered & propagated as far as possible ~~thro~~ to the system when if there is still not a solution, search begins a guess about something is made and added as a new constraint. Propagation can then occur with this new constraint and so forth

example.

Cryptarithmetic Problem

Cryptarithmetic Problem represented is letters assign a decimal digit to each of the letters in such a way that the answer to the problem is correct. If the same letter occurs more than once it must be assigned the same digit each time. No two diff.

Date ___ / ___ / ____

letters may be assigned the same digit.

eg1.

$C_4\ C_3\ C_2\ C_1$

```
  E A T
+ T H A T
---------
A P P L E
```

A = 1
T = 9
P = 0
E = 8
L = 3
H = 2

```
  8 1 9
+ 9 2 1 9
---------
1 0 0 3 8
```

eg 2.

$C_4\ C_3\ C_2\ C_1$

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

M = 1
~~S = 9~~
~~O = 0~~
R = 8
N = 6
E = 5
D = 7
Y = 2

$E + 1 = N$

$N + R(+1) = E + 10$

$E + 1 + R(+1) = E + 10$

$R(+1) = 9$

$R = 8$

Date __/__/____

25/06/24 Genetic Algorithm

A genetic algorithm is a randomized search & optimization technique guided by the principle of natural genetic system. Genetic algorithm is based on models of natural adaptation & evolution.
Genetic algorithms are part of evolutionary computing the rapidly growing area of AI.
GA are adaptive heuristic search algorithm based on evolutionary ideas of natural selection and genetics. It is inspired by Darwin's theory 'survival of fittest'
In nature competition among individual for scanty resources results in the fittest individuals dominating over weaker ones.
GA are intelligent exploitation of random search and used in optimization problems. It also uses historical infor. to direct the search into the region of -dat. better perfor -mance within search space.
In GA the process of finding solutions generates other points or possible solutions as evolution proceeds.
It do not break easily even if the input is changed slightly. It offers significant benefits over other difficult search optimization techniques.

Date __/__/____

GA operators

1. Selection
2. Crossover
3. Mutation

1. Selection

The process that determines ~~fit~~ which solution are to be preserved and allowed to reproduce and which one deserve to ~~perish~~ die out.

The primary objective of selection operator is to emphasize the good solution and eliminating the bad solution in a population while keeping the population size constant. This operation selects the best & discards the rest.

Selection means extracting a subset of genes from an existing population a/c to any definition of quality. Every gene has a meaning so one can derive from the gene a kind of quality measurement called fitness function. Selection can be performed by fitness value.

2. Crossover

Crossover is a genetic operator that combines 2 chromosomes to produce new chromosomes.

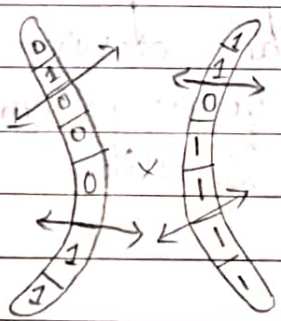
The idea behind crossover is that

Date ___ / ___ / _____

the new chromosome may be better than both of the parents if it takes best characteristics from each of the parent.
Mating between two strings is accomplished with crossover operation which randomly selects a bit position in a string of bits & coordinates the head of one parent to the tail of another parent to produce offspring.

Suppose,
Parent 1: XXX|XXXXX
Parent 2: YYY|YYYYY
Offspring 1: XXXYYYYY
Offspring 2: YYYXXXXX

3rd bit position is selected for crossover



Parent 1 Parent 2

fig. two-point crossover

Offspring 1 : 0101111

Date __/__/____

Offspring 2 : 1100011

ASSIGNIMENT

1. WAP in LISP to check whether two given numbers are even, odd or there is a mixture of two.

equal ( )
eql ( )
evenp ( )
oddp ( )
fp zerop ( )
plusp ( )
minusp ( )
null ( )
listp ( )
atom ( )

returns either T or nill.

2. WAP to generate no.s from i to j by taking input from user.

3. WAP to print even no.s from a given no to a given limit.

4. WAP to generate multiplication table of any given number.

Date __/__/____

2. (defun printnumbers (i j)
(
loop for a from i to j
do (write a)(terpri)))

(printnumbers { 5 12)

26/06/24

3. Mutation

Mutation is the occasional introduction of new feature ~~de~~ into the solution string of the population pool. To maintain diversity in the population. It is used to ensure that all ~~alleles~~ locations of the rule space are reachable that every potential rule in the rule space is available for evaluation. This ensures that selection process does not get caught in local minimum or local optimum. It may happen that crossover and inversion operator will only produce a set of structures that are better than local neighbours but not optimal in global sense, so the mutation operator can overcome this by simply selecting any bit position in a string at random and changing it. for example.

Date __/__/____

Original Offspring : 00101101

Mutated Offspring : 00111101

Algorithm

Step 1 :
Create a population of 'n' elements

Step 2 :
Generate the fitness of each element of a population by fitness function

Step 3 :
Repeat 'n' times -
a. Pick two parents with probability according to relative fitness.
b. Crossover
c. Mutation
d. Add new child to ~~the~~ new population

Step 4 :
Replace the old population with new population & return to step 2.

eg. Consider the problem of finding a global maximum of the following function.

$f_1 : \{0, 1, \ldots, 31\} \to R$

$x \to x^2$

Date ___ / ___ / _____

Initial population which is randomly selected are 13, 24, 8, 19

No. of individual	String (genotype)	x value (phenotype)	$f(x) = x^2$
1	01101	13	169
2	11000	24	576
3	01000	8	64
4	10011	19	361
			$\Sigma f_i = 1170$

$P_i = \frac{f_i}{\Sigma f_i}$	Expected Count ($P_i \times$ No. of individual)	→ (fitness value)
0.144	0.57	
0.492	1.97	
0.054	0.22	
0.308	1.23	
	4.0	

No. of individual	String (genotype)	Crossover point (randomly selected)
1	0110 | 1	4
2	1100 | 0	4
2	11 | 000	2
4	10 | 011	2

Date __/__/____

New Population	x	$f(x) = x^2$
01100	12	144
11001	25	625
11011	27	729
10000	16	256

Assignment AI { Submission Date : 1 July }

1. What is AI? What are its application area?
2. Differentiate between DFS & BFS?
3. What do you mean by Heuristic search? Explain any one heuristic search with eg.
4. What is fuzzy set? Explain different types of operations and functions used in fuzzy logic.
5. What is an expert system? Explain the architecture of expert system.

