CO-2

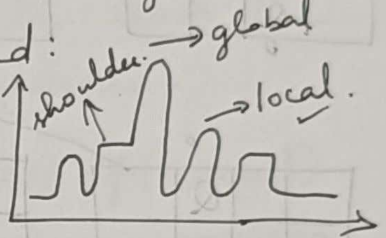
Local Search Algorithms

- focuses only on goal state & not the prev. states.
- very little memory.
- find reasonable soln. in large state spaces.

(ex) Hill climbing | Local Beam Search
Random walk | Genetic algorithm
Simulated annealing.

Hill Climbing search :-

→ depends on the initial state the global maxima (~~glob~~ goal state) is reached.

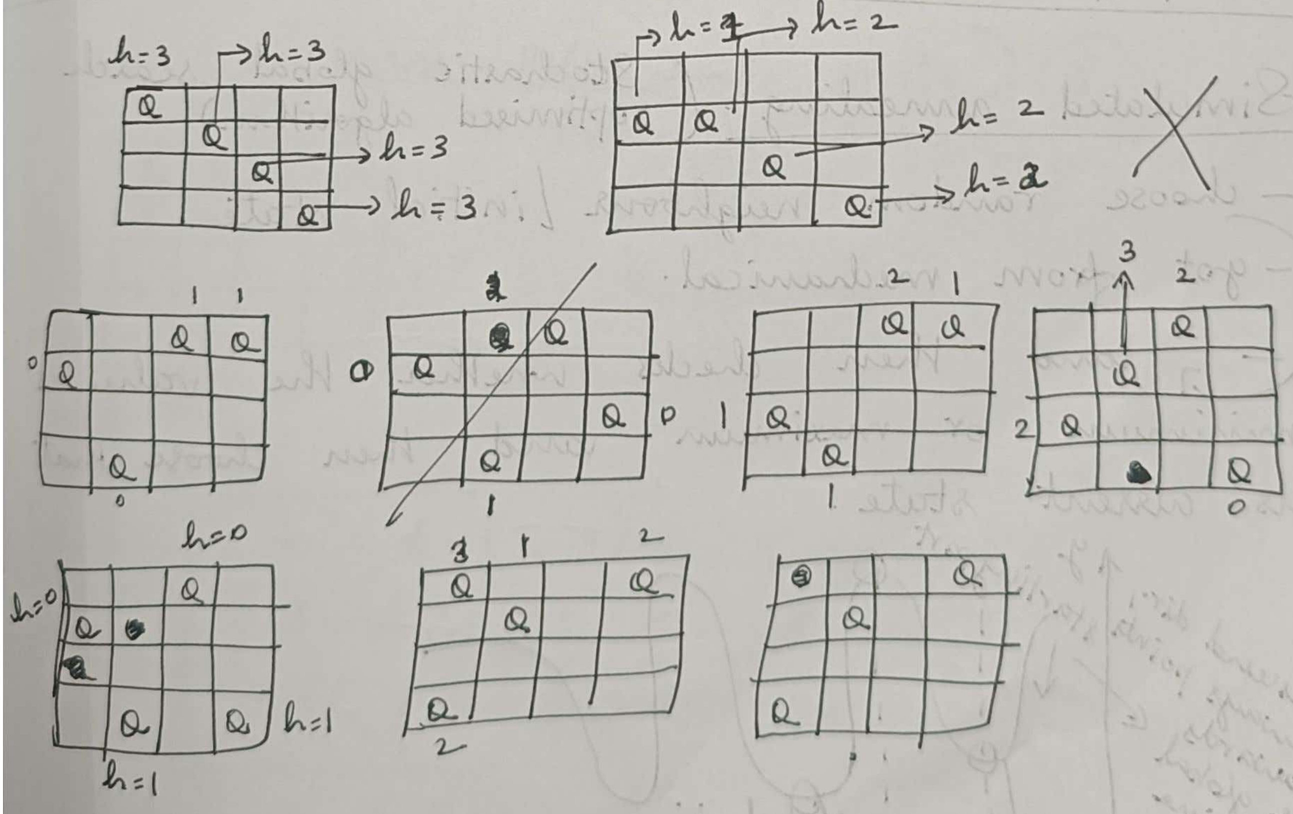


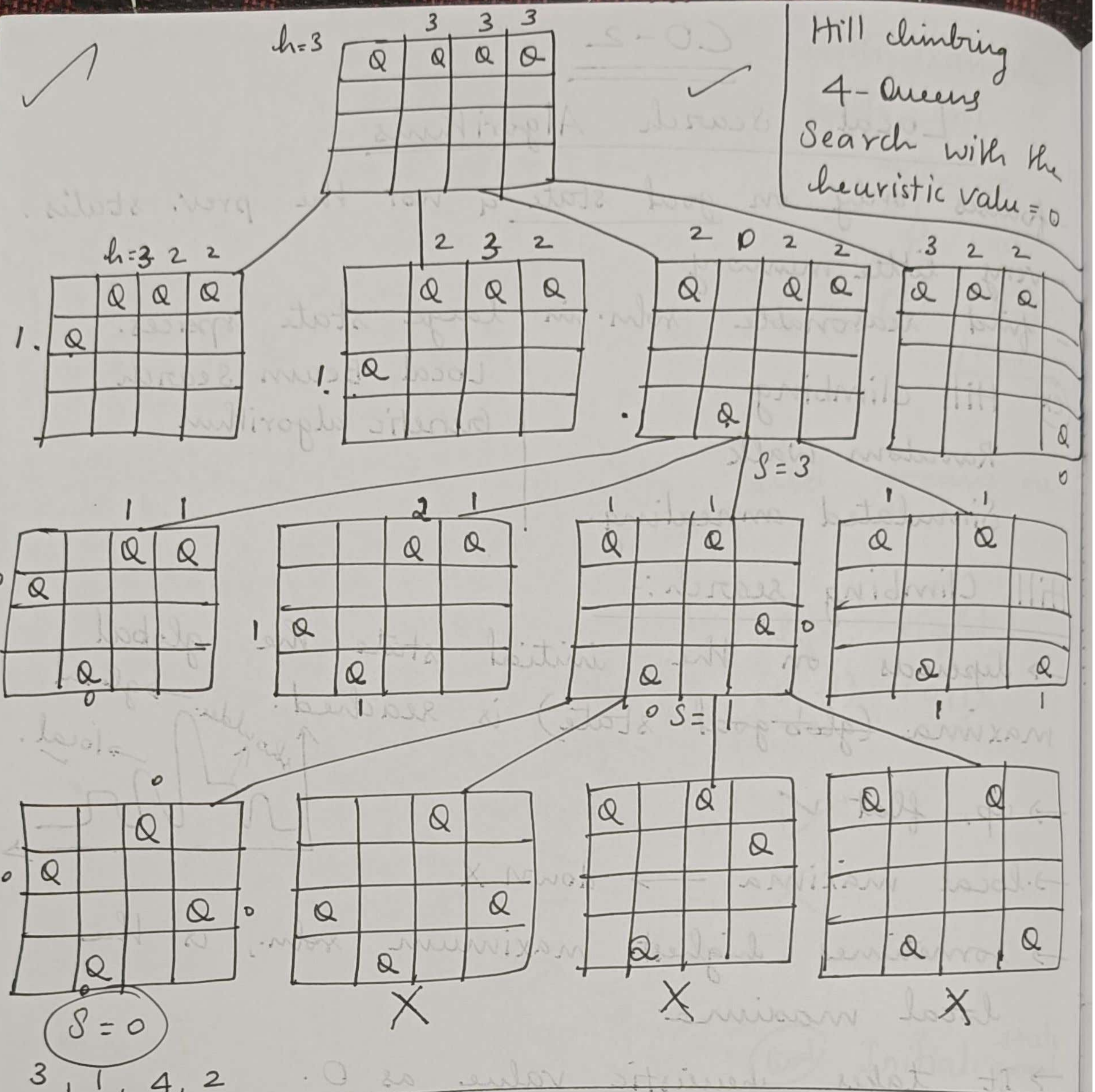
→ up, flat ✓

→ local maxima ⟶ down X

→ sometimes highest maximum soln., is the local maxima.

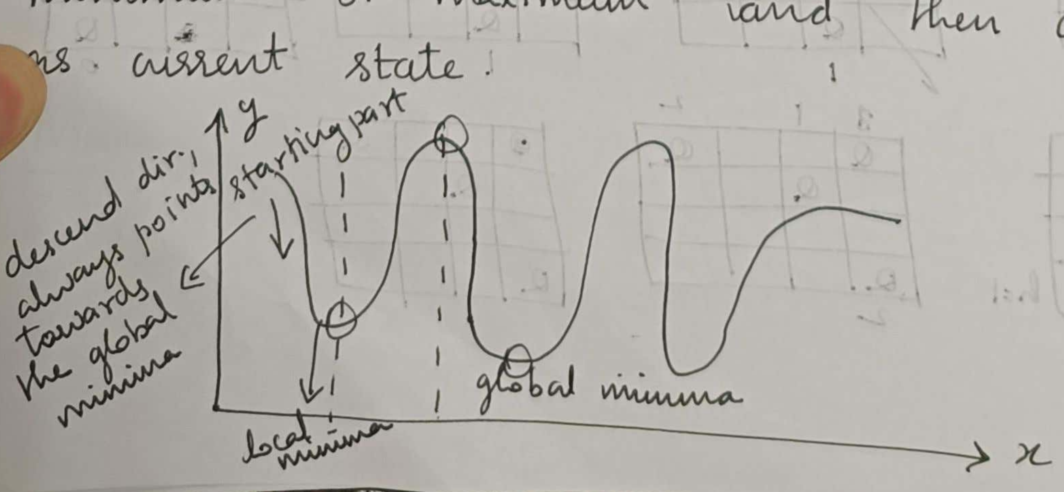
→ It takes heuristic value as 0.





Simulated annealing :- (Stochastic global search optimised algorithm)

- choose random neighbour / initial state.
- got from mechanical.
- → and then checks whether the value is minimum or maximum and then choose that as current state.



- Some action is made in the local minima to achieve the global minima is called as the stimulated annealing.
- uses a time stamp.
