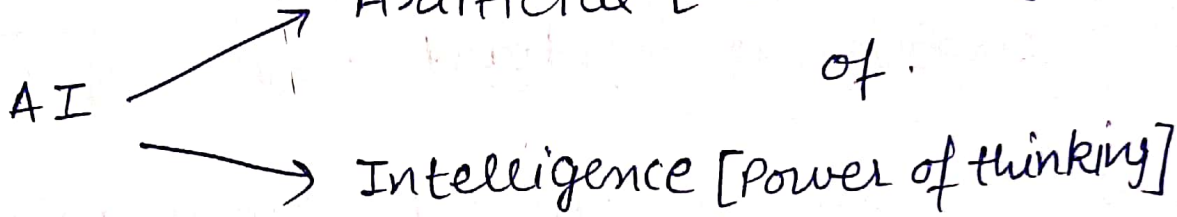
Dr. R K Chaudhary

Artificial Intelligence -

AI is the study of How to make computer do things which people do Better [machine + Human Intelligence]

↳ AI can cause a machine to work as Human



Reasons of Boost in AI -

(i) Software device can be made to Solve Real time problems.

(ii) Creation of virtual assistant [SIRI, CORTANA]

(iii) Robots development [Helps in dangerous environment condition]

(iv) New JOB opportunities.

Goals of AI -

(i) Replication of Human Intelligence.

(ii) Solving problems that recquire Knowledge.

(iii) Building a machine can do human Intelligence task [CHESS, Proving theorem, automated car driving]

Scanned with OKEN Scanner

(iv) Provide advice to the user.

(v) Intelligent connection b/w perception & communication.

Applications

(1) Gaming - chess, Poker, tic-tac-toe.
↳ machine can think large no. of moves.

(2) NLP - Natural Language processing.
↳ Machine can understand human language.

(3) Healthcare - Fast diagonesis
↳ Robotic surgery.

(4) AI in finance - Adaptive Intelligence.
↳ automatic chatbots, algo. trading.

(5) Data security - Helps in making data application more secure
↳ AEG Bot

(6) Expert system - Integration of Software machine & Special Info to provide Reasoning & advice.

Scanned with OKEN Scanner

Dr. R K Chaudhary

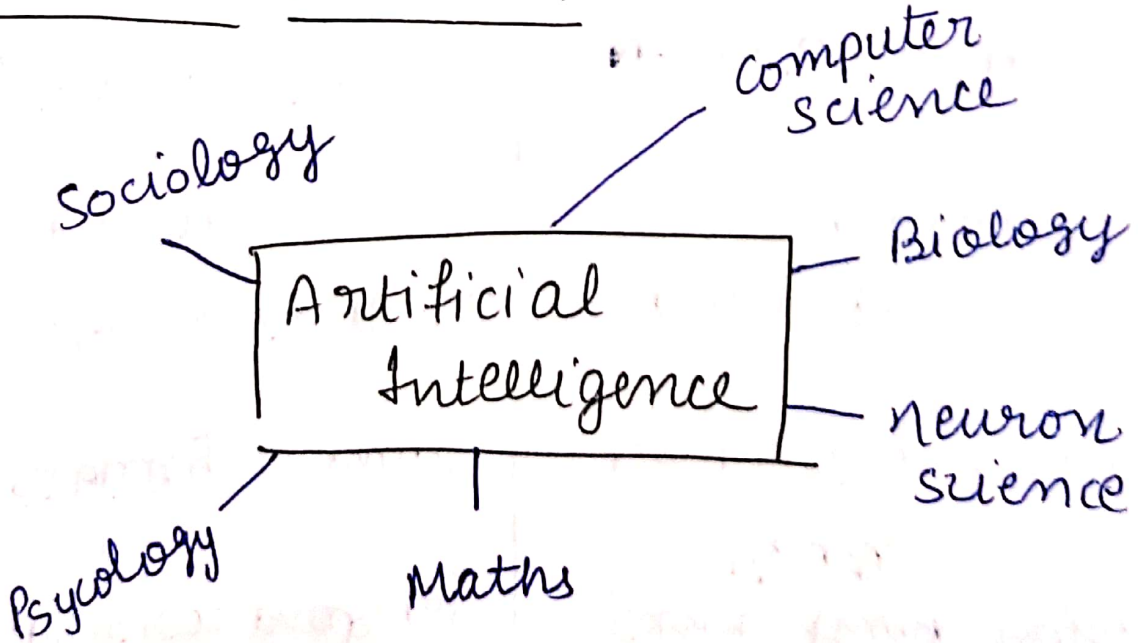
(7) computer vision - understand the visual automatic by machine (Image Processing)

(8) Speech Recognition - Extracting the meaning of sentence by human talk. [Slang Removal, noise Removal]

(9) Robotics - Talk & Behave like humans.
↳ Erica & Sofia.

(10) AI in e-commerce - Automatic Recomendation of Product Service Request

AI is comprised of -



↳ Reasoning
↳ Learning
↳ Problem Solving

↳ Language understanding

Scanned with OKEN Scanner

Advantages	Disadvantages
Accuracy ↑, Error ↓	costing ↑
Fast decision making	can't think beyond limits.
Reliability is more	No feeling & Emotions
Usefulness in Risky Areas.	more dependency on machine.
Digital Assistant	No original thinking

BFS - Breadth First Search -

↳ Explores all the nodes at given depth before proceeding to the next level.

↳ Use Queue to implement.

Algorithm -

(1) Enter starting node on Queue

(2) If Queue is Empty, then return fail and stop.

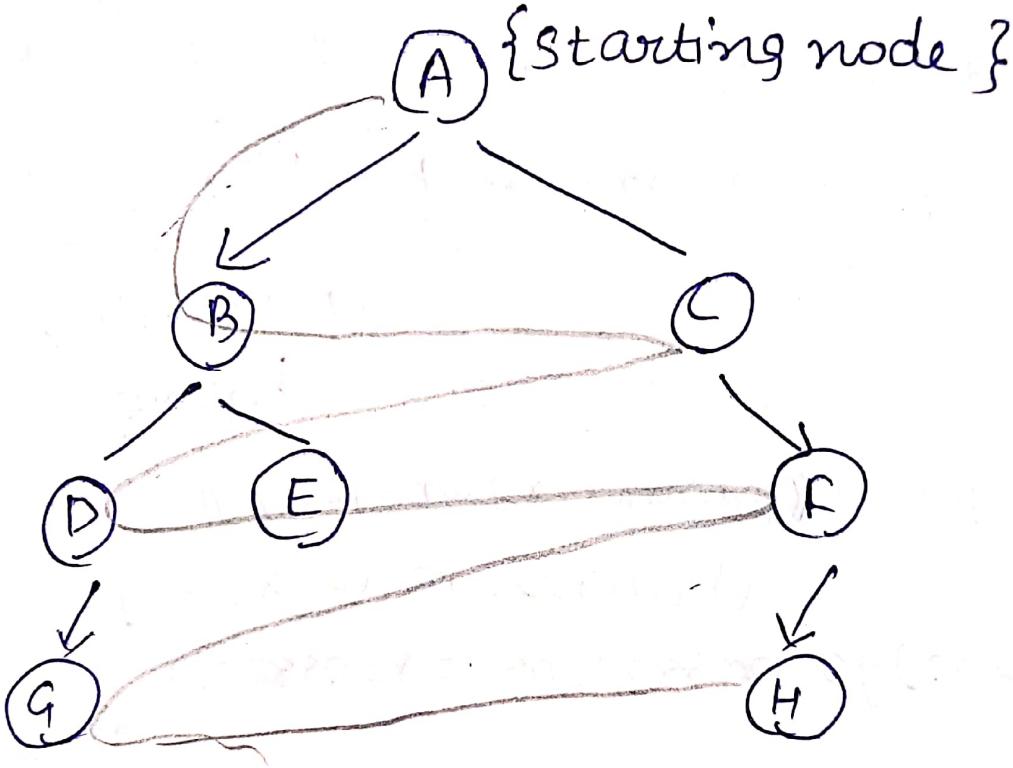
(3) If First Element on queue is Goal node then return success and stop.

Scanned with OKEN Scanner

Else

(4) Remove and Expand First Element from Queue and place Children at end of Queue.

(5) Go to step (ii)



Initial Queue {A} = Goal node ×

↳ Remove from Queue & add its successor to Queue

{B, C}

↓ ↳ (D, E)

{C, D, E}

{D, E, F}

{E, F, G}

↓

{G, H}

Scanned with OKEN Scanner

Pseudocode of BFS

```
List open, closed, successors = { };
Node root_node, current_node;
 insert_last (root_node, open)

while not_enpty (open);
    current_node = remove-first (open);
    insert-last (current_node, closed);
    if (goal (current_node)) return currentnode;
    Else
       successors = successors of (current_node;
       for (x in successors)
          if (not-in (x, closeed)) insert-last (x, open)
                                                ;
       Endif
    Endwhile.
```

Advantage - find a solution if exists.

→ Try to find minimal solution in least no. of steps.

Disadvantages -

→ More memory.
→ needs lot of time.
→ solution is far from node root

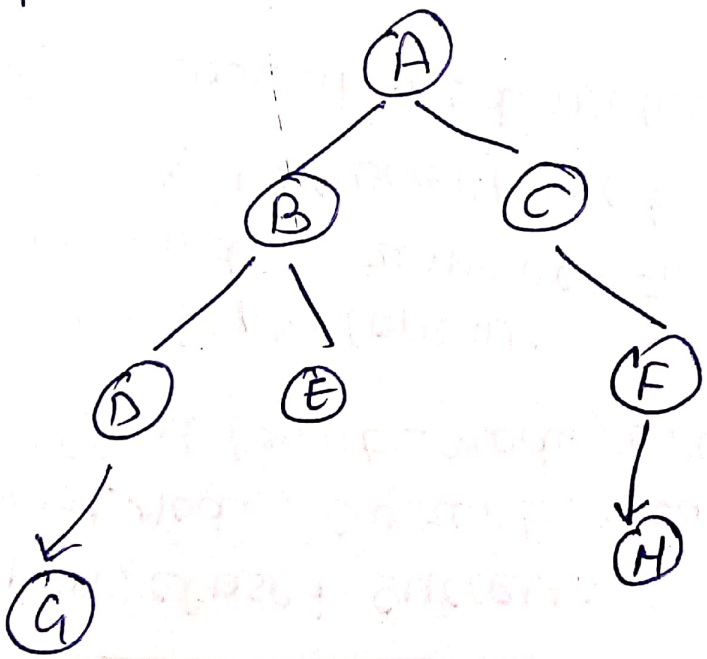
Scanned with OKEN Scanner

Dr. R K Chaudhary

Time complexity = $O(b^d)$ — depth.
↳ Branching Factor

DFS – Depth First Search

→ Recursive algorithm.
→ starts from root node and follows each path to its greatest depth node before moving to next path.
→ Implemented using STACK (LIFO)



Algorithim – (PUSH)

(i) Enter root node on stack.
(ii) Do until stack is not empty. → POP
(iii) ↳ (a) Remove node
↳ (i) if Node = Goal Stop.
(ii) Push all children of node in stack

Scanned with OKEN Scanner

Advantage - Less memory

↳ Less time to Reach goal node if transversal is Right path.