Date __/__/____

Initial population which is randomly selected are 13, 24, 8, 19

No. of individual	String (genotype)	x value (phenotype)	$f(x) = x^2$
1	01101	13	169
2	11000	24	576
3	01000	8	64
4	10011	19	361
			$\Sigma f_i = 1170$

$P_i = \frac{f_i}{\Sigma f_i}$	Expected Count (P_i × No. of individual)	→ (fitness value)
0.144	0.57	
0.492	1.97	
0.054	0.22	
0.308	1.23	
	4.0	

No. of individual	String (genotype)	Crossover point (randomly selected)
1	0110 | 1	4
2	1100 | 0	4
2	11 | 000	2
4	10 | 011	2

Date ___ / ___ / _____

New Population	x	$f(x) = x^2$
01100	12	144
11001	25	625
11011	27	729
10000	16	256

Assignment AI { Submission Date : 1 July }

1. What is AI? What are its application area?

2. Differentiate between DFS & BFS?

3. What do you mean by Heuristic search? Explain any one heuristic search with eg.

4. What is fuzzy set? Explain different types of operations and functions used in fuzzy logic.

5. What is an expert system? Explain the architecture of expert system.

27/06/24 Date __/__/____

Neural networks.

A neural network is an artificial represen-tation of human brain that tries to simulate its learning process. A artificial neural network (ANN) is often called neural network or simply neural net.

An ANN is composed of a no. of interconnected units. Each unit has an input output characteristics and implim-ents a local configuration of function. The output of any unit is determined by its input output characteristics its interconnection with other unit and possible external input. The network usually develop an overall functionality through one or more forms of training.

A common topology used for representing ANN is weighted directed graph. The nodes in the graph can be on or off at any instance of time.

At each time all the 'on' nodes send an impulse along their outgoing arcs to the neighbouring nodes. All the nodes turn their incoming impulses weighted according to the arcs. All the nodes which exceed the threshold turn on at the next instant at all the other next turn off.

Date __/__/____

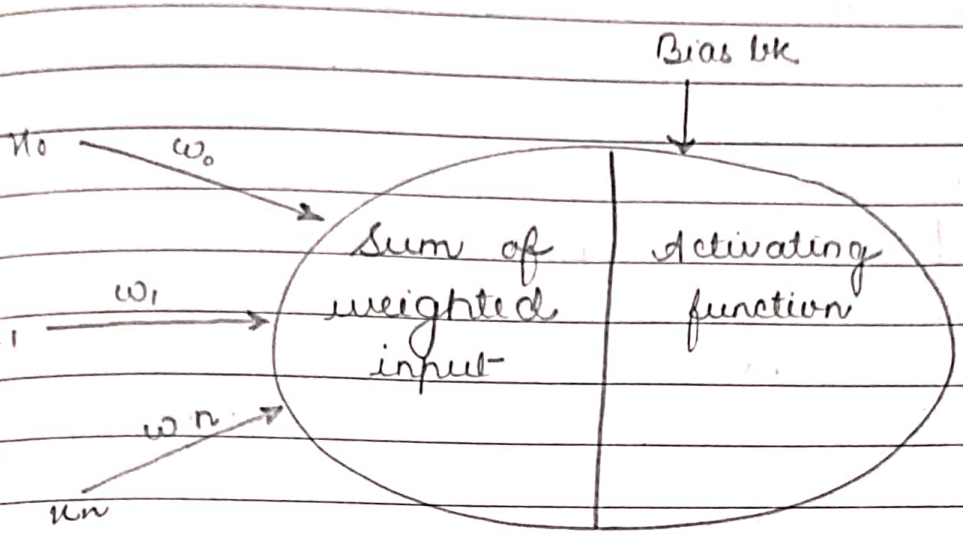


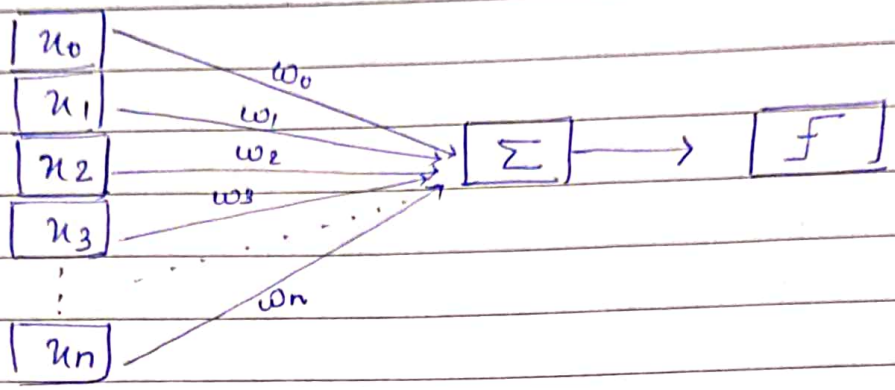
fig. Artificial Neuron

The processing element consists of two parts. The first part consists of adder for summing of input signals, weighted by their respective synapsis of a neuron. And the second part consists of an activation function. The activation function is denoted by y_i, defines the output of the neurons in terms of individual localfield 'I'. The threshold function passes information only when the output of it part of artificial neurons exceeds the threshold value 'T'.

$$\phi I = \begin{cases} 1 & \text{if } I \geq 1 \\ 0 & \text{if } I < 0 \end{cases}$$

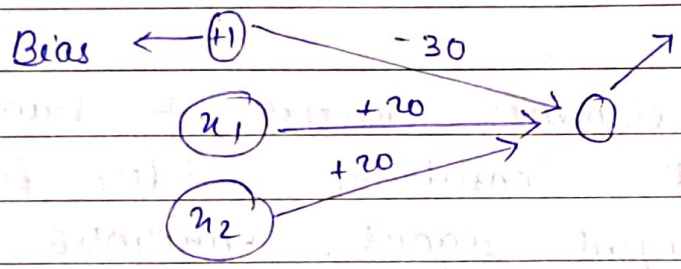
Date __/__/____

Perceptron



eg.