- a modified version of stochastic hill climbing.

(problem, schedule)
↳ mapping from time to temperature.
↳ problem

(ex) If there is a raw material (iron) in local minima it is heated (melted) and then cooled down to get a structure where a shaking happens and we are supposed to reach the global minima from ~~global~~ local minima.

$\boxed{P = e^{\Delta E / T}}$, $\Delta E = f(\text{successor}) - f(\text{current})$

considers.
$\Delta E > 0$ (current state as initial state)
$\Delta E < 0$

(ex) $f(\text{current}) = 16$

$f(\text{successor}) = 15$

$\Delta E = -1$, $T = 20$, $P = e^{-1/20} = 0.9512$.

(Please Refer and write) ✱

I. State

Q	Q	Q	Q
3	3	3	3

Given $S = [0,0,0,0]$

$T = 100$

$\alpha = 0.90$

α – ?

$f(S) = \dfrac{3+3+3+3}{2} = 6.$

(conflict b/w pairs)

Note

$$\boxed{\begin{array}{c} T_n = \alpha * (T_n - 1) \\ P = -\Delta E / T \\ \Delta E = f(s)' - f(s) \end{array}}$$

$\cancel{T_n = 0.90 - 99}$

$\cancel{P = \Delta E / T}$

$\cancel{P = \frac{2}{100}}$

$= 4 - 6 = -2 < 0$

$T = 0.90 \times 100 = 90$ (accept)