Disadvantages

↳ No guarantee of finding Solution

Pseudocode

```
List open, closed, successors = {};
Node root_node, current_node;
insert-first (root_node, open)

while not-empty (open);
    current_node = remove-first (open);
    insert-first (current_node, closed);
    if (goal (current_node)) return current_node;
    Else
        successors = Successors Of (current_node);
        for (x in successors)
            if (not-in (x, closed)) insert-first (x-open)
    End if
End while.
```

Time complexity = $O(b^d)$

Space = $O(bd)$

Scanned with OKEN Scanner

Dr. R K Chaudhary

Heuristic Search

Tries to solve a problem in minimum steps / cost

↳ tries to optimize a problem using heuristic function (Informed search)

H. function - gives an estimation of the cost of getting from node "n" to Goal state.

{Helps in selecting optimal node for expansion}

Types

Admissible (understimate)

$H(n) \leq H'(n)$ {Goal}

$H(B) = 3$

$F(n) = $ ~~[illegible]~~

$= H(n) + G(n)$

B to G (by B, C, F)

Actual = 11

$3 < 11$

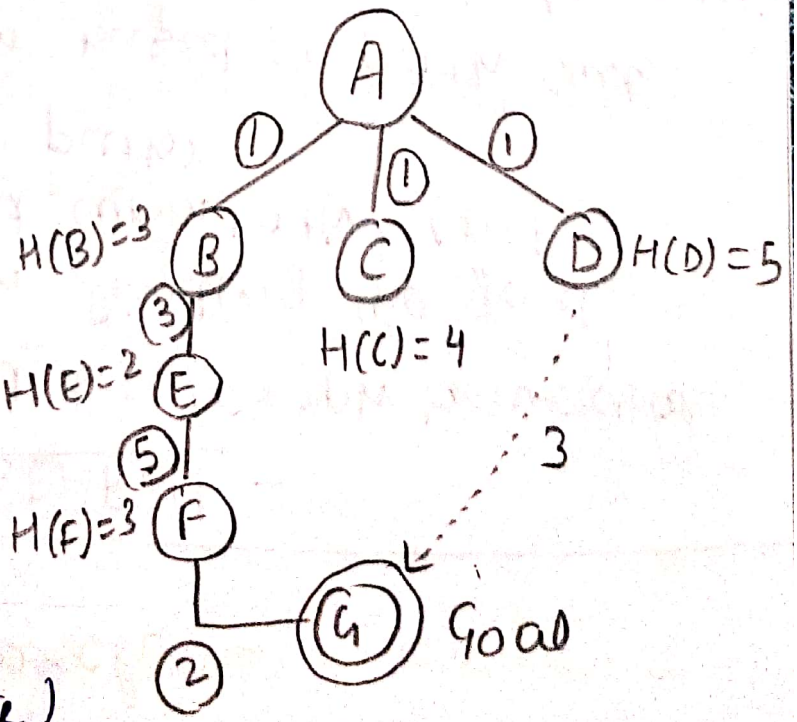
(Admissible)

A → G by D

$5 \not\geq 4$ (overestimate)

Non-admissible (overestimate)

$H(n) > H'(n)$



Scanned with OKEN Scanner

Best first Search -

UNIFORM COST SEARCH -

↳ used for weight tree / Graph traversal.

↳ Goal is to path finding the goal node with lowest cummlative cost (optimal path)

↳ Node Expension Based on path costs

↳ priority Queue is used for implementation

↳ {High priority to minimum cost}

↳ Backtracking.

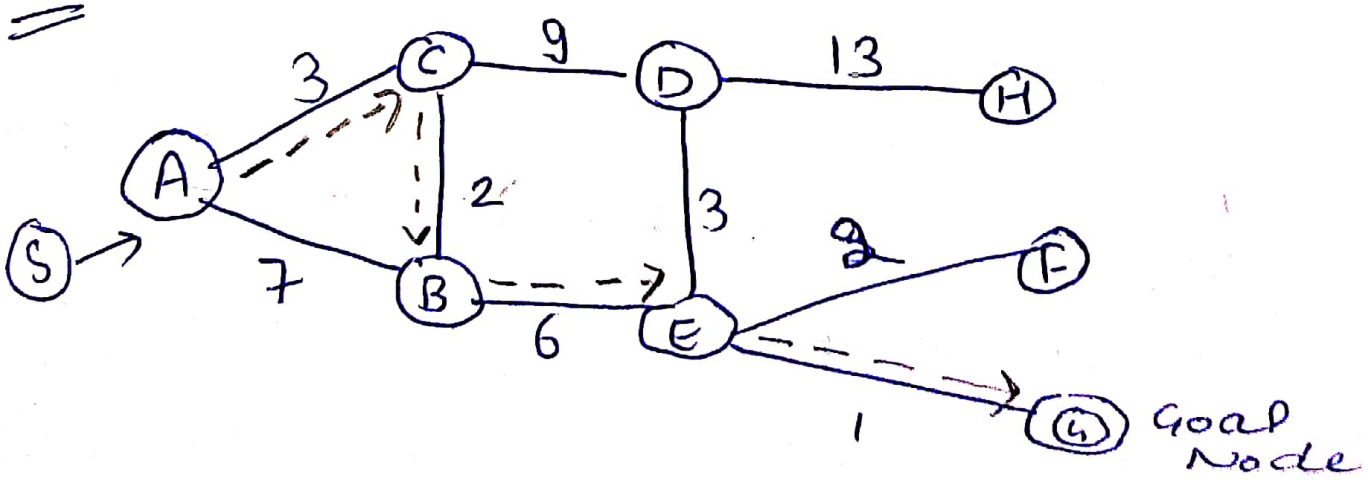
Advantages

↳ optimal solution

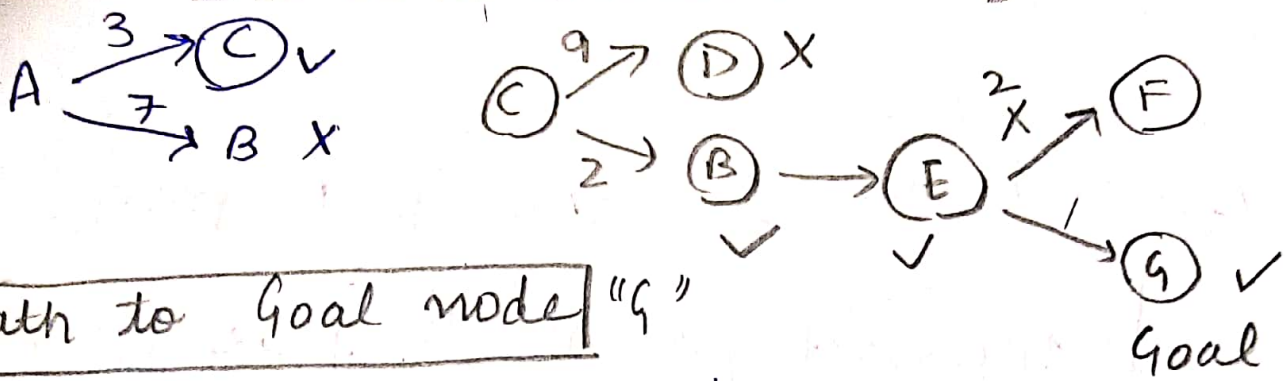
Disadvantages

↳ Stuck in infinite loop.

Ex



Scanned with OKEN Scanner



Path to Goal node "G"

$$A \xrightarrow{3} C \xrightarrow{2} B \xrightarrow{6} E \xrightarrow{1} G$$

12

Path to Goal node - "H"

$$A \to C \to B \to E \to G \xrightarrow{1} E \xrightarrow{2} F \xrightarrow{3} E \xrightarrow{3} D \xrightarrow{13} H$$

12 + 21 = 33

INFORMED / Heuristic search.

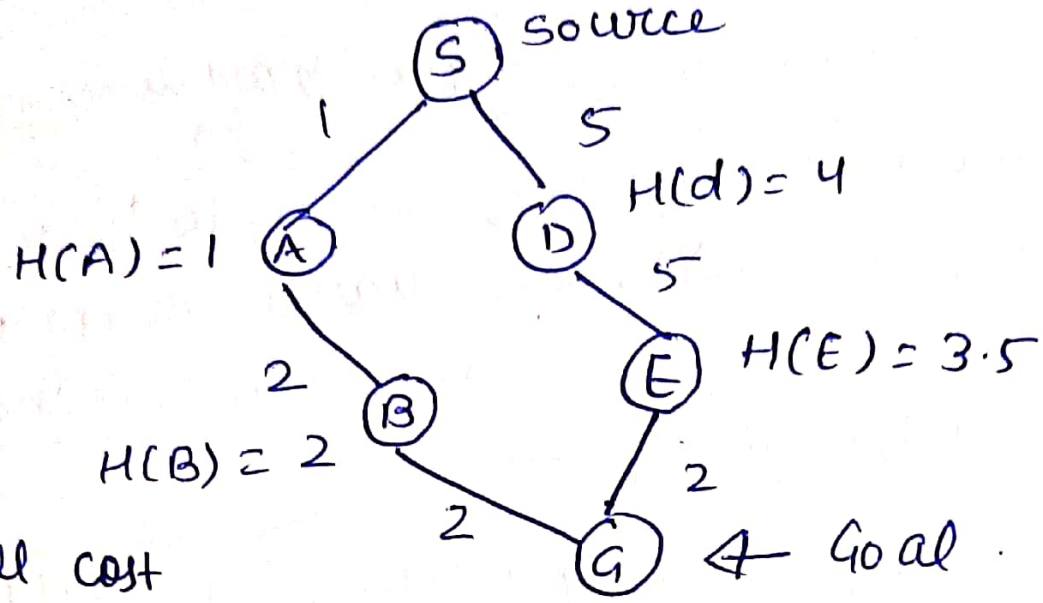
Heuristic function

↳ Information about Goal state is present.
↳ Better than uninformed search.
↳ find optimal solution to Reach Goal state
↙ minimum path cost → using Heuristic ~~Search~~ function.

It is a search which tries to reduce amount of search that must be done by making intelligent choices for the nodes that are selected for Expansion.

Scanned with OKEN Scanner

Example (1) Best fit Search algo
(2) A* search algo



overall cost

$F(n) = \underbrace{G(n)}_{\text{Path cost}} + H(n)$ → Heuristic value {For every node}

S → A (1) { $F(n) = 1+1 = 2$ } $2 < 9$

S → D (5) { $F(n) = 5+4 = 9$ }

A → B [$F(n) = 3+2 = 5$ }

Back tracking is not possible

Total path cost = { S → A → B → G]
= 5

Scanned with OKEN Scanner

Dr. R K Chaudhary

Best First Search - (GREEDY SEARCH)

↳ uses Evaluation algo (Function) to decide which adjacent node is most promising and then explore.

