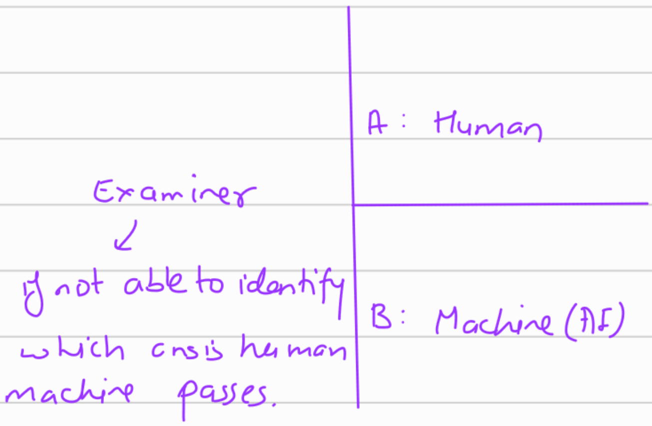
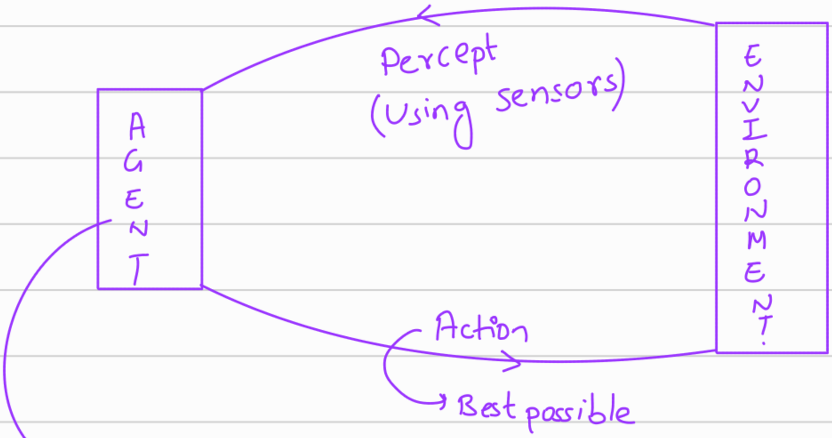
AI

- We try to build a AI which take best possible decision (Rationality to be maintained)
- To test intellegence of AI we use turing test.



- For a machine to be intellegent it must have some anylysing & reasoning skills.
- It should deal with natural language processing.
-



System which percive some environment & take some action to achieve a goal autonomusly.

- Performance of the agent determined using successful accomplishment of task.
- Environment : Observable — Non Observable

  can be det. by current I/P ← Deterministic — Stochastic → element of uncertainity at current state.

  Episodic — Sequential.

  Static — Dynamic.

- Agent v/s ideal agent v/s autonomous agent

  ideal agent → Rational

- Problem solving using search (Intelligent Search) →

1. State-Space Search →

- We need to define problem in form of state-space search.
- legal state, Action, Initial state, Goal state ⇒ parameter

legal state ↓ defined by environment boundry if it is inside it is legal else it is illegal

Action ↓ take certain action & move to next intermediate condition

Goal state ↓ when we reach goal state we mark it a goal state & move forward.

- Plan → Is a sequence of action to achieve a goal
- Path cost → The cost of taking a action