$x_1, x_2 \in \{0, 1\}$



$h = \{ w_1 x_1 + w_2 x_2 + b_k w_0 \}$

x_1	x_2	h_θ
0	0	$(-30)\,0$ { if -ve put 0
0	1	0
1	0	0
1	1	$(10)\,1$ { if +ve put 1

therefore,

$y = x_1$ AND x_2

25/06/24 Date __/__/____

An ANN can be defined as a data processing system consisting of a large no. of simple highly interconnected processing elements. (Artificial neurons). These artificial neurons are usually org. in a sequence of layers with full or random connections b/w layers. The set of nodes that receive the unprocessed signal from the input data constitute the first layer of nodes. The set of hidden nodes which receive the outputs from the nodes of first layer of nodes constitute the second layer of nodes. Similarly, we can define third, fourth etc layers.

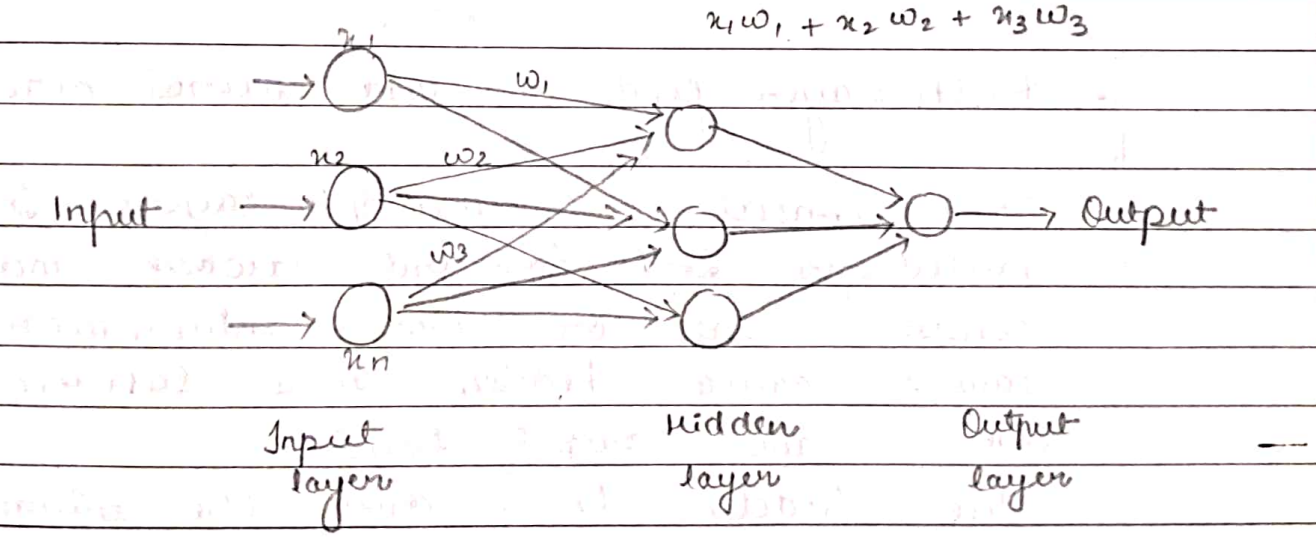


fig.

Type of neural network

- Single layer feed forward neural network.

This network consists of a single layer of weights, where the inputs

Date __/__/____

are directly connected to the output

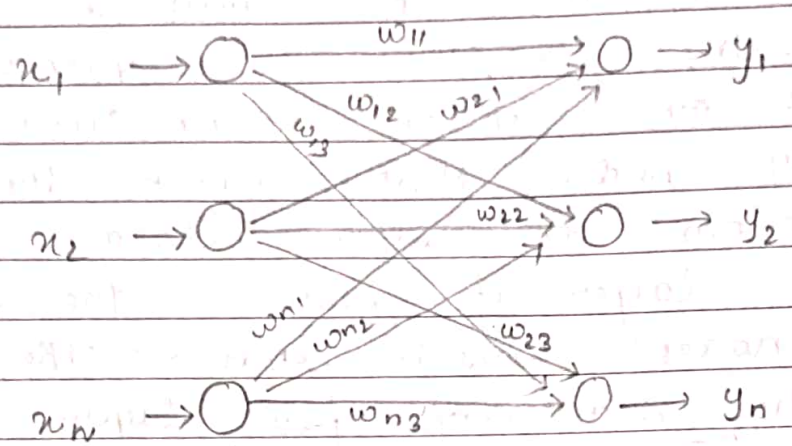
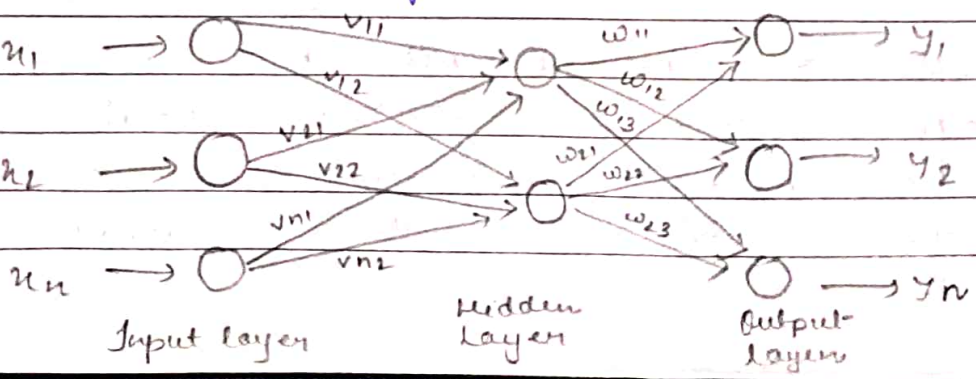


fig. Single layer feed forward neural network

- Multi layer feed forward neural network.