Step 1

	Q	Q	Q
Q	3	2	2
1			

$S = [1, 0, 0, 0]$

$$f(S') = \frac{1+3+2+2}{2} = 4$$

Step 2

$T = 90 \quad f(s) = 4$

		Q	Q
Q		1	2
1	Q		
	2		

$$f(s') = \frac{1+2+1+2}{2}$$

$$= \frac{\cancel{0}6}{2} = \cancel{2.5}\ 3$$

Step 3

		Q	Q
Q		1	1
1	$\cancel{2}$		
	Q		$\cancel{\ }$
	0		

$$f(s') = \frac{2}{2} = 1$$

$\Delta E = 1 - 3 = -2 < 0$

$T = 0.90 \times 81 = 72.9.$

~~A* Search~~

$\cancel{\boxed{f(n) = g(n) + h(n)}}$

Step 4

$S = [1, 3, 0, 0]$

$S' = [1, 3, 0, 2]$

Q		Q	
			Q
	Q		

$T = 72.9$

$f(S) = 1$

$f(s') = 0$

$\Delta E = 0 - 1 = -1 < 0$

$$f(s') = \frac{1+3+0+2}{2} = \frac{5}{2}$$

So accept with probability

$$f(s') = \frac{1+1+1+2}{2} = \frac{8}{2}\ 2.5 = 3$$

$P = -\Delta E / T$

$\Delta E = 3 - 1 = 2 > 0$

$P = 0.9729 <$

Steps $S' = \{1, 3, 0, 2\}$

$S = [1, 3, 0, 1]$

$f(s') = 0$

$T = 65.61$, $\Delta E = 0 - 3 = -3 < 0$

$T = 65.61 \times 0.90 = 59.049$

		Q	
Q			
			Q
	Q		

Constraint Satisfaction Problems:- (CSP)

components → Variables
→ Domain
→ Constraints

- can't schedule two classes at a time.

(ex) Map colouring, Sudoku.

8 - Queens CSP:-

$Q_1, Q_2 \text{ ------ } Q_8$ — var

row placement $\{1 \text{ to } 8\}$ — domain

$$\left.\begin{array}{l} Q_i \,!= Q_j \; (j \,!= i) \\ |Q_i - Q_j| \,!= i - j \end{array}\right\} \longrightarrow \text{constraints.}$$

→ Backtracking based (DFS) ⟶ all possible solns.,

Crypt arithmetic:-

- char / num should be unique.
- each char must have unique value.
- digits should be from 0 - 9
- start char of num., shouldn't be 0.
- addition of num. with itself is always even.
- carry can only be '1'.

~~SEND + MONEY~~

ex

$$\begin{array}{r} S\ E\ N\ D \\ +\ M\ O\ R\ E \\ \hline M\ O\ N\ E\ Y \\ \hline 1\ .\ .\ .\ . \end{array}$$

$$\begin{array}{r} 4\ 3\ 2\ 1 \\ +\ 4\ 3\ 2\ 1 \\ \hline 8\ 6\ 4\ 2 \end{array}$$

S	
E	
N	
D	
M	
O	
R	

$$\begin{array}{r} T\ O \\ G\ O \\ \hline O\ U\ T \\ 1\ \ \ \ \end{array}$$

SEND $\boxed{9}\ \boxed{5}\ \boxed{\ }\ \boxed{\ }$

MORE $\boxed{1}\ \boxed{0}\ \boxed{\ }\ \boxed{\ }$

MONEY $\boxed{1}\ \boxed{0}\ \boxed{5}\ \boxed{\ }\ \boxed{\ }$

10

ex

$$\begin{array}{r} T_2\ O_1 \\ +\ G\ O_1 \\ \hline O\ U\ T \\ \boxed{1}\ \ \ \ \ \ \end{array} \qquad \begin{array}{r} 2\ 1 \\ +\ 8\ 1 \\ \hline 1\ 0\ 2 \end{array}$$

T	2
O	1
G	8
U	0

ex

$$\begin{array}{r} S\ E\ N\ D \\ +\ M\ O\ R\ E \\ \hline M\ O\ N\ E\ Y \end{array}$$

$$\begin{array}{rcccc} & S & E & N & D \\ & \boxed{9} & \boxed{5} & \boxed{6} & \boxed{7} \\ + & \boxed{1} & \boxed{0} & \boxed{8} & \boxed{5} \\ & M & O & R & E \\ \hline \boxed{1} & \boxed{0} & \boxed{6} & \boxed{5} & \boxed{2} \\ M & O & N & E & Y \end{array}$$

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

- Two characters ~~of~~ won't have the same value.

Back Tracking Search:- (Success / Failure)

① assignment, => var = value

② Inference (forward checking)

(ex) A = Red

A, B, C

Red, Blue, Green

Then forward check and remove 'Red' from Database.

Then, assign,

(DOMAIN)

B = {Green, Blue}

C = {Green, Blue}

D = {Red, Blue, Green}

A = Red ✓ consistent

B = Red X Inconsistent.

Local Beam Search:-

- keeps track of K states rather than one.
- begins with K-random states.

(ex) Choosing best outgoing student with academic.

① select states from country with best engineering result.

② select k cities with the best result.

③ select k colleges. with the best result.

④ select k divisions " " "

⑤ select k students who scored max. marks.

⑥ select a student among them with max-marks.

(ex) 4 Queens Problem

(Refer slide)

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

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

	$\boxed{9}$	$\boxed{5}$	$\boxed{6}$	$\boxed{7}$
+	$\boxed{1}$	$\boxed{0}$	$\boxed{8}$	$\boxed{5}$
$\boxed{1}$	$\boxed{0}$	$\boxed{6}$	$\boxed{5}$	$\boxed{2}$

$\cancel{0} + E \cancel{+ 0} = N$ (when carry $= 0$)

× $E = N$

- No two chars. can have same value.

$1 + E + 0 = N$ (when carry $= 1$)

$\boxed{1 + E = N} \rightarrow$ ①

consider,

$N + R = E$ (when carry $= 0$)

$\cancel{0} + N + R = E + 10$

Sub ①,

$\cancel{0+}\ 1 + \cancel{E} + R = \cancel{E} + 10$

× $\boxed{R = 9}$ We cannot take because 9 is already assigned to 'M'.

$1 + N + R = E + 10$

$1 + \cancel{E} + 1 + R = \cancel{E} + 10$

$\boxed{R = 8}$ ✓

D + E = Y + 10

5 + 6 = 11 ✗

5 + $\boxed{7}$ = 12 ✓

6 + $\boxed{7}$ = 13 ✓

So, Y may be 2 or 3

7 can be for D or E.