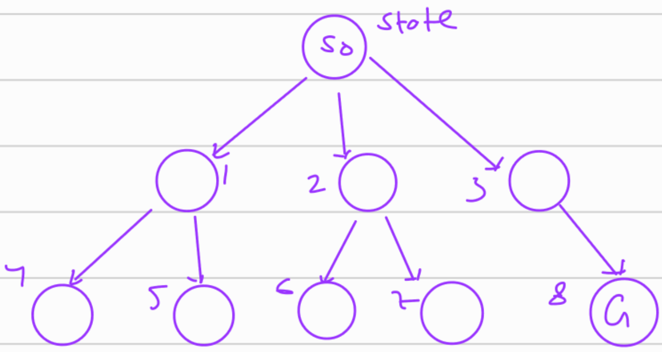
$S \rightarrow$ set of state

$S_0 \rightarrow S_0 \in S$ & S_0 is initial state

$A \rightarrow S \Rightarrow S$

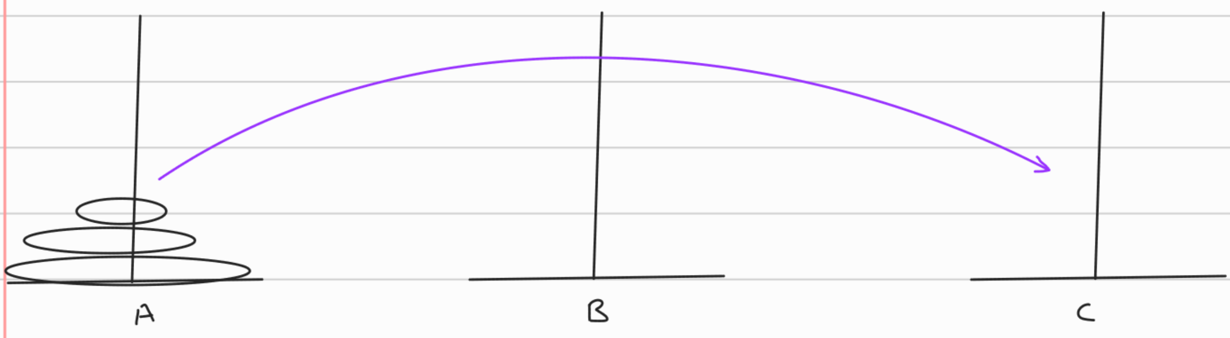
$G \rightarrow G \in S$ & G is goal state

- Problem is represented using directed graph where node = state & edges are action.



(Blind Search Numbring)

→ Pegs & disk problem → (small lies on larger always)



→ **8 Puzzle problem** →

5	4	
6	1	8
7	3	2

Initial

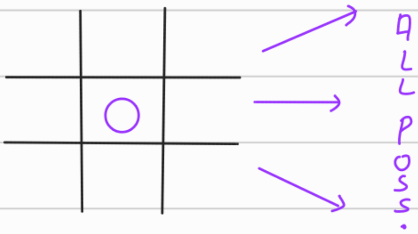
→

1	4	7
2	5	8
3	6	

Goal.

Action → move up, down, left, right.

→ **Tic-Tac-Toe** →



→ **8 Queen problem** →

Same as ADA

→ **Water-Jug problemp** →

Ex 3 Jugs of Cap 12, 8 & 3 & some water & we have to find out 1 Liter of water.

Steps →

1. List of state (Initial state)
2. Repeat
   ↳ Current State - goal state
   ↳ success
   ↳ Expand (Action)
   ↳ Current - goal state

issues →

↳ Expension (bounded/unbounded)
↳ Path or state (What to return)
↳ Steps or Cost
↳ Complexity (Space/Time)

Evaluating parameter for search →

1. Completeness → Successful
2. Optimality → Path cost / Path length (optimal solution)
3. Complexity → (Time/Space complexity)

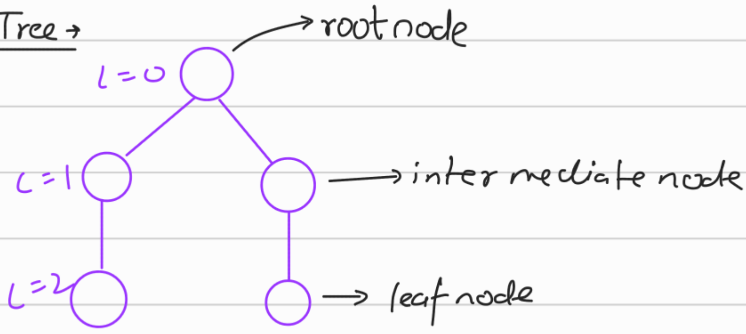
Types of Search →

(i) Blind Search or uniformed solution →

ⓐ BFS ⓑ DFS ⓒ Iterative depening Search ⓓ Iterative broadning search

(ii) Informed Search
(iii) Constraint Satisfaction Search
(iv) Adversary Search

① Search - Tree →



- branching factor
- Depth
- Path
-

- Breadth first search →

Already done

TC → $O(b^d)$
SC → $O(b*d)$

b = branching factor
d = depth

Same for DFS (∞ tree no terminate ∴ Not complete, Not optimal)

Limited depth search

We restrict depth & ↑ iteratively

called depth first iterate deepning (DFID)

Bidirectional Search →

Goal → initial & initial → goal meet some where in middle

Q. What if it is a graph?? - Nothing different.

Uniform cost search →

- variation of BFS, cost = g(n)
- Instead of queue it is priority queue (min heap)
- Its complete
- Complexity same as bfs

Informed Search →

↳ Heuristic

↳ Heuristic function $h(n)$

is the estimate of optimum cost from a node to goal.