It consists of multiple layers. In multilayer feed forward network may consist one or more intermediary layers called hidden layers between input and output layer.
The hidden layer does the intermediate computation before directing input to output layer.



Date ___/___/___

- Recurrent Neural Network

Recurrent Neural Network has atleast one feedback loop there could be neurons with self-feedback links i.e the output of a neuron is fed back into itself as input.

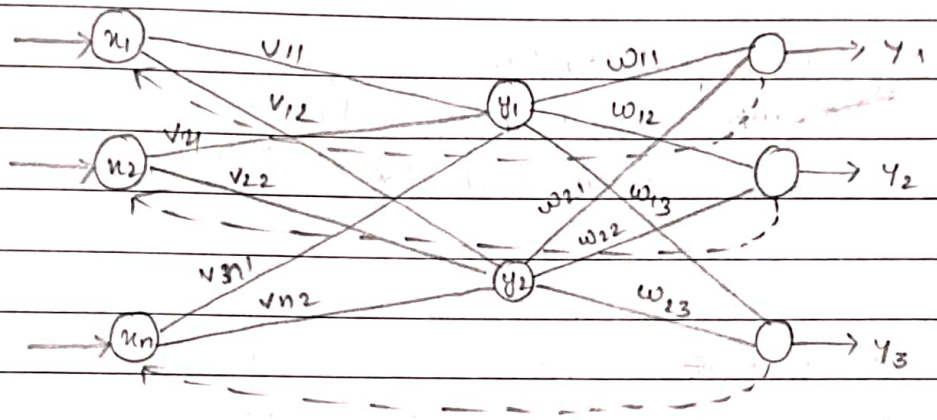


fig. Recurrent Neural Network

1. WAP to find factorial of any number using recursion.

2. Write a function to draw a pattern like
   ×××××
   ×××××××
   ×××××
   ×××××

Date ___ / ___ / _____

1.
```
( defun factorial (n)
 ( if ( = n 0)
    1
    ( * n ( factorial (- n 1))))
)
```

2.
```
* * * * *
```

```
( defun pattern(n)
 ( loop for i from 1 to n
    do ( format t "* ")))
```

3.
```
* * * * *
* * * * *
* * * * *
```

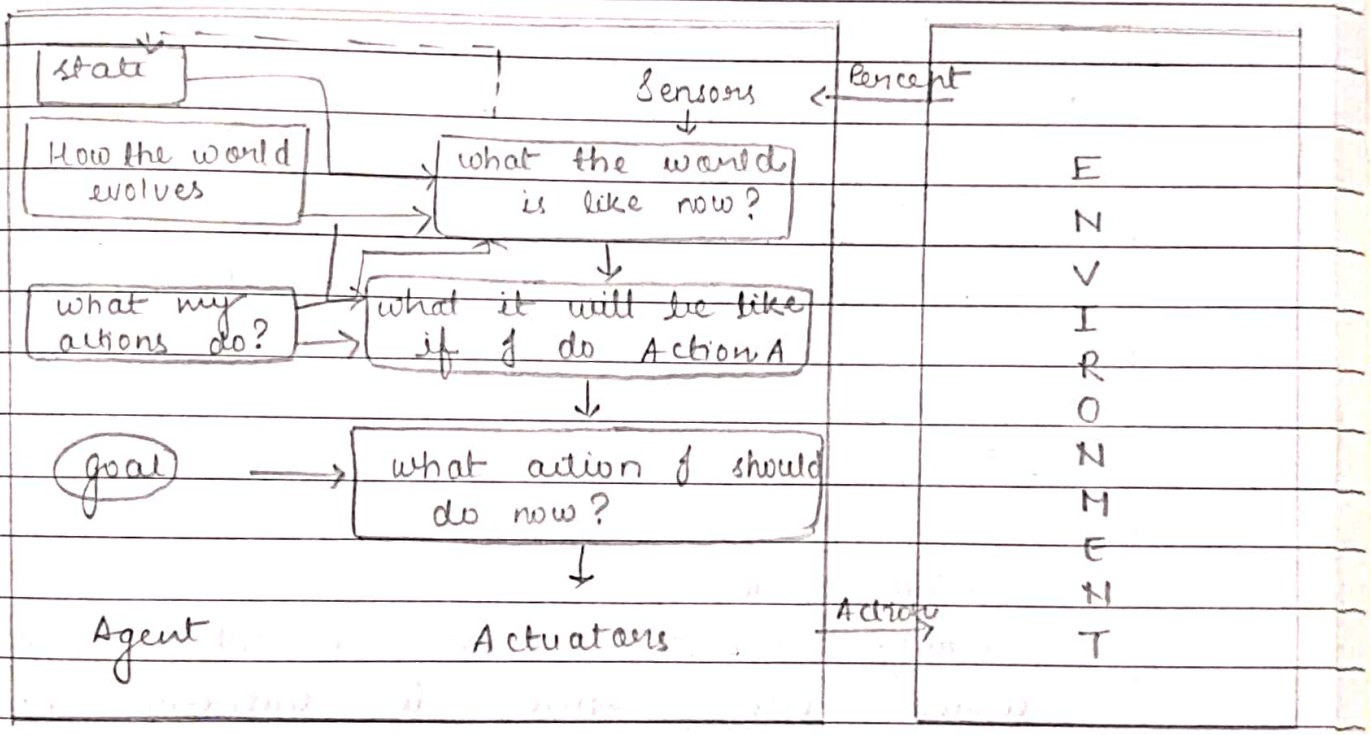
```
( defun pattern1 (x, y)
 ( loop for i from 1 to y
    do ( loop for j from 1 to x
       do ( format t "* "))
       (terpri)
    ))
```

1/07/24

Date __/__/____

Goal Based Agent-

Goal Based Agent- expand the capabilities of model based agent by using goal info.
Goal information describes situations that are desirable. This allows the agent a way to choose among multiple possibilities, selecting the one which reaches a goal state.