So, allot D = 7,

Game Playing

- Minimax → b^m $(O(b^m))$
- Alpha beta coding

Minimax alg. procedure:-

- Zero - sum
- have perfect information.
- discrete
- finite
- deterministic

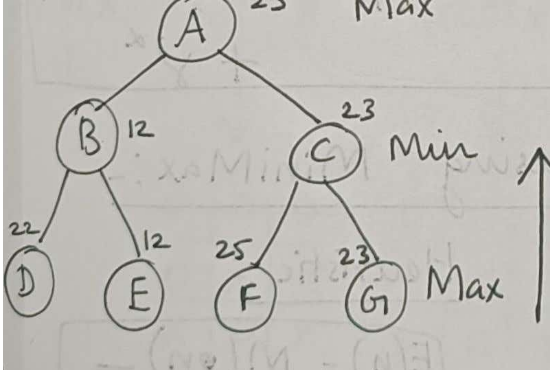
(5) Types of Games

- DFS applied + Backtracking + Recursive

(ex) Player 1 ⇌ Player 2 (Maximize)

Player 1 → Maximize

Player 2 → Minimize

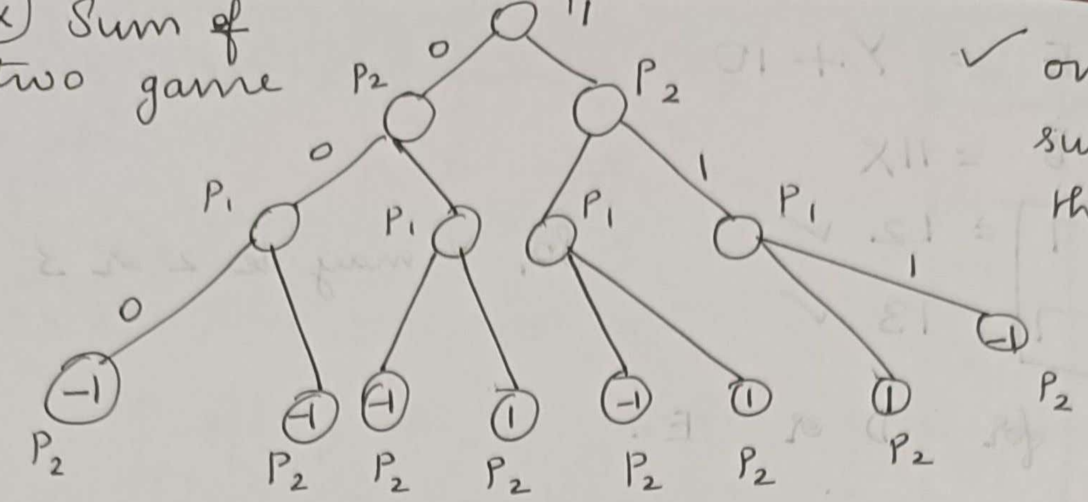


— Based on the child node it is moving to upwards.

- utility values like 0/1 for sucess/failure for terminal nodes, 1-win, 0 - draw, -1 - loss.

(ex) Tic Tac Toe game

ex) Sum of two game



✓ only if sum of is 2 then ans is '1'

Board Games:-

1. Initial State
2. Goal State
3. Path cost
4. Actions

fast - DFS
Time & Space - BFS
Optimize - A* Search

no. of nodes

Utility function:- Used to map the each state of the terminal nodes.

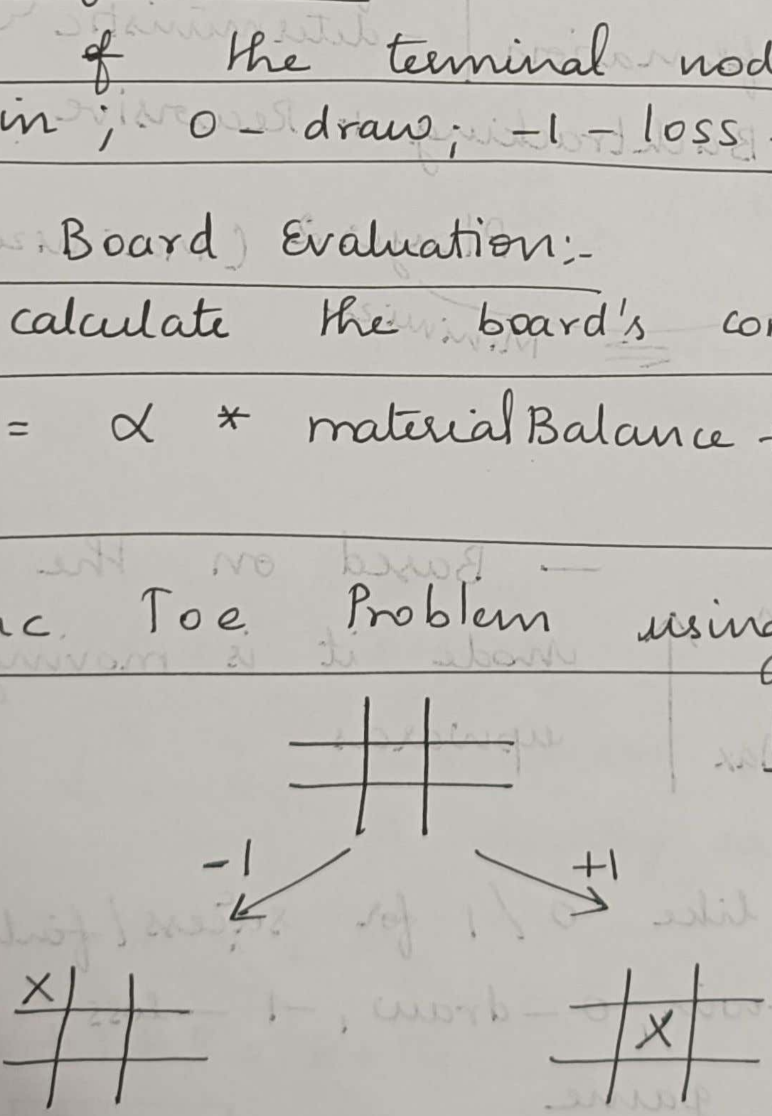
$\boxed{1 - \text{win} ;\ 0 - \text{draw} ;\ -1 - \text{loss}.}$