↳ category of Heuristic Search & Informed search.
{ combination of Heuristic function }

↳ priority Queue is used to store cost of node

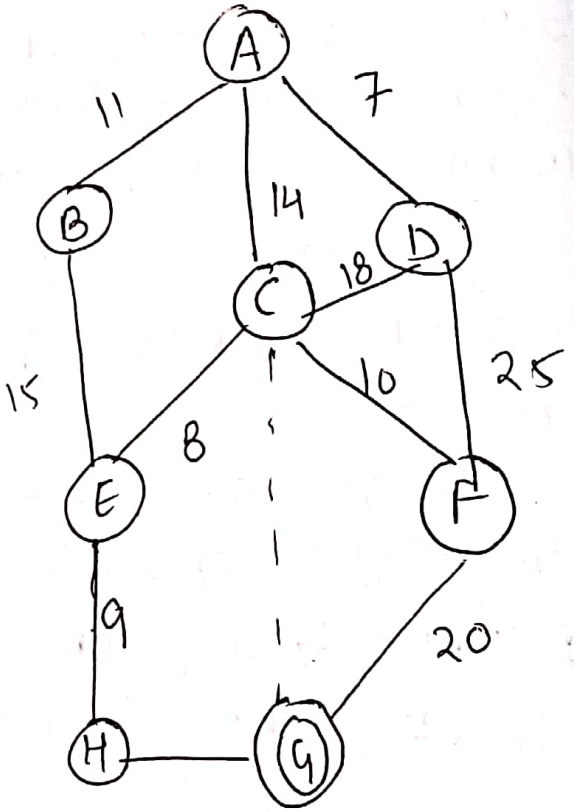
Algorithm :-

1. place starting node into the OPEN list.
2. If the OPEN list is emty Stop & Return failure
3. Remove the node n, from the OPEN list which has the lowest value of h(n), and places it in the closed list.
4. Expand the node n, and genrate the successors of node n.
5. check each successor of node n, and find wheather any node is goal node or not. If any successor node is goal node, then return success and terminate search, else proceed to step 6.
6. For each Successor node, algo checks for evaluation function f(n) & then check if the node has been in either open or closed

Scanned with OKEN Scanner

list. If the node has not been in both list then add it to open list.

(7) Return to step 7.



straight line distance

A → G = 40
B → G = 32
C → G = 25
D → G = 35
E → G = 19
F → G = 17
G → G = 0
H → G = 10

open	closed
[A]	[ ]
[C, B, D]	[A]
[F, E, B, D]	[A, C]
[G E B D]	[A, E, F]
[E, B, D]	[A, C, F, G]

Scanned with OKEN Scanner

Dr. R K Chaudhary

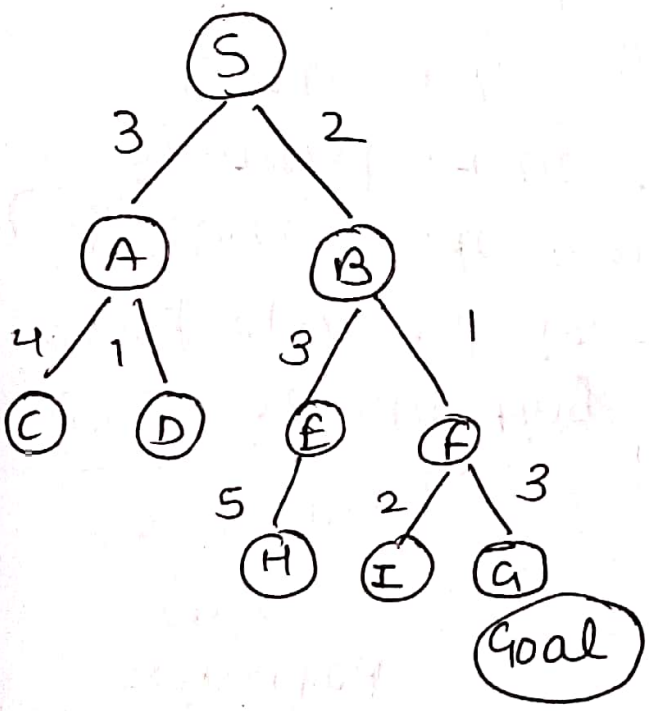
Path = [A → C → F → G] = 44

Complexity

Space = $O(b^d)$ Time = $O(b^d)$

b → branching factor d → depth.

Example 2



node (n)	H(n)
A	12
B	4
C	7
D	3
E	8
F	2
H	4
I	9
S	13
G	0

open	closed
[B, A]	[S]
[F, E, A]	[S, B]
[G, I, E, A]	[S, B, F]
[I, E, A]	[S, B, F, G]

Path.

A* Algorithm for Search -

↳ uses heuristic function (h(n)) and cost to reach the node "n" from start state g(n)

↳ Finds shortest path through search space

↳ Fast and optimal result.

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

- $h(n)$ → heuristic value (child node)
- $g(n)$ → cost to Reach node
- $f(n)$ → Estimated cost

Algorithm -

① Enter starting node in OPEN list.

② If open list is empty Return fail.

③ Select node from open list which has smallest value of (g+h)
↳ if node = goal, return success

④ Expand node 'n' generate all successors
↳ compute (g+h) for each successors node

⑤ if node "n" is already in open/closed attach to back pointer.

⑥ Go to (ii).

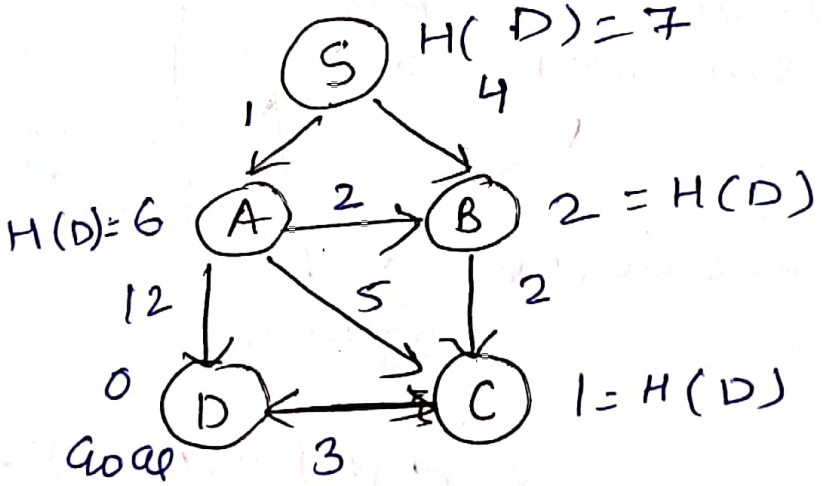
Dr. R K Chaudhary

Advantages

→ Best search algo.
→ optimal & complete.
→ solve complex problem

Disadvantages

→ Doesn't always produce shortest path.
→ complexity issues.
→ Requires memory.



S → A
S → B

S → A = 1 + 6 = 7

S → B = 4 + 2 = 6 ✓

S → B → C = 4 + 2 + 1 = 7

S → B → C → D = 4 + 2 + 3 + 0 = 9

S → A → B = 1 + 2 + 2 = 5

S → A → C = 1 + 5 + 1 = 7

S → A → D = 1 + 12 = 13

solution ↑

S → A → B → C = 1 + 2 + 2 + 1 = 6

S → A → B → C → D = 1 + 2 + 2 + 3 + 0 = 8

S → A → C → D = 1 + 5 + 3 + 0 = 9

Scanned with OKEN Scanner

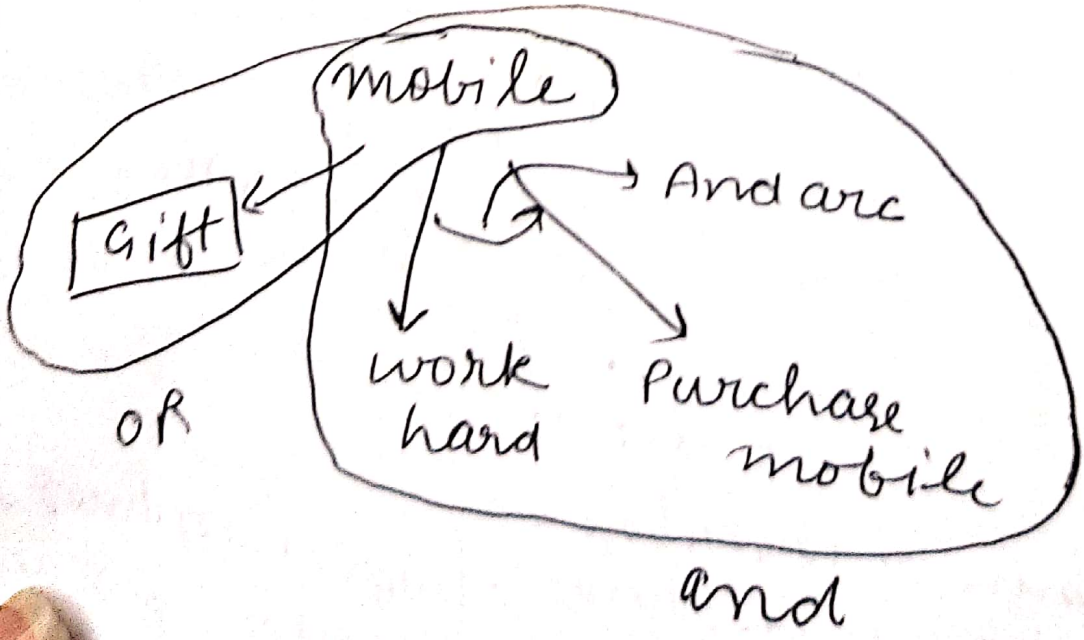
AO* algorithm -

AND OR Graphs

useful for Representing the problem Solution that can be solved by decomposing them into smaller set of problems, all of which must be Solved.

→ ANDARCS → May point to "n" no. of successors node

→ Algo like Best fit search can be used that has the ability to handle ANDARCS.



Dr. R K Chaudhary

Futility - should be chosen to correspond to threshold value if Est cos > Futility stop search.

Algorithm

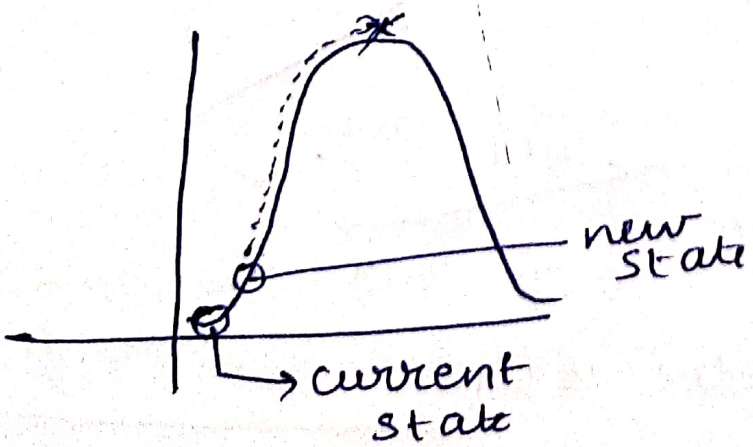
HILL CLIMBING

- local search algo
- Greedy approach
- NO Back tracking

↳ variant of generate and test method in which feedback from test procedure is used to help generator decide which direction to move in search space

↳ always moves in single direction.

↳ It is like DFS.