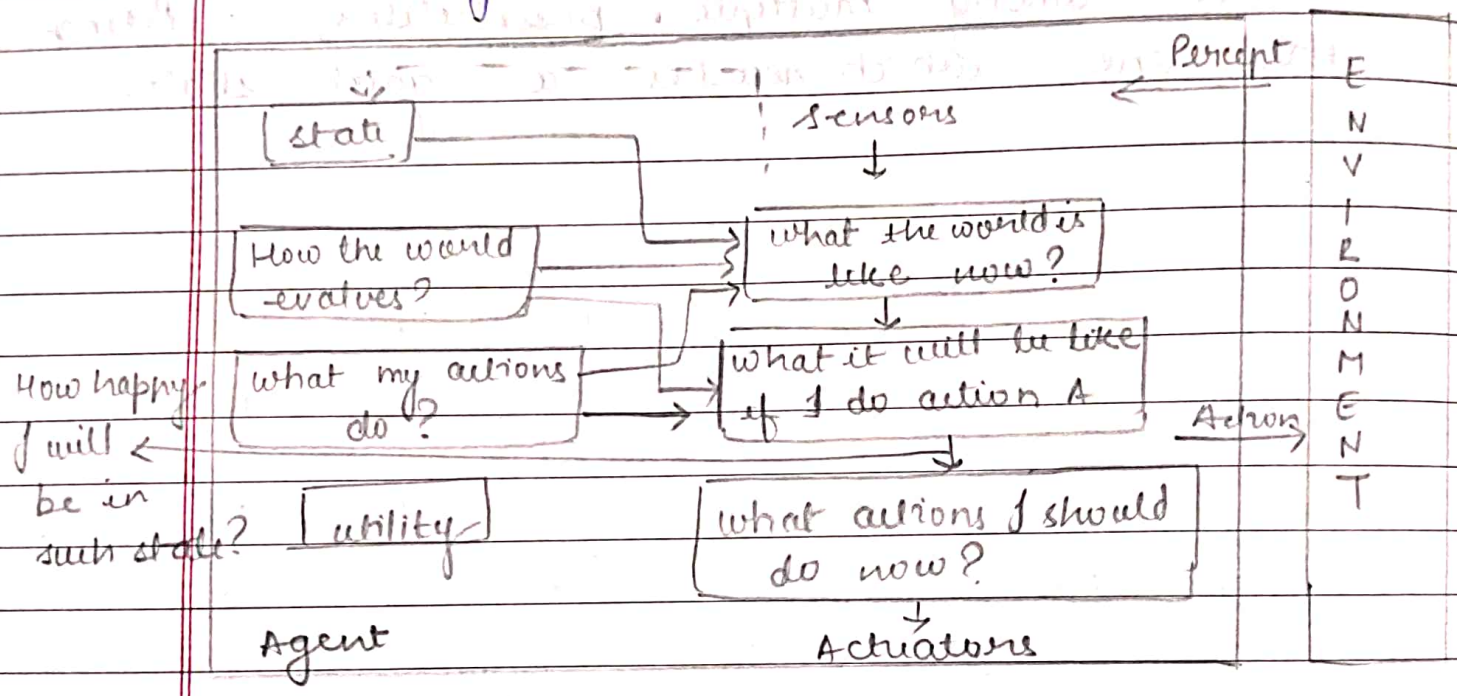


Utility Based Agent-

Goal based agents only distinguish b/w goal states and non-goal states. It is possible to define a measure of how desirable a particular state is. This measure can be obtained through the use of a utility functions which maps a state to a

Date __/__/____

measure of the utility of the state. A more general performance measure should allow a comparison of different world states according to exactly how happy they would make the agent

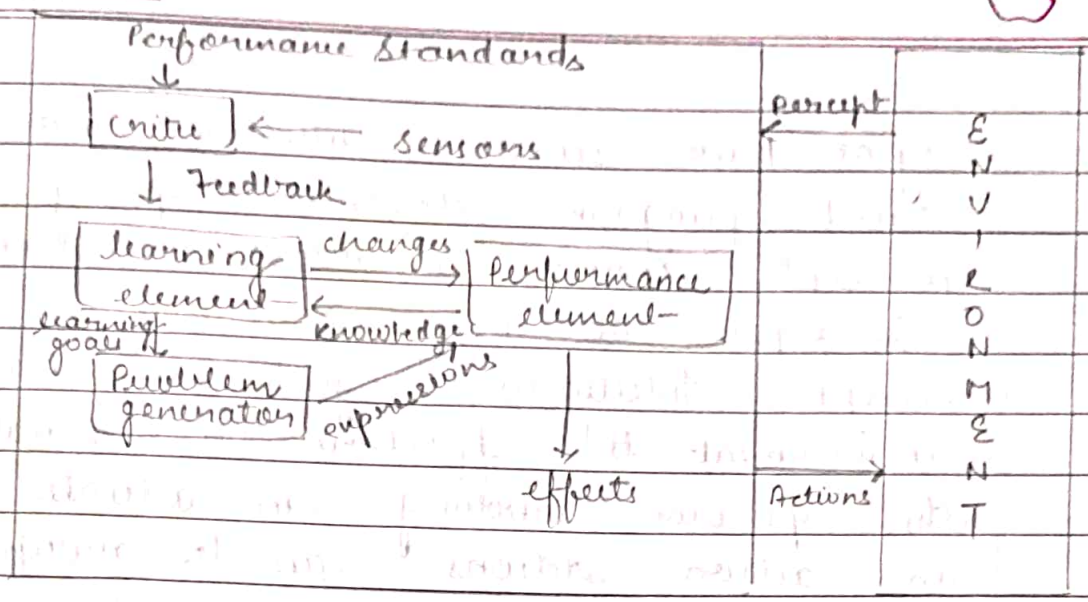


learning agent

learning agent has an advantage that it allows the agent to initially operate in unknown environment to become more confident than its initial knowledge alone might allow.