Static Board Evaluation:-

- to calculate the board's configuration.

$$\boxed{SBE = \alpha * \text{materialBalance} + \beta * \text{centre control} + \gamma^{\alpha}}$$

Tic Tac Toe Problem using MiniMax:-



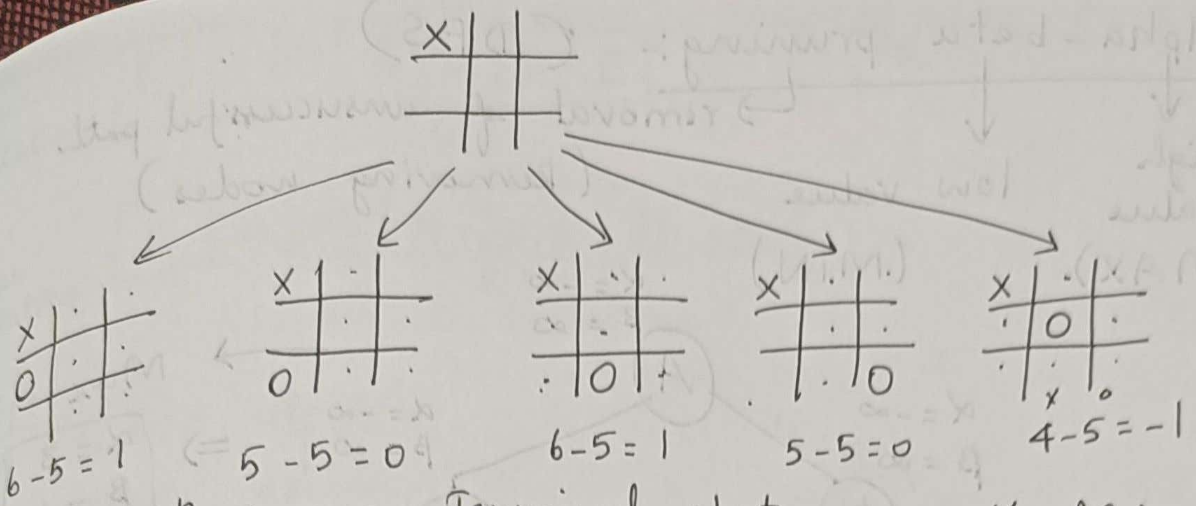
Heuristic :-

$$\boxed{E(n) = M(n) - O(n)}$$

(Total)