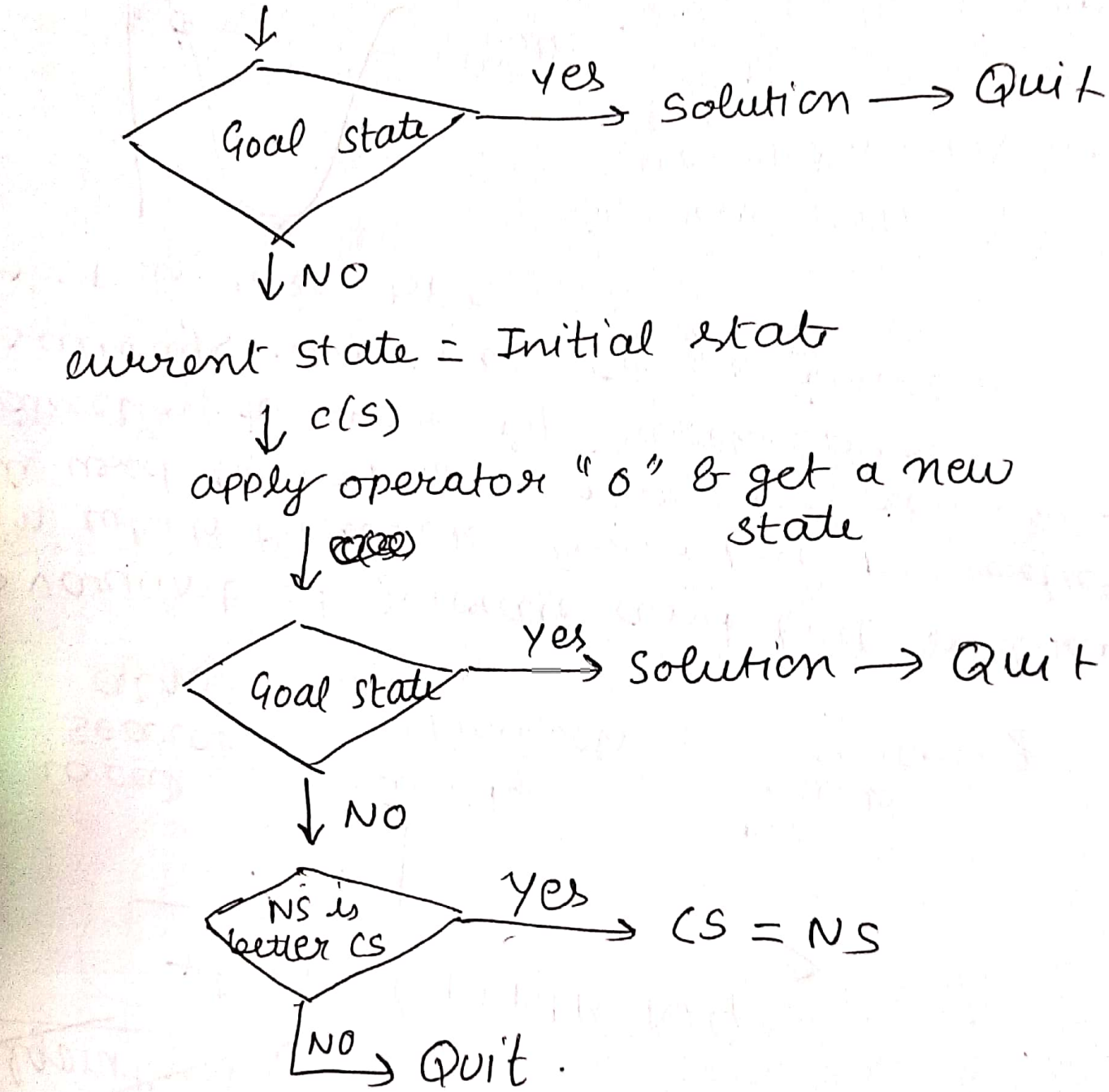
if new state is better than current state

↓

new state = current state

Scanned with OKEN Scanner

Flowchart

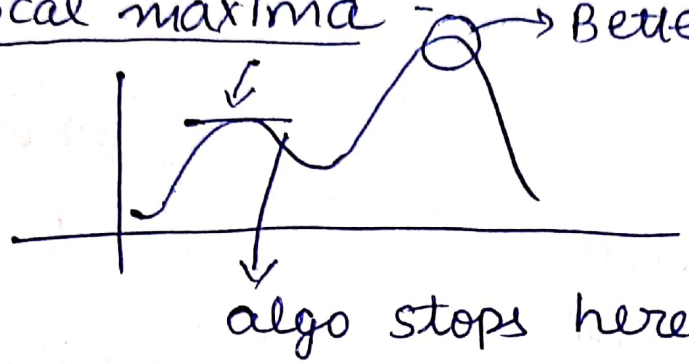


Scanned with OKEN Scanner

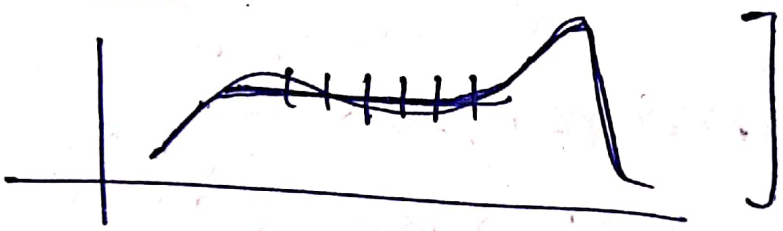
Dr. R K Chaudhary

Limitations

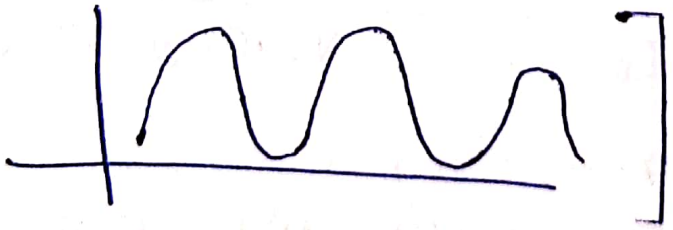
① Local maxima



② plateau :



③ Ridge



Turning test :

↳ Alan turing in 1950

↳ used to determine "whether or not machines can think intelligently like humans"

Scanned with OKEN Scanner

Basic configuration

↳ There will be a human Interrogator (examiner) on one side of wall and other side a machine and human.

↳ machine Intelligent → Knowledge Base

when human Interrogator can't distinguish Response given by machine and human.

⇒ machine has passed the test and it is intelligent.

PEAS GROUPING of AI agents

↳ used to group similar type of agents together.

↳ PEAS
- Sensors (Devices from which agent percives observation from Env)
- Actuators (H/w, S/w through which agent performs)
- Environment (surrounding things & conditions)
- Performance measure
  - o/p we get from agent
  - ↳ Results obtained after Agent processing

Example - self driving cars

[P] - comfort, safety, time, legal driving.

[E] - condition of Roads, Signals, crossing.

[A] - Steering, Breaks, horn accelerator

[S] - Cameras, GPS, speedometer

State space search - used in problem solving

↳ process used in AI which successive configuration of states of an instance are considerd with intention of finding a GOAL State with desired property.

↳ Problems are modelled as state space
Set of states in which a problem can be

↳ Repersentation:

S: (s, A, Action(s), Result(s,a), cost(s,a))

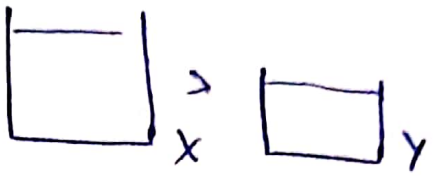
- S: Set of all possible states
- Action(s): which action is possible for current state
- Result(s,a): State Reached by Performing action (a) on state (s)
- cost(s,a): costing

Scanned with OKEN Scanner

Dr. R K Chaudhary

Water JUG Problem and its State Representation

Two Jugs of different capacities are given



x-litre y-litre

* No Marking is there on the jug.

↳ Goal is to fill Exactly "L" Liters of water into y-litre Jug.

State is Represented as $\langle x, y \rangle$

- x: integer amount of water in x-litre Jug
- y: integer amount of water in y-litre Jug.

Example Suppose capacity of two Jugs.

2 l 3 l.

Goal To get Exactly '1' litre of water in 2L of Jug.

$\langle 0,0 \rangle$ {initial state

$\langle 0,3 \rangle$

$\langle 2,1 \rangle$

$\langle 0,1 \rangle$

$\langle 1,0 \rangle$

Scanned with OKEN Scanner

CRYPTARITHMATIC PROBLEM

→ Type of constraint satisfaction problem.

constrains
- → No two letters have same value
- → sum of digits must be as shown in problem
- → There should be only one carry forward.

* Digits that can be assigned to a word / alphabet (0-9) (Range)

```
  T O
  G O
-------
  O U T
```

T	2
O	1
G	8
U	0

O → 1

2 1

8 1

---

1 0 2

Dr. R K Chaudhary

		C_2	C_1	
1	9	5	6	
	1	0	8	5
1	0	6	5	

SEND
MORE
MONEY

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

$E + O = N$

$E + O = N$

if $C_2 = 0$

$E = N$

i.e., X.

So, $\hookrightarrow C_2 = 1$

let $E = 5$

$E + O + C_2 = N$

$N = 6$

$C_1 + 6 + R = 5$

$1 + 6 + R = 5$

$7 + R = 5$

$D + E = Y$

$D + 5 = Y$

$7 + 5 = 12$

Scanned with OKEN Scanner

6

```
  EAT
  THAT
-------
 APPLE
```

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

	8	1	9	
9	2	1	9	
1	0	0	3	8

$C_1 + T = P$

$9 + 1 = 10$

$E + H = 10$

$8 + H = 10$

```
  SOME
  TIME
-------
 SPENT
```

S	1
O	9
M	3
E	4
T	8
I	5
P	0
N	6

	1			
1	9	3	4	
8	5	3	4	
1	0	4	6	8

$C_1 + S + T = P$

$1 + 1 + 8 = 0$

$0 + 1 = E$

$= 14$

Scanned with OKEN Scanner

Dr. R K Chaudhary

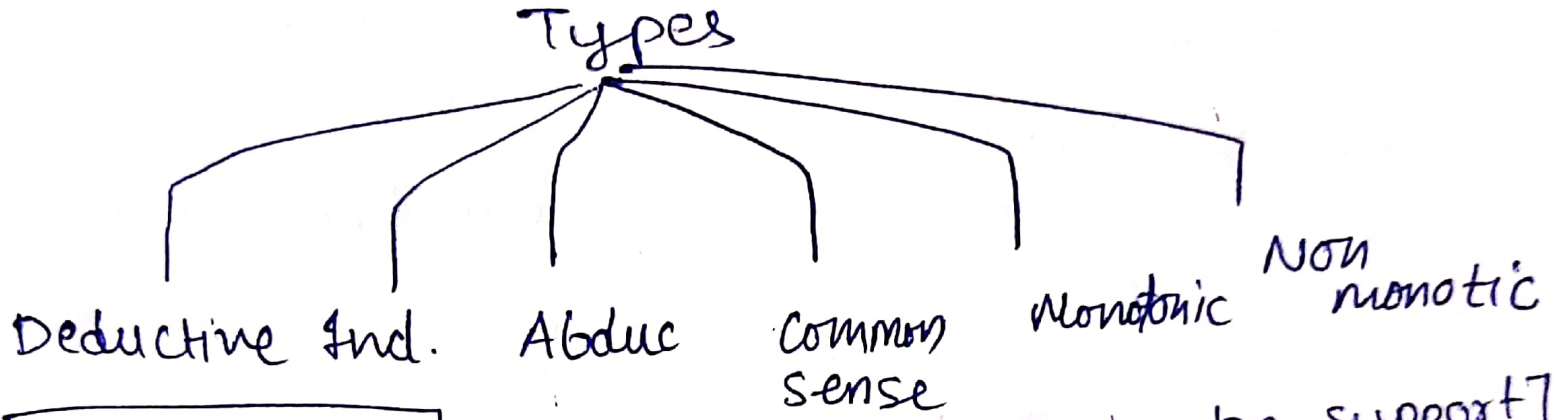
unit - 3

Reasoning

↳ conclusion are drawn Based on what is most likely to be true.