The most important decision b/w the learning agents which is responsible for making improvements and performance element for selecting ext. actions

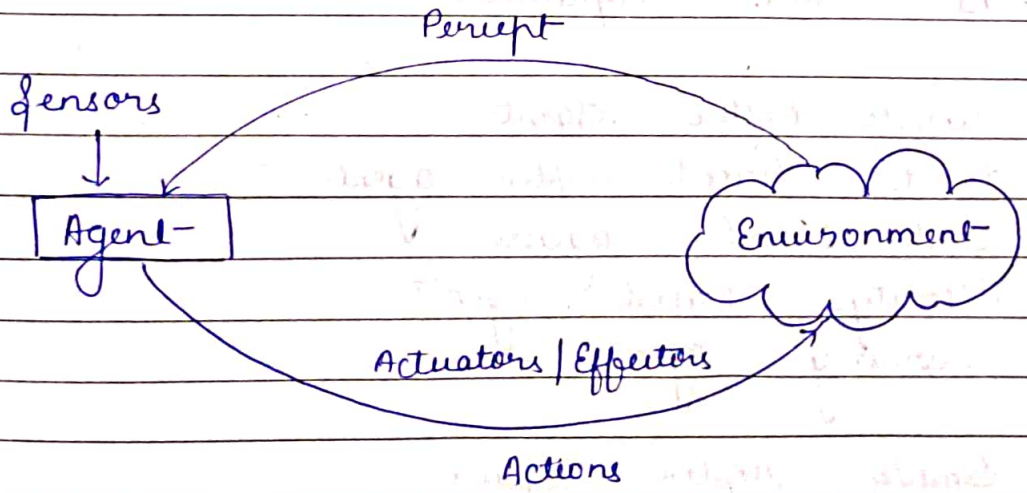
Date ___ / ___ / _____



29/06/29

Intelligent Agent-

Intelligent Agent is an autonomous entity which observes through sensors and act upon an environment using actuators and direct its activity towards achieving goal. Intelligent agent may also learn or use knowledge to achieve their goal. It may be very simple or may be very complex.



Date __/__/____

Agent have sensors, actuators and have goals. Agent program elements mapping from percept sequence to actions. A complete set of input at any given time is called percept. Intelligent agent can change the environment to effectors or actuators. An operation involving an actuator is called an action. Actions can be grouped into action sequences.

PEAS represents the foundational component that define AI agents behaviour. PEAS stands for P- performance measure
E- Environment
A - Actuators
S - Sensor

Intelligent agent are grouped in 5 classes based on their degree of perceived intelligent and capabilities.

1. Simple Reflex Agents
2. Model based reflex agents
3. Goal based agents
4. Utility based agents
5. Learning agents.

1. Simple reflex agents
SRA act only on the basis of current percept, ignoring rest of percepts history.

Date __/__/____

Agent have sensors, actuators and have goals. Agent program elements mapping from percept sequence to actions. A complete set of input at any given time is called percept. Intelligent agent can change the environment to effectors or actuators. An operation involving an actuator is called an action. Actions can be grouped into action sequences.

PEAS represents the foundational component that define AI agents behaviour. PEAS stands for P- performance measure
E- Environment
A - Actuators
S - Sensor

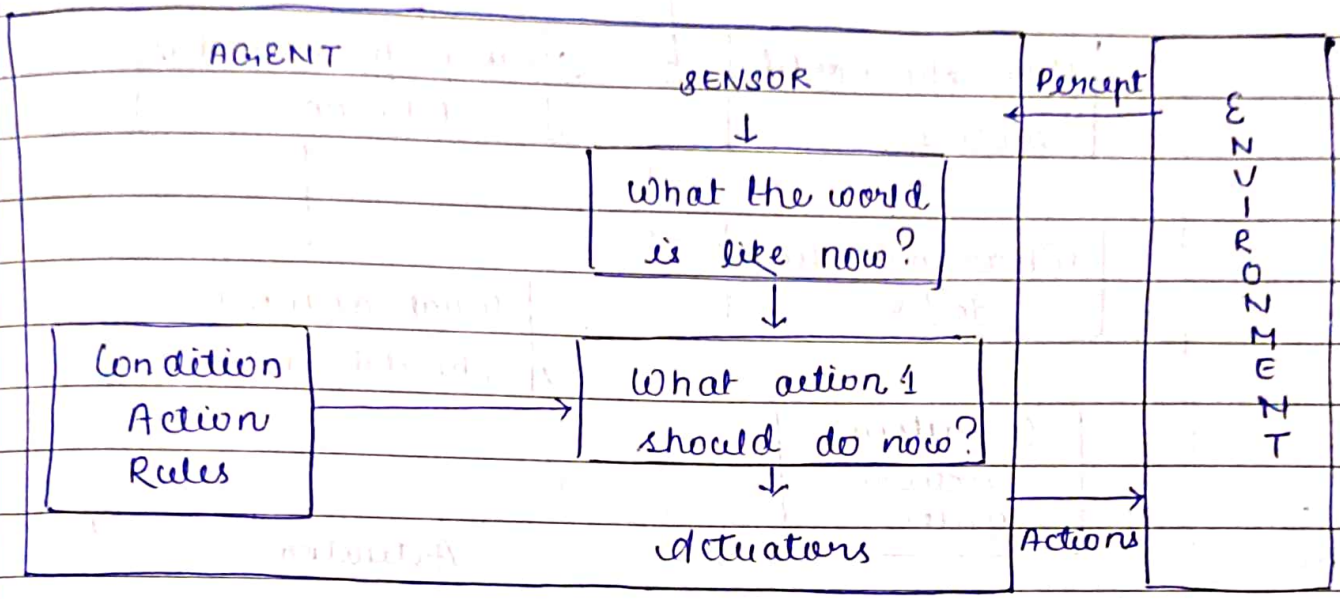
Intelligent agent are grouped in 5 classes based on their degree of perceived intelligent and capabilities.