Me

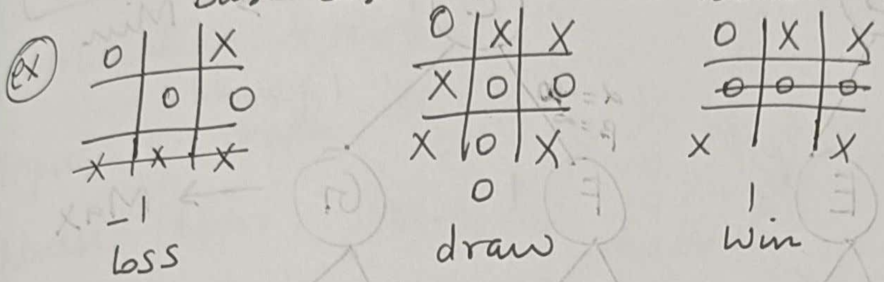
opponent

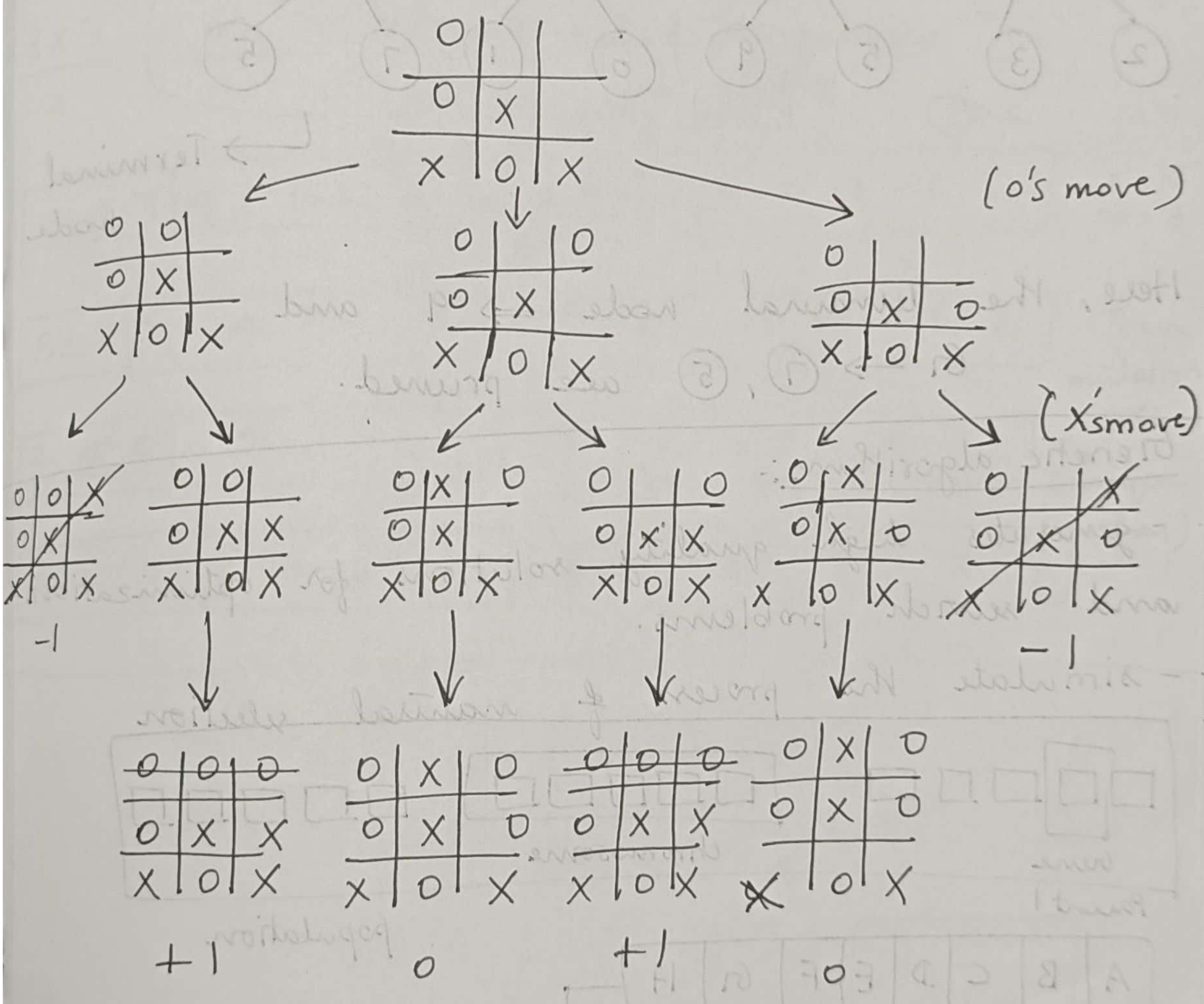


Base-case - Terminal state

X - Min

O - MAX





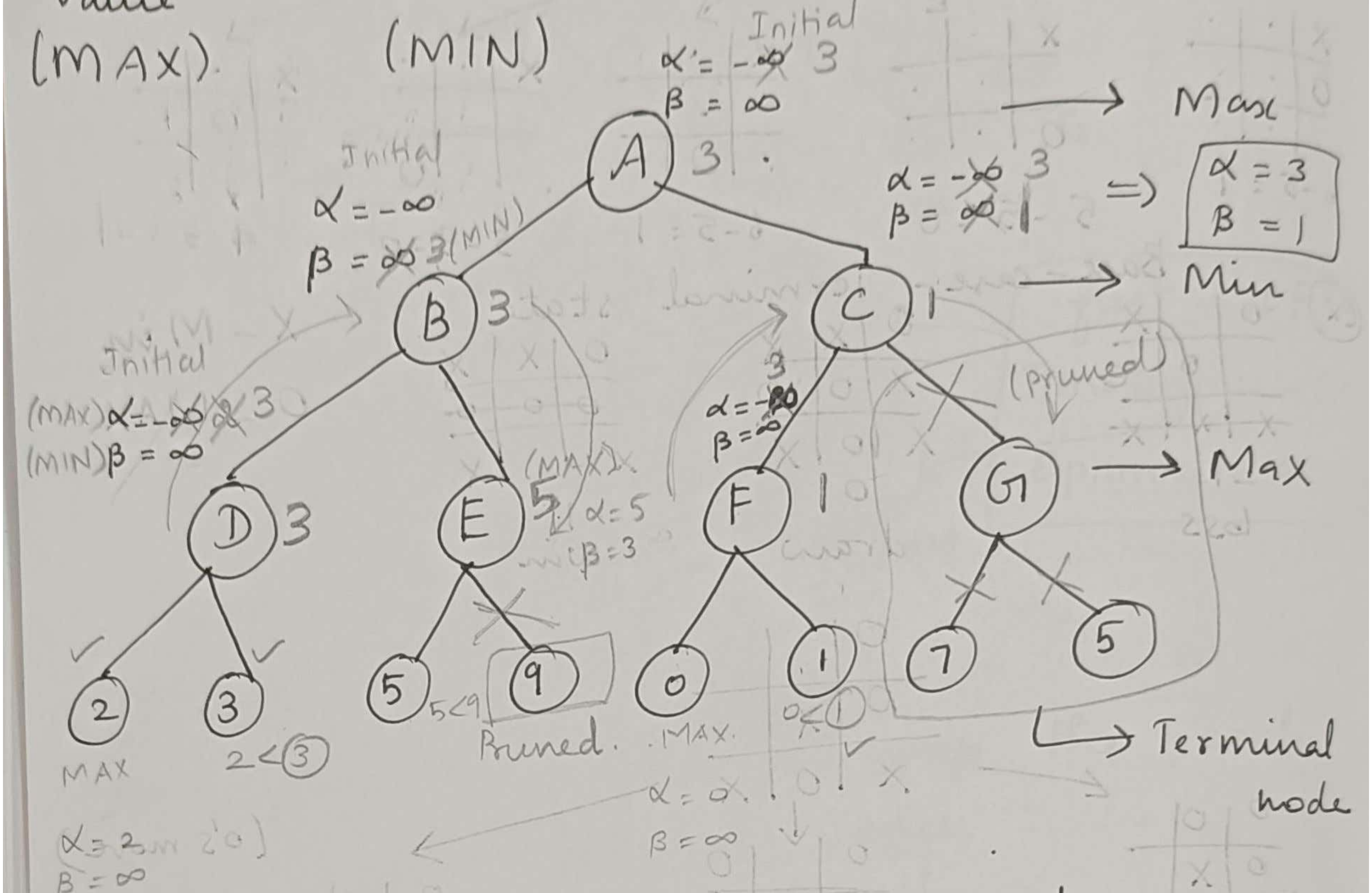
(O's Move)

Alpha-beta pruning:- (DFS)

high value (MAX)

low value (MIN)

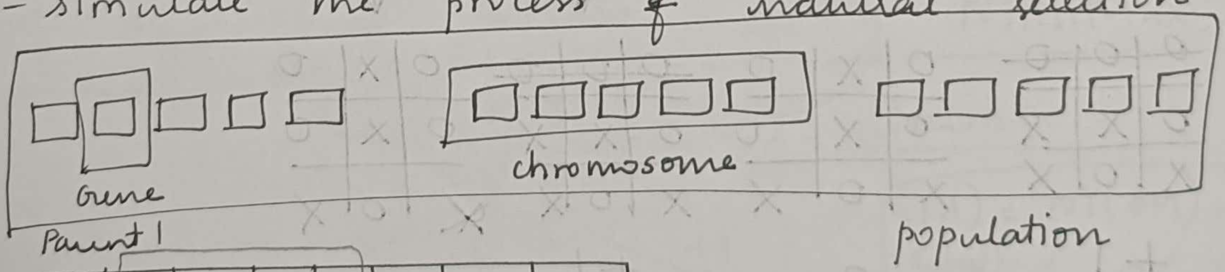
pruning ⟶ removal of unsuccessful path (Removing nodes)