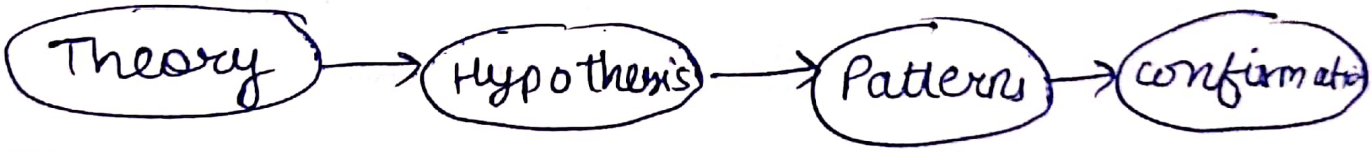
↳ a process jisme new knowledge generate karte hai Based on previous knowledge

Types



Deductive [jo statement conclusion ke support mein bolte hai]

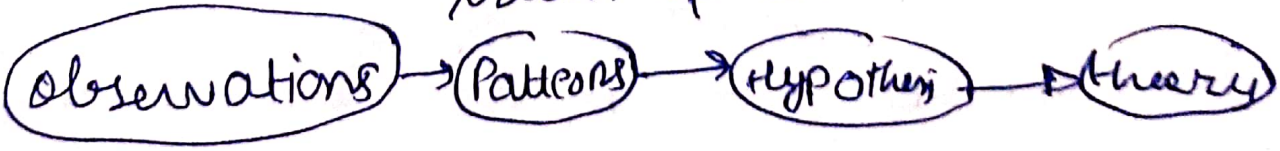
truth of premises → truth of conclusion



Inductive: cause effective Reasoning

Premises provide probable support to conclusion.

Premises truth doesn't guarantee truth of conclusion



Scanned with OKEN Scanner

Abductive -

starts wing single or Multiple observation seek to find most likely Explanation or conclusion for observation.

↳ Extension of deductive Reasoning but premises do not guarantee the conclusion -

Common sensee (informal form)

gained through expisences.

- Stimulates human abilility to make preasumptions about events occur everyday
- It Relies good judgement & Rather than Exact Knowledge

Monotonic Reasoning -

jab conclusion Nikal jata hai to vo hamesha same hi rehta hai no matter Kitni new information add ho rhi hai.

Scanned with OKEN Scanner

Non monotonic -
conclusion invalid ho jate hai kabhi kabhi after adding new information
↳ It deals with incomplete & uncertain models.

UNCERTAINITY

→ lack of Exact info or Knowledge that helps us to find correct conclusion

Sources

① uncertain Inputs
↳ Missing data ↳ Noisy data

② uncertain knowledge.
↳ Multiple causes lead to Multiple Effects
↳ Incomplete knowledge of causality in domain

③ uncertain outputs
↳ Abduction, induction are uncertain
↳ Default Reasoning.
↳ Incomplete decision influence

Scanned with OKEN Scanner

uncertainity may be caused by problems -

↳ ① missing/unavilable data
② unreliable / ambigous data
③ Imprecise / Inconsistent Representation of data.
④ Guess based ⑤ Default based

Baye's theorem

↳ Based on conditional probability
↳ Based on Prior Knowledge of conditions that might be Related to Event.

* calculate P(B/A) with knowledge of P(A/B)

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

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

$$P(A/B) P(B) = P(B/A) P(A)$$

$$\boxed{P(B/A) = \frac{P(A/B) P(B)}{P(A)}}$$

Scanned with OKEN Scanner

Dr. R K Chaudhary

Applications

1. Robot/automatic machine - next step is calculated based on prev step -
2. forcasting - weather prediction.
3. Solve Monty hall problem.

Bayesian Belief N/w in AI -

Defines probabilistic independencies & dependencies among the variables in N/w

↳ It is a probabilistic graphical model represents a set of variables & their conditional dependencies using a directed acyclic Graph.

↳ Built from probability distribution.

↳ Consists of → DAG
→ table of conditional probabilities

↳ Node : corresponds to Random variable → Continous
→ Discreate

↳ Arc/Directed arrows : Rep. casual Realtionship or conditional probabilities among Random variable.

Scanned with OKEN Scanner

Fuzzy logic system Architecture

Fuzzy set theory

↳ Mathematical tool for dealing with the concept used in natural language

Fuzzy logic: allows intermediate logic values to be defined b/w conditional Equations. [Resembles human Reasoning Involves possibilities b/w YES, NO]

→ Mathematical tool to Represent uncertainity and vagueness.

Characteristics -

(i) Exact Reasoning is viewed as limited case of approximate Reasoning.

(ii) Everything is a matter of degree

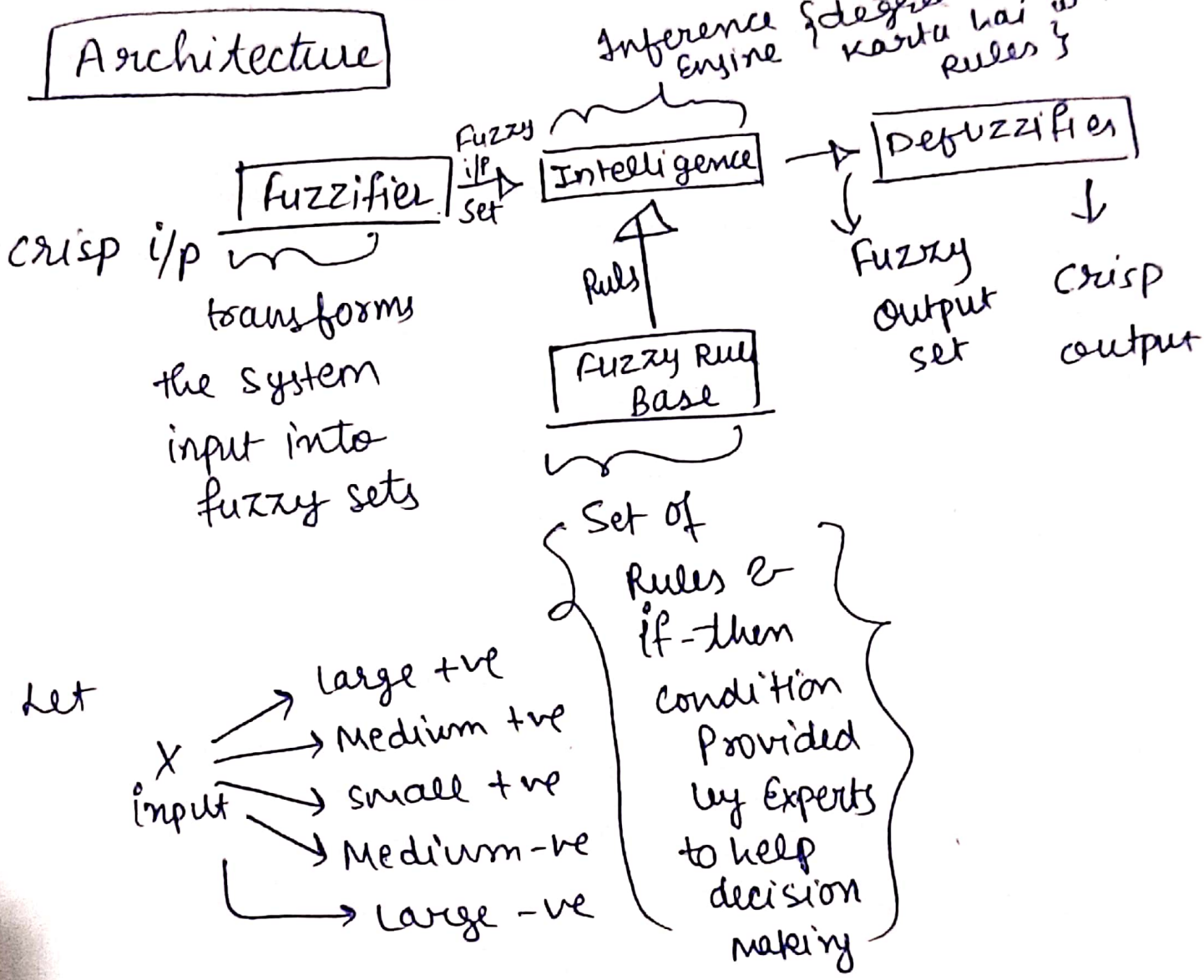
(iii) Knowledge is interpreted as coll^{n} of variables.

(iv) Any logical system can be fuzzified.

Scanned with OKEN Scanner

Dr. R K Chaudhary

Architecture



Scanned with OKEN Scanner

*** Dempster - Shafer theory

* also known as theory of belief functions
  → mathematical framework for Reasoning under uncertainty & combining Evidence
  ↳ alternative to traditional prob theory
  ↳ handle uncertain or incomplete info more flexibly.

Basic Belief assignment - isme yeh hota hai ki jaise koi event occur ho raha hai to yeh zarroori nhi hai ki uska answer yes or No mein hi dia jae, Balki possibility bhi dikhai ja skti hai like for Exp - I m not sure about that but it might happened.

Combination of Belief functions

The dempster's rule of combination is used to ~~test~~ merge belief functions

Dempster's Rule – mathematical formula use in dampster's Shafer theory to combine Evidence or belief functions.