1. Simple Reflex Agents
2. Model based reflex agents
3. Goal based agents
4. Utility based agents
5. learning agents.

1. Simple reflex agents
SRA act only on the basis of current percept, ignoring rest of percepts history.

Date __/__/____

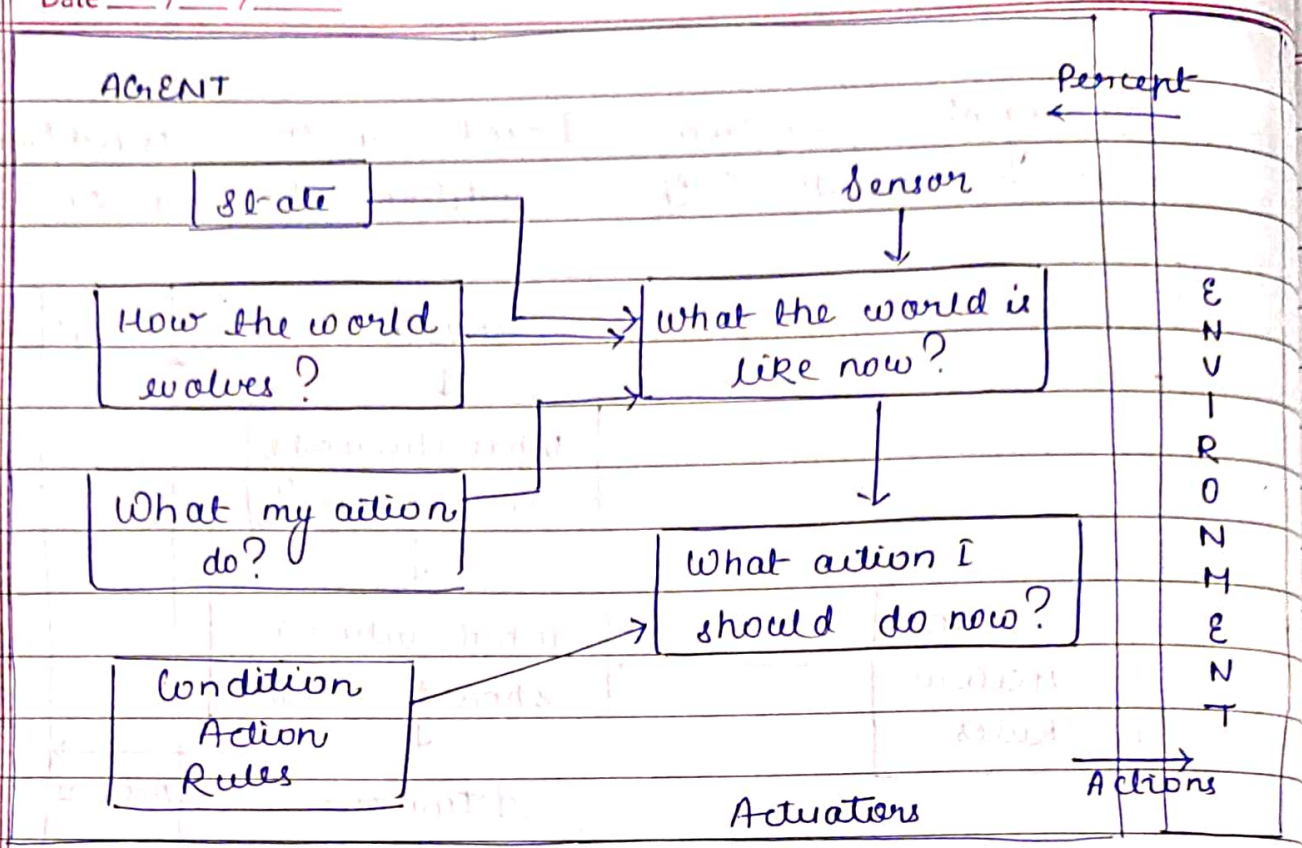
The agent-function based on the condition action rules: "If condition then action".



2. Model based reflex agent

A model based agent can handel partially observable environment. If current state stored inside the agent maintaining some kind of structure which describes the part of the world which can not be seen. This knowledge is about 'how the world works" is called the model of the world, hence the name "Model Based Agent".

Date ___ / ___ / ___



3/07/24

Properties of fuzzy set-

- Equality of 2 fuzzy sets.
  Fuzzy set A is considered ~~to be~~ equal to fuzzy set B if and only if

$$\mu_A(x) = \mu_B(x) \quad \forall x \in X$$

$A = \{ 0.3/1 , 0.5/2 , 1/3 \}$
$B = \{ 0.3/1 , 0.5/2 , 1/3 \}$

$A = B$

- Inclusion of one set into another fuzzy set.
  i.e fuzzy set $A \subseteq X$ is included in another

fuzzy set $B \subseteq X$:

$\mu_A(x) \leq \mu_B(x) \quad \forall x \in X$

eg. let $X = \{1, 2, 3\}$

$A = \{0.3/1, 0.5/2, 1/3\}$

$B = \{0.5/1, 0.55/2, 1/3\}$

$A \subseteq B$

- Cardinality of fuzzy set A is expressed as a sum of values of the membership function of 'A'

$$Card_A = \mu_A(x_1) + \mu_A(x_2) + \dots + \mu_A(x_n) = \Sigma \mu_A(x_i)$$

for $i = 1, 2 \dots n$

eg $A = \{0.3/1, 0.5/2, 1/3\}$

$card_A = (0.3 + 0.5 + 1) = 1.8$

- A fuzzy set A is empty if and only

$\mu_A(x) = 0 \quad \forall x \in X$

eg $A = \{0/1, 0/2, 0/3\}$

$\mu_A(x) = 0$

Date __/__/____

- alpha-cut { α-cut }

An alpha-cut of a fuzzy set $A \subseteq X$ is an ordinary set

$$A_\alpha \subseteq X$$

$$A\alpha = \{ \mu_A(x) \geq \alpha, \forall x \in X \}$$

eg $X = \{1, 2, 3\}$

$A = \{ 0.3/1, 0.5/2, 1/3 \}$

$\alpha = 0.5$

$A\alpha = A_{0.5} = \{2, 3\}$

- Operations of fuzzy set.