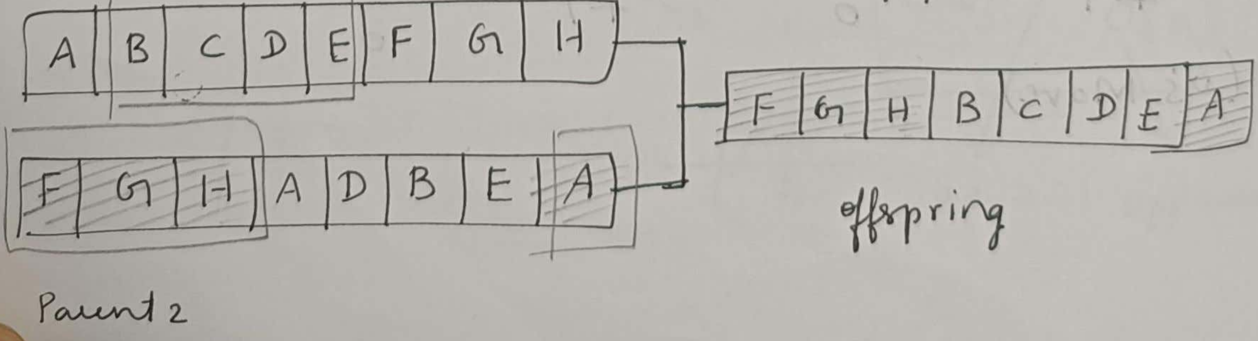


Here, the terminal node ⟹ 9 and G ⟶ 7, 5 are pruned.

Genetic algorithms:-

- generates high quality solutions for optimization and search problems.
- simulate the process of natural selection





Before mutation

F	G	H	B	C	D	E	A

After mutation

F	G	M	B	C	D	E	N

Operators:-

1. Selection operator
2. Crossover operator
3. Mutation operator

Input: k-states

fitness: Non-attacking queens

$$\frac{8 \times 7}{2} = \frac{56}{2} = 28 \text{ non-attacking queens}$$

fitness % + probability

Parents	Offspring
247 | 48552	327 | 52411
327 | 52411	247 | 48552
244 | 15124	327 | 52411
325 | 43213	244 | 15124

Then mutation goes (unique char)

CO - 3

Knowledge Based Agent :- (KBA)

Knowledge base. - includes all the rules, info and facts necessary for reasoning and decision

- KBA use stored info. and reasoning tech., to make intilligent decisions and solve pblms.

(ex) Health care, Business Intelligence.

KBA → Knowledge Base
KBA → Inference engine

1. Facts
2. Rules
3. Updates (agent learns or introduces a new knowledge)

Sensors ⟶ senses
Actuators ⟶ perform actions

Functions of KBA

Functions of KBA → Tell (info)
Functions of KBA → Ask (query)
Functions of KBA → Perform (performs action)

Tell - The Door is open

Ask - Whether I need to close the door?

Perform - Any one option is selected.

TELL ⟶ ASK ⟶ ACTION (PERFORM)

WUMPUS WORLD PROBLEM :-

1. Pits ● (Breeze)
2. Wumpus (Stench)
3. Treasure ⊠
4. Breeze B
5. Stench S
6. Arrows → (Agents use to kill the Wumpus)
7. Agent

PEAS

Performance measure:- (Rewards or Punishments)

1. Agents get gold and return back safely
2. Agent dies (pit or Wumpus)
3. Each move of agent
4. Agent uses arrows.

Environment:-

- Agent initial pos = [1,1]
- A cave with 16 [4X4] rooms

Actuators:-

- Move forward : Move to next room
- Turn right / left : Rotate agent 90 degrees.

Sensors:-

(ex) Breeze detected near a pit.

Stench detected near a Wumpus.

Glitter detected when treasure is in a room

Scream triggered when wumpus is killed.

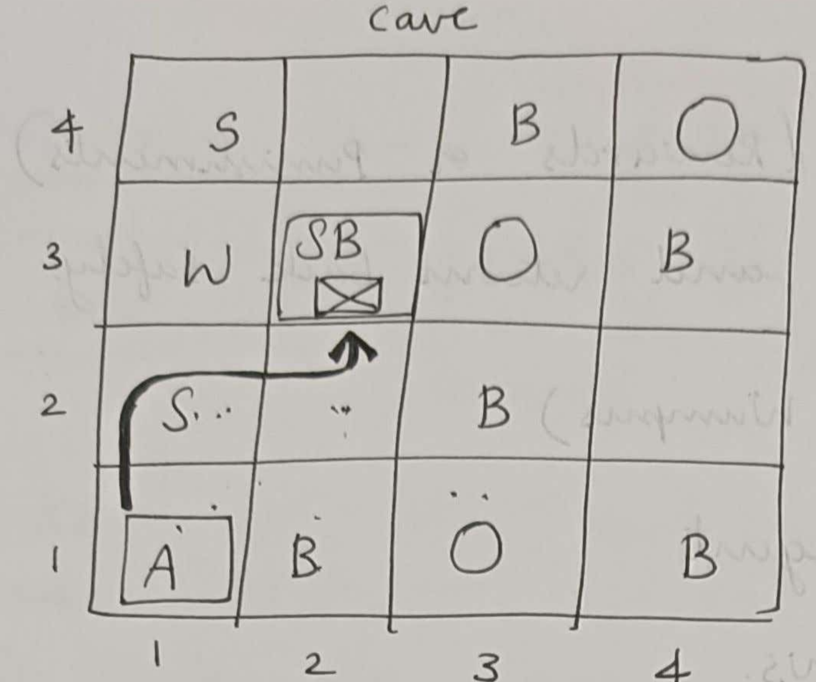
Bump a sound when agent hits a wall.