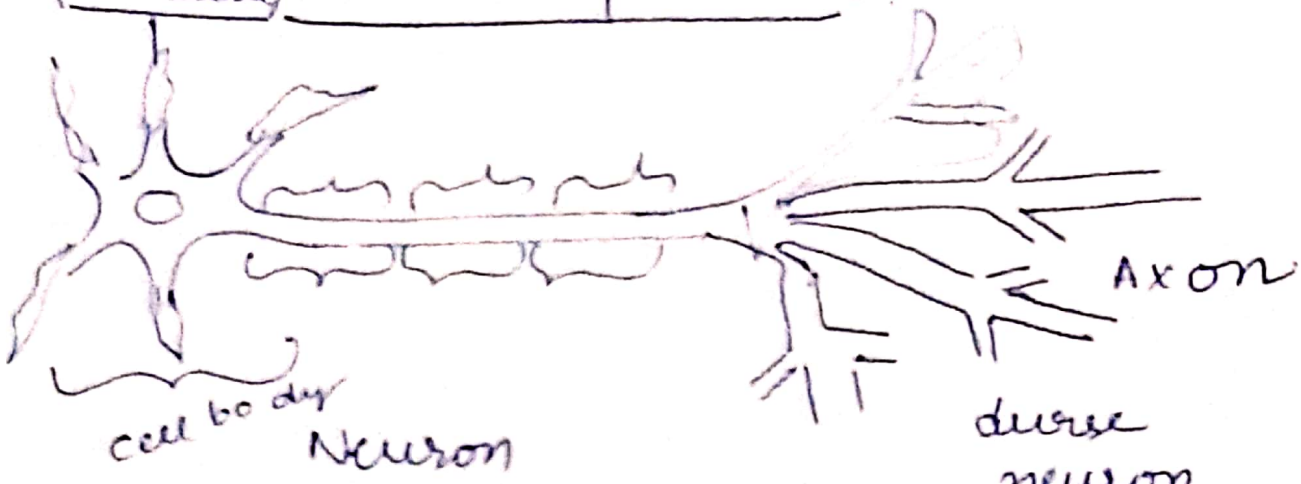
$$m_n(z) = \frac{\Sigma_{x \cap y = z} m_{n-2}(x) m_{n-1}(y)}{1 - \Sigma_{x \cap y = \phi} m_{n-2}(x) m_{n-1}(y)}$$

Applications –

- Decision making under uncertainity
- information fusion in sensor networks
- fault diagnosis

Scanned with OKEN Scanner

Dendride Neural network (Biological)



dusre neuron ke cell body attached hota hai

Dendride

→ Msg Recieve karta hai

→ generate karta hai action potential

cell body

→ information recieve karke integrate karta hai

Axon -

→ action perform karta hai msg Recieve hone ke baad

Dr. R K Chaudhary

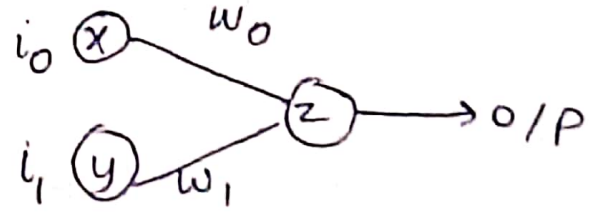
(Artificial neuron network)

Neuron - Basic unit hota hai jo mimic karta hai Biological neurons ko

Types of neural Network

- FNN {feed forward} Neural network } (one direction mein flow karti hai information)
- CNN {Convolutional neural network (image processing & pattern Recognition)}
- RNN { Recurrent Neural network } {sequence data }

Characteristics -



(1) Neurally implemented mathematically model.

(2) Bohut sare interconnected processing Elements hote hai.

Input = $i_0 w_0 + i_1 w_1$ (weighted sum)

Advantages

(1) Ability to learn non linear & complex Relationship

(2) Easy Generalization

(3) No Restrictions on Input

Scanned with OKEN Scanner

Dr. R K Chaudhary

Unit - IV

Game playing & current trends in AI -

↳ fundamental area in AI research for Decades.

current trends -

- **Reinforcement learning** -
  - → trial & Error ke through play kia jata hai
  - → learning from actions & consequences.
- **Generative adversarial Networks (GAN's)** -
  - → used to Genrate content with in Games. { creating Realistic textures, characters & Environment }
- **procedural content Generation (PCG)** -
  - → used to genrate content dynamically levels, maps & quests jo ki help karte hai player ko replayability Enhance karne mein.
- **Player Behavior prediction** -
  - used to predict player behavior
- **Real time strategy Games** -
  - Games Required long term planning, resource management & quick decision making.

Scanned with OKEN Scanner

AI ka role continously evolving, development, optimization.

MIN - MAX Procedure -

- Recursive or Backtracking algorithm used to make decision & game theory
- Provide karti hai optimal move assume karke opponent player bhi optimally khel raha hai.
- used in games like chess, tic-tac-toe, (two players game)
- It computes the min-max decision of current state
- two players hote hai EK Max EK min dono fight karte hai opponent player - get minimum benefit while they get Maximum.
- Max will select Maximized value & Min will select Minimized value
- depth first search use hoti hai Exploration ke lie game tree mein.
- terminal node tk jane ke Baad Backtrack karti hai as Recursion.

Scanned with OKEN Scanner

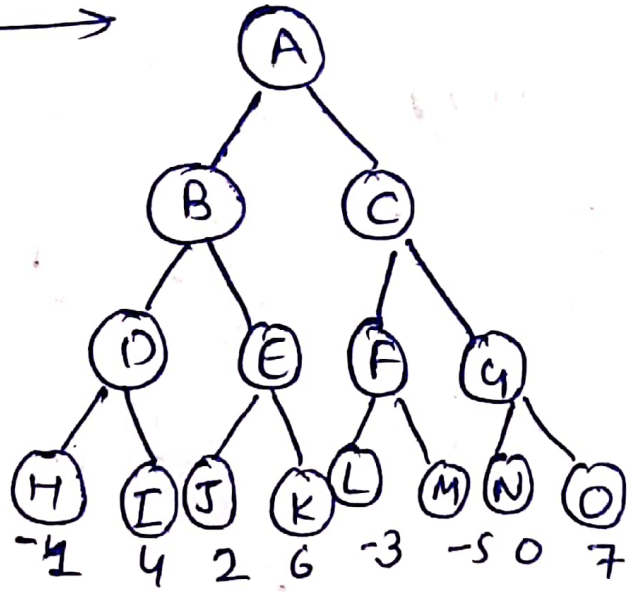
Working

- algorithm mein DFS apply karenge or terminal node tk jayenge, terminal node ki value compare karenge jo values hongi or backtrack karenge jab tk initial nhi ajati.

Steps – (1) Generate game tree. make utility function to get utility values.

Maximizer first turn lega with worst carse (-∞) value and minimizer - best (+∞)

Initial state →



Max

Min

Max

terminal node

Utilities value for D Max (-1, -∞)

⇒ Max (-1, 4) = 4

For E = (2, -∞) = Max (2, 6) = 6

For F = (-3, -∞) = Max (-3, -5) = -3

For G = max (0, -∞) = (0, 7) = 7

(Max chunna hai utilities value mein

Scanned with OKEN Scanner

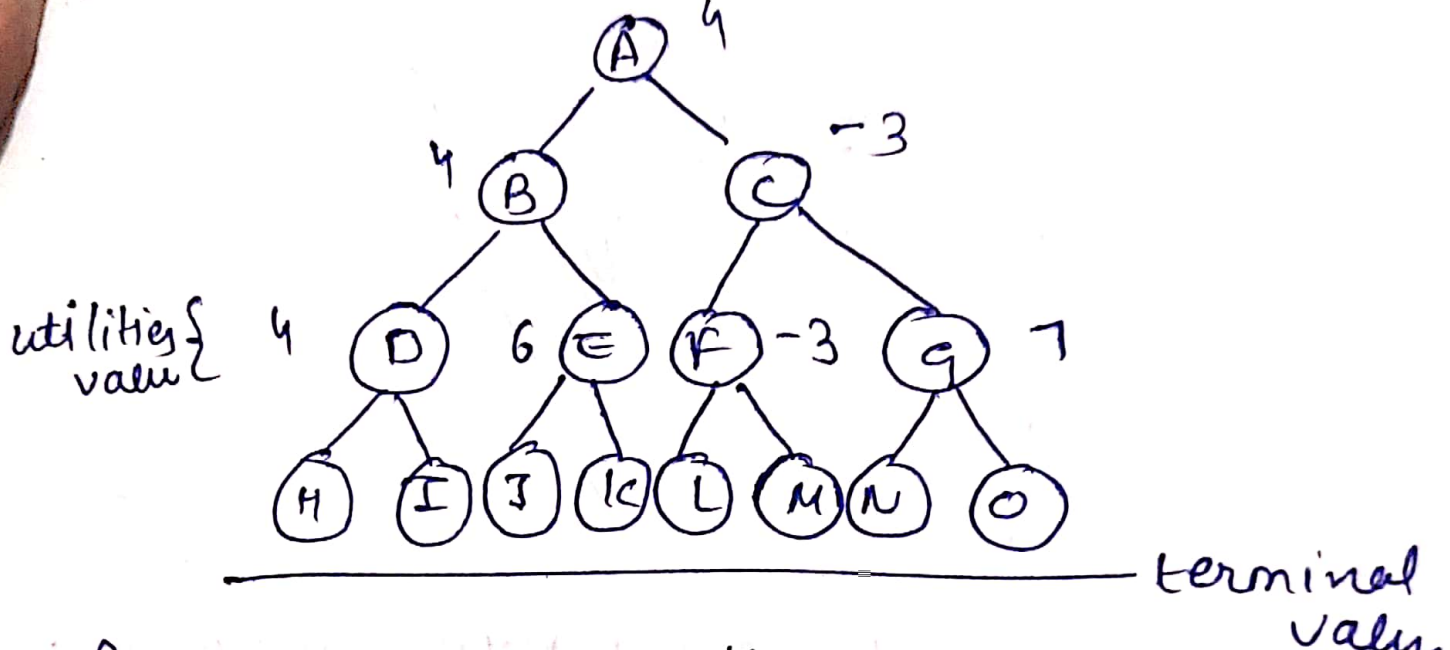
ab minimizer ki turn ayegi islie compare value with +ao.

For B = min(4,6) = 4

For C = min(-3,+7) = -3

Then se Maximizer ki turn aai

for node A = max(4,-3) = 4



Properties of algorithm

- Complete (soln jaroor milega agar Exist karta hai)
- Optimal
- Backtracking
- Recursive
- time complexity = $O(b^m)$ b → branching factor
- Space complexity = $O(bm)$ m → max depth.

Scanned with OKEN Scanner

Limitations -

- Slow

Alpha Beta Pruning

↳ Modified version of Min-Max algo

↳ without checking each node, find correct Min-max decision - called Pruning.

ye 2 threshold parameter involve karti hai {Alpha Beta}

↳ Alpha → highest value (-∞) {Maximizer ka kaam karta hai}

Beta → lowest value (+∞) {minimizer ka kaam karta hai}

↳ it Removes all the nodes jo 'final Result ko affect nhi karti

condition for Alpha beta pruning $\alpha >= \beta$

Key Points

- alpha ki value sirf Max player update karega
- Beta ki value sirf Min player update karega.
- back track karte samay, node values upper nodes ko pass hongi {Alpha, Beta ko nhi}
- Alpha, Beta ki values child node ko dijayengi.