- Complement

If 'A' is the fuzzy set its complement can be found as follows.

$$\mu_{\neg A}(x) = 1 - \mu_A(x)$$

$A = \{ 0.3/1, 0.5/2, 1/3 \}$

$\bar{A} = \{ 0.7/1, 0.5/2, 0/3 \}$

- Intersection

Fuzzy intersection is a lower membership in both set of each element. The fuzzy intersection of 2 fuzzy sets A and B

Date __/__/____

on universal set X. is

$$\mu_{A \cap B} = \mu_{min}(\mu_A(x), \mu_B(x)) \quad \forall x \in X$$

$A = \{1/a, 0.3/b, 0.2/c, 0.8/d, 0/e\}$

$B = \{0.6/a, 0.9/b, 0.1/c, 0.3/d, 0.2/e\}$

$A \cap B = \{0.6/a, 0.3/b, 0.1/c, 0.3/d, 0/e\}$

- Union

In fuzzy set union is the reverse of intersection Union is the largest membership of the element in the either set.

The fuzzy operation for forming the union of 2 fuzzy set A & B on universal set ~~A and~~ x can be given as

$$\mu_{A \cup B} = \mu_{max}(\mu_A(x), \mu_B(x)) \quad \forall x \in X$$

$A = \{1/a, 0.3/b, 0.2/c, 0.8/d, 0/e\}$

$B = \{0.6/a, 0.9/b, 0.1/c, 0.3/d, 0.2/e\}$

$A \cup B = \{1/a, 0.9/b, 0.2/c, 0.8/d, 0.2/e\}$

6/4/24 Date __/__/____

Proposition

A proposition is a declarative sentence i.e either true or false but not both.

Logical Connectives

$\neg$	→	Negation
$\wedge$	→	conjunction
$\vee$	→	Disjunction
$\rightarrow$	→	Implication
$\leftrightarrow$	→	Bi-implication

Truth table of Implication

X	Y	$X \rightarrow Y$
F	F	T
F	T	T
T	F	F
T	T	T

Truth table of Bi-implication

X	Y	$X \leftrightarrow Y$
F	F	T
F	T	F
T	F	F
T	T	T

{used in FOPL}

Date __/__/____

Properties of statements

1. A statement is satisfiable if there is some interpretation for which it is true.
   eg $P \vee Q$

2. Contradiction - A statement is contradictory if there is no interpretation for which it is true.
   eg. $P \wedge \neg P$

P	$\neg P$	$P \wedge \neg P$
T	F	F
F	T	F

3. Tautology - A statement which is true for every interpretation.
   eg $P \vee \neg P$

P	$\neg P$	$P \vee \neg P$
T	F	T
F	T	T

4. Equivalence - Two sentences are equivalent if they have the same true value under every interpretation.

Date __/__/____

$P \wedge Q \equiv \neg(P \vee Q)$

P	Q	$P \wedge Q$	$P \vee Q$	$\neg(P \vee Q)$
T	T	T	T	
T	F	F	T	
F	T	F	T	
F	F	F	F	

eg.

$P \equiv \neg(\neg P)$

$p \rightarrow q \equiv \neg P \vee q$

FOPL - { First Order Predicate Logic }

quantifiers $\exists_x$, $\forall_x$

Pilot (father-of (bill)) { Predicate logic

Unification

In order to determine contradiction in predicate logic we need a ~~matching~~ magic procedure that compare two literals and discover whether there exists a set of substitution that makes them identical Where is a recursive processor we use to do this. This process is called unification.

To attempt to unify two literals we first check if their initial predicate symbols are same if they are we can proceed, otherwise

Date __/__/____

They can't be unified regardless of their argument.

eg. likes (Harry, milk) } can be unified
likes (Harry, cricket) }

likes (Harry, milk) } can't be unified
plays (Harry, cricket) } as predicate literals

Any substitution that makes 2 or more expression equal is called unifier for the expression and the process is unification the matching rules are

a) a different predicates or constants can't match only identical one

b) a variable can match another variable any constant or a predicate expression with restriction that the predicate exp. must not contain any instance of the variable being matched.

eg we want to unify following expression p(x, x) p(y, z)

Compare 'x' & 'y' and decide that if we substitute y for x (y/x) they should match

p(y, y) p(y, z)

Now, compare 'y' & 'z' and substitute (y/z)

p(y, y) p(y, y)

Date __/__/____

The entire unification process is succeded with a substitution that is a composition of two substitution we found
we write the composition as
$(y/x)\ (y/z)$

5/07/24 ASSIGNMENT.

Q. WAP to print square, cube of given numbers.
eg number: 1 square : 1 cube : 1
number: 2 square: 4 cube: 8

Q WAP to find power of given number

Q WAP to generate patterns

```
*
* *
* * *
* * * *
```

```
0
0 1
0 1 2
0 1 2 3
0 1 2 3 4
```

```
* * * *
* * * *
* * * *
* * * *
```

```
0 0 0 0 0
0 1 2 3 4
0 2 4 6 8
0 3 6 9 12
0 4 8 12 16
```

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```