Logic

Logic → Propositional Logic (T/F) (PL)

Logic → First Order Logic (FOL)

- horn logics
- higher order logics
- three-valued logics
- probabilistic logics



S - Stench
B - Breeze
O - Pit
A - Agent
⊠ - Treasure
W - Wumpus

- Agent [1,1]

$$\begin{tikzcd} A \arrow[r] \arrow[d] & \boxed{B} \arrow[r] \arrow[d] & o \\ S & o & \end{tikzcd}$$

$$\begin{tikzcd} S & \square & \\ A \arrow[u] \arrow[r] & B \arrow[u] \arrow[r, "X"] & o \end{tikzcd}$$

__Goal Path:-__

A ⟶ S ⟶ Space ⟶ Treasure.

[1,1] ⟶ [2,1] → [2,2] → [3,2]

__Propositional Logic:-__ (PL)

- Logical constants
- Wrapping parantheses
- Literals. (P, ¬P)
- sentences are combined by connectives.

P - atomic   ¬P - non-atomic

¬ - negation
∧ - and
∨ - or
⟹ - implies
⟺ - equivalent

Formal Grammer :-

Sentence

Atomic (T/F)

Complex $(\wedge, \vee, \neg, \Rightarrow)$

$\boxed{\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow}$ → operator precedence.

ex $\neg A \wedge B = (\neg A) \wedge B$

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$.
F	F	T	F	F	T	T
F	T	T	F	T	T	F
T	F	F	F	T	F	F
T	T	F	T	T	T	T

ex

P	H	$\neg H$	$P \vee H$	$(P \vee H) \wedge \neg H$	$((P \vee H) \wedge \neg H) \Rightarrow P$
F	F	T	F	F	T
F	T	F	T	F	T
T	F	T	T	T	T
T	T	F	T	F	T

P – It is Hot   Q – It is humid

- A symbol is a sentence.
- It rep. semantics.

ex $(P \wedge Q) \longrightarrow R$

If it is hot and humid, then it is raining.

$Q \longrightarrow P$

If it is humid then it is hot.

- semantics determines the interpretation.
- evaluated based on " True value " based on the $\boxed{\text{knowledge base. (always true)}}$

Case 1 FFF ....

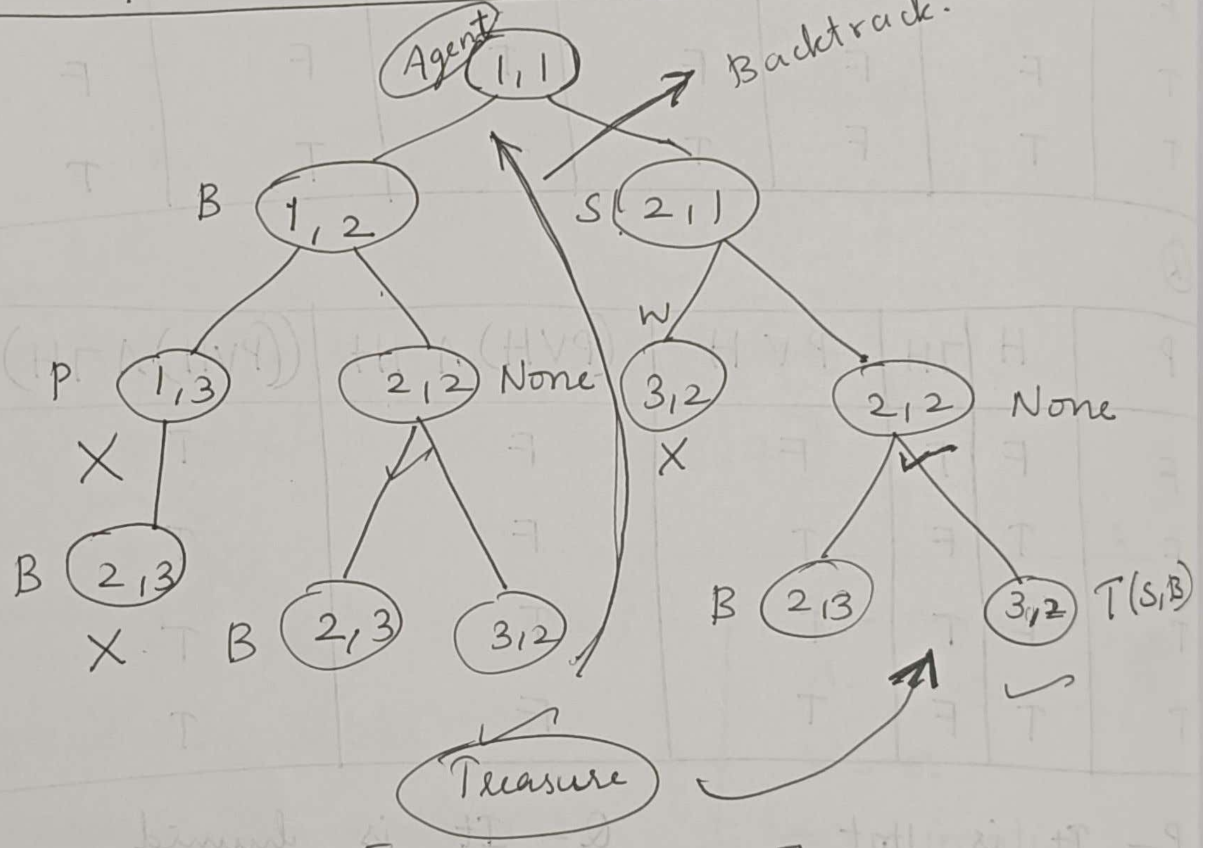
P = F Q = F R = F

Case 6 TFT

P = T Q = F R = T

$(Q \rightarrow P) \longrightarrow$ True.

State Space tree for Wumpus Problem :-



Rules of KB [Wumpus World] :-

So apply PL, FOL.

R1: $\neg P_{1,1}$

R2: $B_{1,1} \Longleftrightarrow P_{1,2} \vee P_{2,1}$

R3: $B_{2,1} \Longleftrightarrow P_{1,1} \vee P_{2,2} \vee P_{3,1}$

R4: $\neg B_{1,1}$

R5: $B_{2,1}$