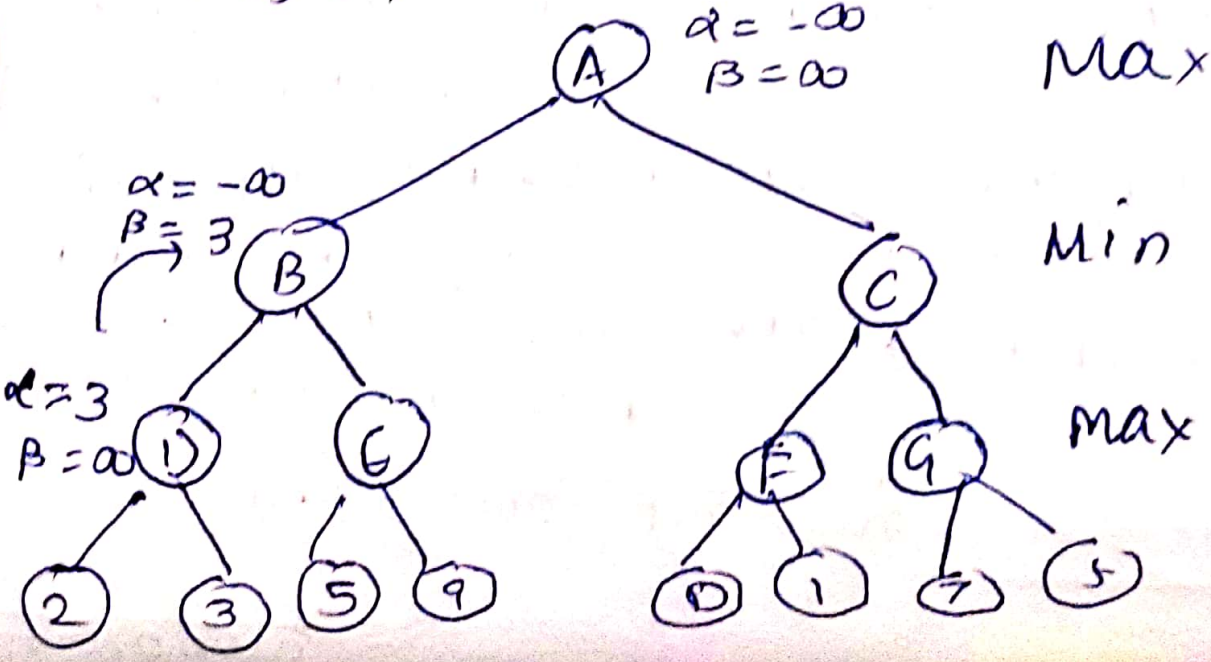
Scanned with OKEN Scanner

working -

(1) Max player A node start karega jaha $\alpha = -\infty$, $\beta = +\infty$ these value of α & β pass hongi B jaha pe $\alpha = -\infty$, $\beta = +\infty$, or node B child D ko dega.

(2) Node D, α ki value calculate karenge Max ki turn leke α ko compare karenge 2 se, 3 se or Max value nikalenge = 3. to $\alpha = 3$ bcoz D & node value will also 3.

(3) Backtrack karenge node B, jaha pe Min value find kareng means β ki $\because \beta = +\infty$. ko compare kareng $\min(\infty, 3)$ = 3 so, node B pe $\alpha = -\infty$, $\beta = 3$



Scanned with OKEN Scanner

ab $\alpha = -\infty$, $\beta = 3$ node E ko pass karenge

(4) E pr Max ki turn hai, islie α ki value change hogi, α ki value compare karenge 5 se so, $\max(-\infty, 5) = 5$ at node E $\alpha = 5$, $\beta = 3$ when $\alpha \geq \beta$ so, successor of E will pruned & algo will not traverse next at node E & value will be 5

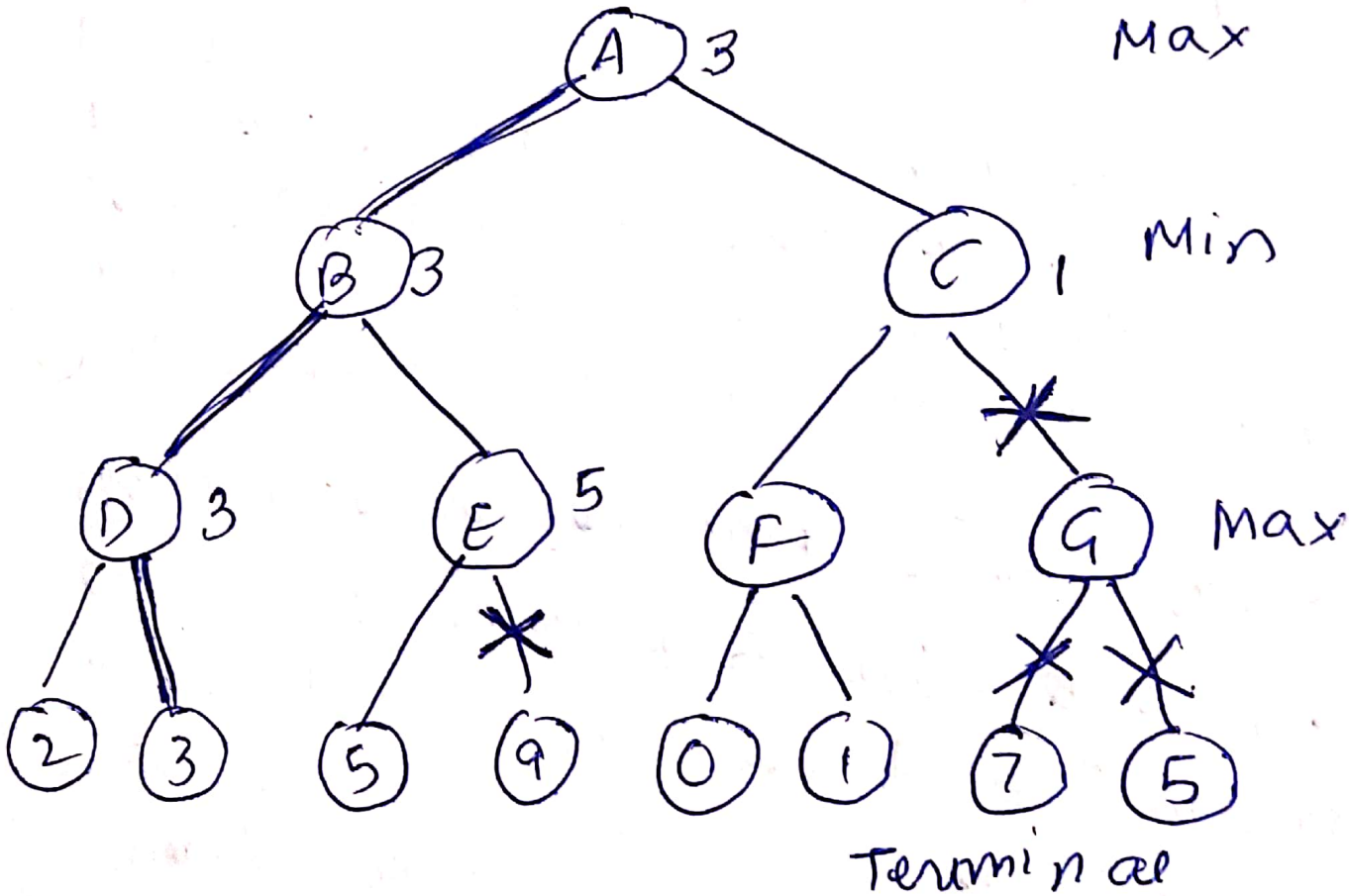
(5) next step mein Backtracking hogi, vapis A pe jayega & vaha Max play karega α ki value change hogi $\max(-\infty, 3) = 3$, $\beta = +\infty$. ab yeh value node A se node C pe jayengi

C → $\alpha = 3$, $\beta + \infty$ passed to node F.

At 5 α ki value change hogi kaunki Max play karega left child ke sath compare karenge value (jo ki 0) hai $\max(3, 0) = 3$ & uske baad Right se compare karenge $\max(3, 1) = 3$ but F node ki value 1 ho jaegi

Scanned with OKEN Scanner

Node F 1 value return karega node C ko
at C $\alpha = 3$, $\beta = +\infty$, β ki value change hogi
kyuki min Exist karta hai compare with ∞
$\min(\infty, 1) = 1$, at C = $\alpha = 3$, $\beta = 1$ again
it satisfies the condition $\alpha \geq \beta$. so,
G will be pruned & it will not
transverse next.



H K Chaudhary

Game development using AI –

(1) Adaptive dificultly - adjusting difficultly devel, optimal level, skill level.

(2) player behavior prediction - analyze Karta hai, behavior player ka, player patterns, allow developers to tailor gaming Experience.

(3) Emotion Recognition & Response -

Environment ke hissab se pta lagana what will be the next move of opponent.

(4) Pathfinding & Navigation -
help characters finding optimal paths & avoiding obstacles.

(5) Machine learning for Game testing -
used to automate testing process, identifying bugs, optimizing game performance & Ensuring better Quality assurance.

Scanned with OKEN Scanner

Dr. R K Chaudhary

Planning & Learning

↓

arranging a sequence of actions to achieve a Goal. It involves decision making about future action ek aise Environment mein jo ki aim karte hai particular target ko achieve karne ka

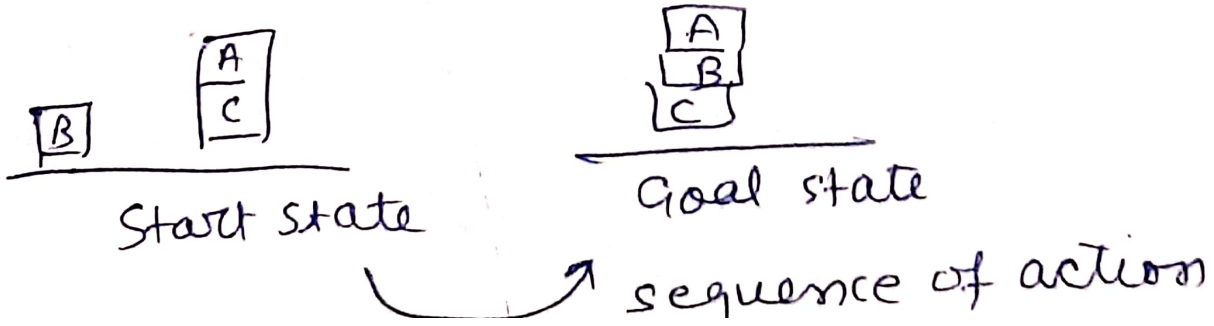
Related Key concepts -

- **Search Algo** - planning often involves searching through a space actions & their consequences & DFS, BFS, A*, heuristic commonly used to Explore Space.
- **State space Repersentation** -

  Problem ko terms of state, actions or transitions mein Repersent karna. or Sequen determine karna intial state se Goal state t
- **Forward & Backward planning** -

  ↳ Predicting the consequences of action from current to future actions.

Backward - from goal state work Backwards to determine actions lead to that state



Scanned with OKEN Scanner

machine learning -

↳ a subset of artificial intelligence jo Enable karta hai systems ko learn & improve karne ke lie Expirence se without being Explicit programmed

Machine learning types -

(1) Supervised
(2) unsupervised
(3) Semi supervised
(4) Reinforcement

(1) Supervised learning -

involves training algorithm using labled data { input se seekh ke output deti hai }

- Classification { Predicting discreate categories or class }
  - For Exp - Determine karo Email spam hai ya nahi (Catagorial / Numerical)
  - ho skta hai
- Regression { Predecting continous numerical values }
  - Exp - Estimating houses prices Based on features (only numerical value)

Scanned with OKEN Scanner

unsupervised learning

↳ Ek aise Type hoti hai machine learning mein jisme patterns, Relationship & structure predict karti hai, without specific guidance

↳ it includes clustering similar data points, detecting anomalies

↳ no Explicit or correct answers provided (Pehle se nhi milta kuch bhi) {unsupervised learning mein Kasak nhi milti Ha_ha_ha}

↳ clustering - Ek types ke items ko Ek sath group out karna

↳ Anomaly Detection - jo kisi Group mein Exist nhi karte unko Detect Karke delete Karna

Dr. R K Chaudhary

Reinforment learning -

by teach karna computer agent ko learn karne ke lie {trial & error method se}
Matlab dimaag lagao galtiyan karo
Recursion / iterative approach lago jab tk theek result na ajae.

isme as a Result punishment or Reward milta hai punishment - jb galat hote hai & Reward - jab shi hote hai, jisse yeh pta chlta hai ki agle ka step kya hoga!

Semi - supervised
semi means adha (Half)
to semi supervised meaning
Supervised + unsupvised.
user get both at a time labled data or unlabbled data & kai baar labelling of data modles ke lie mehanga padta hai to kuch cheeze unabbled hi chod di jati hai!! yeh model ki accuracy Badhane ke lie use hota hai Especially waha jaha pe labled data limited hota hai

Scanned with OKEN Scanner

Types of Games

	Deterministic	Non deterministic
outcome / Nature	outcome pta chal jata hai player ke behavior se. (Prediction se)	uncertainity hoti hai Kon jeetega
Ex	Tic tac toe, Chess	Card games, Poker
Strategy	Perfect Information & can decide their move	incomplete information Adaption to uncertainity
Complexity	strategies can be deeply analyzed	Complexity arises from inherent
Info availability	Complete info about the game	Info hidden or uncertain

Scanned with OKEN Scanner

Dr. R K Chaudhary

Games

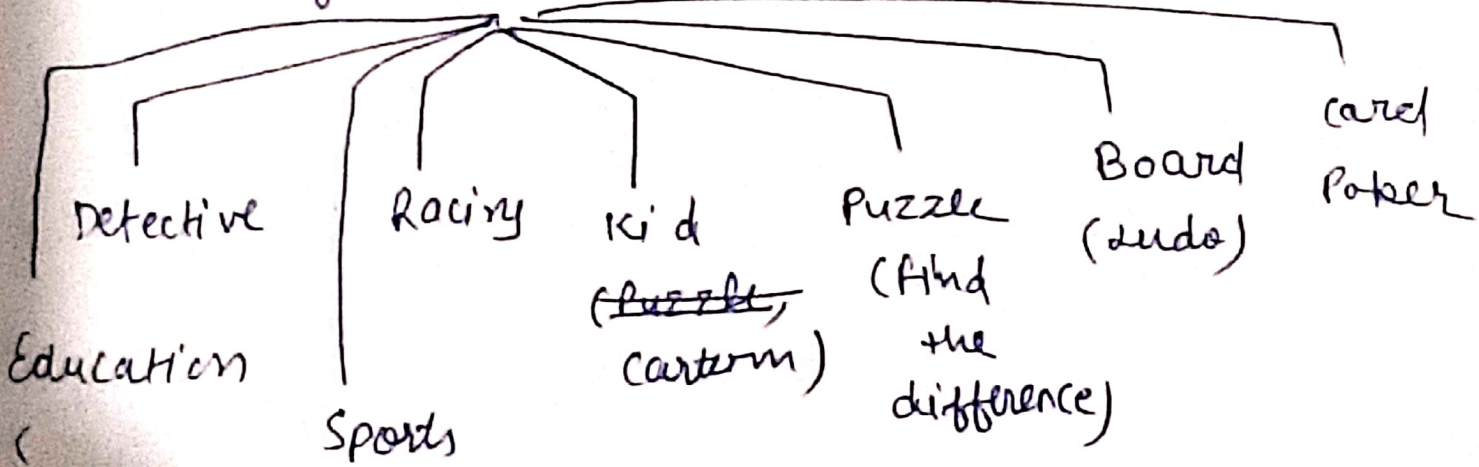
(1) AR Powered – visual & audio content, uses Gestures & body control
For Ex Pokemon Go

(2) AI ML – uses tech of AI ML
For Exp – Alpha Go, Stockfish

(3) Hyper casual – free & Easy to play

(4) Multiplyer – one or more player can play
Ex – New super Mario Bose

(5) Mobile game {Andriod & ios}



Predicate logic

↳ Extension of Propositional logical
↳ Known as FOL (First order logic)
↳ Includes → ① Predicates
→ ② Quantifiers.

Predicate - Relation b/w 2 obj.

Eg. Simba likes orange.

likes (simba, oranges)

General form likes (x, y)

Quantifiers → defines scope of objects

→ ① universal (∀) for all
② Existential (∃) There Exists

Eg ↳ Connection b/w ∀ & ∃

$\forall x$ (likes (x, orange)

$\sim \exists x \sim$ likes (x, oranges)

Scanned with OKEN Scanner

Resolution Example

1. $\forall_x \forall_y$ Eats $(x, y) \rightarrow$ food (y)

   $\forall_x \forall_y \sim$ Eats $(x, y) \vee$ food (y)

   $\sim$ Eats $(x, y) \vee$ food (y)

2. $\forall_y$ food $(y) \rightarrow$ likes (Milie, y)

   $\sim$ food $(y) \vee$ likes (Milie, y)

3. Bread is food.

   food (bread)

4. Mango is food

   food (Mango)

5. Neha Eats burger

   Eats (Neha, burger)

6. $\forall_x$ Eats (Milie, x) $\rightarrow$ Eats (Neha, x)

   $\sim$ Eats (Milie, x) $\vee$ Eats (Neha, x)

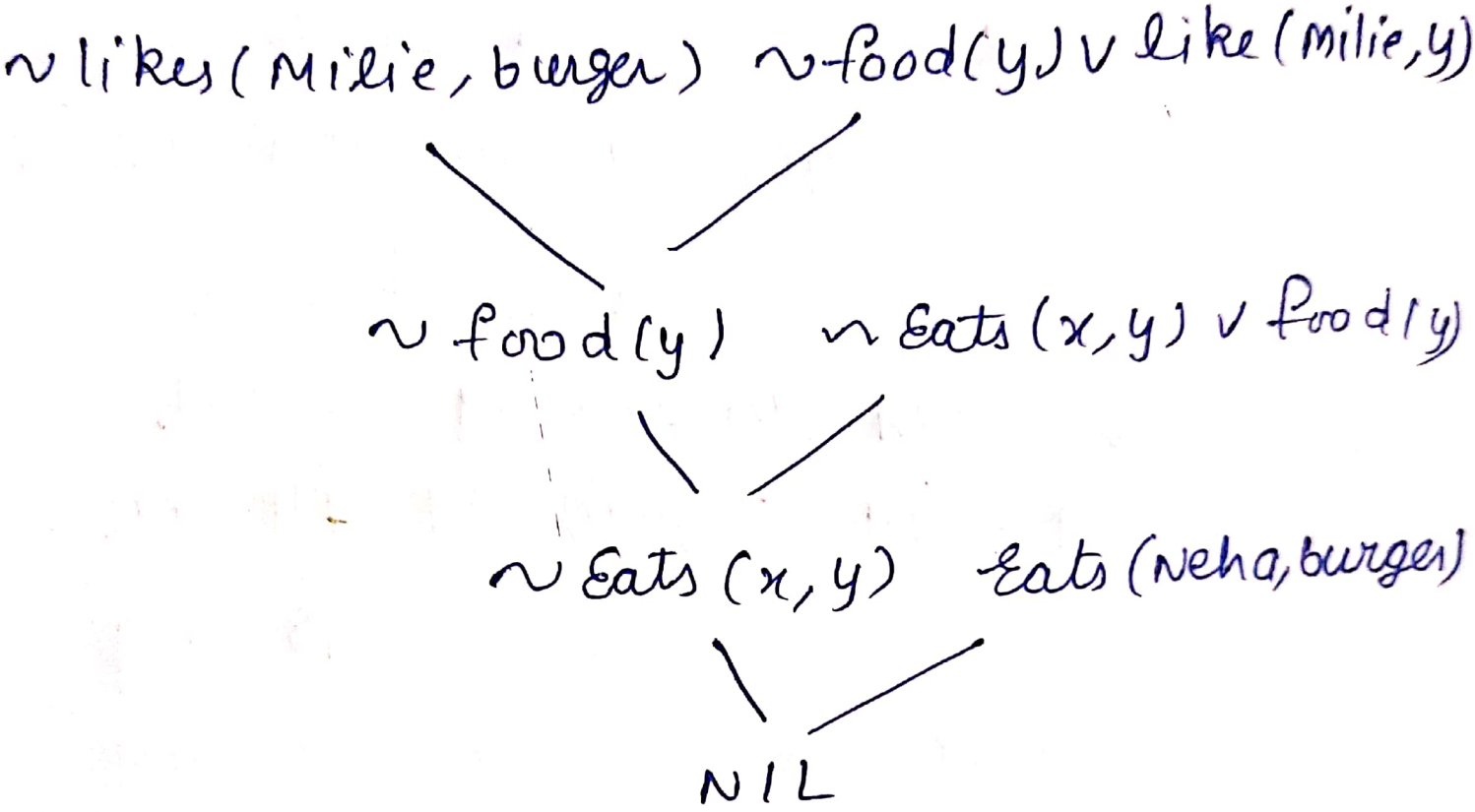
Scanned with OKEN Scanner

Dr. R K Chaudhary

Ques Does milie like burger?

Assume millie doesn't like burger

$\sim likes(Milie, burger)$



$\sim likes(Milie, burger)$ is Contradiction

$\therefore$ likes(Milie, burger) is true.

$\Rightarrow$ Milie likes burger.

Scanned with OKEN Scanner

Means End analysis

↳ Problem solving strategy used in AI

↳ Badi (complex) ~~forms ko Subgo~~ Problems ko choti problems mein convert karke or goals ko sub goals mein todke unko achieve karne ka tareeka means end analysis hai

↳ ye help karta hai to work systematically towards solution by Reducing the gap between current & End state.

Process

(1) **Identify the Goal** - define the Goal (Problem ko represent kare) or (aim to solve problem)

(2) **Identify current state** - understand From where you are starting or Resource Kya hai problem ko solve karne ke.

Scanned with OKEN Scanner

(3) Identify Subgoals - Breakdown goals into more manageable ~~Goals~~ subgoals.
(jo ki goal ko achieve karne ke close le jae)

(4) Identify Means to achieve subgoals - what resources or actions are acquired to achieve the subgoals. (steps needed)
Considering means or methods to address goals.

(5) plan & Execute - create a plan that outlines the order in which you will address subgoals and actions or steps needed for each one. Execute ur plans by working towards these goals.

(6) Iterate & adjust -
Means end analysis Ek aisa process hai jisme jitna age badhoge utne obstacles ayenge un obstacles ko clear karne ke lio process ko iterate karo & adjust kro jab tk main goal pr na pauch jao.

Scanned with OKEN Scanner