- 8 Puzzle deviation from ans is manhatten heuristic distance

↳ $h(c) = \Sigma$ 1s deviation

$g(c)$ = height of tree

Best first search →

- Uniform cost search

Greedy Search →

- Est of lowest cost value.
- Find sol^n quickly - may not be optimal.

A* Algo → Cost function = $g(n) + h(n)$

$g(n)$ → path cost

$h(n)$ → node cost

2	8	3
1	6	4
7		5

⟶

1	2	3
8		4
7	6	5

$f(n) = 0 + 4 = 4$

$g(n) = 1 \quad \because h = 1$

↓

2	8	3
1		4
7	6	5

$f(n) = 3$

$g(n) = 2$

⟶

2		3
1	8	4
7	6	5

$f(n) = 3$

$g(n) = 3$

⟶

	2	3
1	8	4
7	6	5

$f(n) = 2$

$g(n) = 4$

↓

1	2	3
	8	4
7	6	5

$f(n) = 1$

$g(n) = 5$

⟵

1	2	3
8		4
7	6	5

$f(n) = 0$

$g(n) = 6$

Admissible/Underestimate

estimated Actual

$h(n) \le h^*(n)$ Underestimate

$h(n) > h(n)$ Overestimate

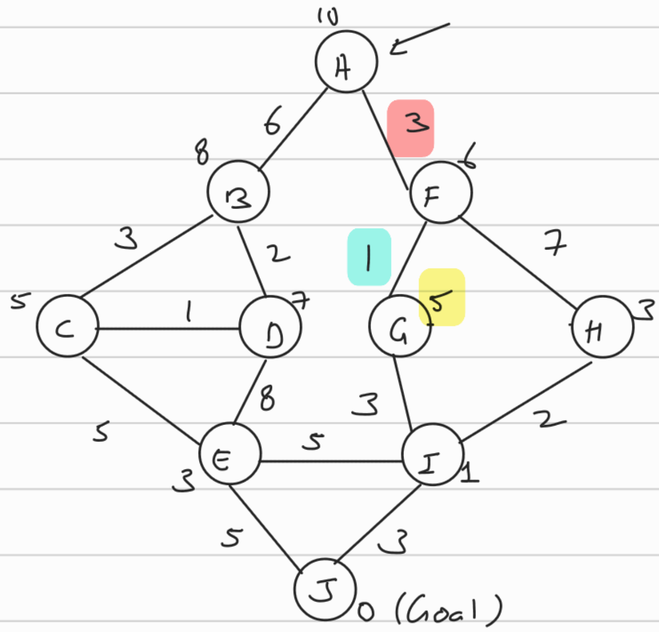
A* Algorithm→

- Greedy best search algo.
- open { s } close { }
- Cost function $f(n) = g(n) + h(n)$

$g(n)$: cost of reaching a node n from starting node

$h(n)$: hueristic / estimate cost of reaching the goal from n

- Till we get goal state
- Expand the node with min. cost path.



{A}

$f(B) = 6 + 8 = 14$

$f(F) = 3 + 6 = 9$ ← Min

A → F

{F}

$f(G) = (3 + 1) + 5 = 9$ ←

$f(H) = (3 + 7) + 3 = 13$

A → F → G

{G}

$f(I) = (3 + 1 + 3) + 1 = 8$

A → F → G → I

$f(E) = -$

$f(J) = (3 + 1 + 3 + 3) + 0 = 10$ → Optimal cost

A → F → G → I → J (Optimal path)

- In underestimate we explore solution furthur.
- Always optimal solution
- Provide solution if present
- Complexity is high in term of space & time.

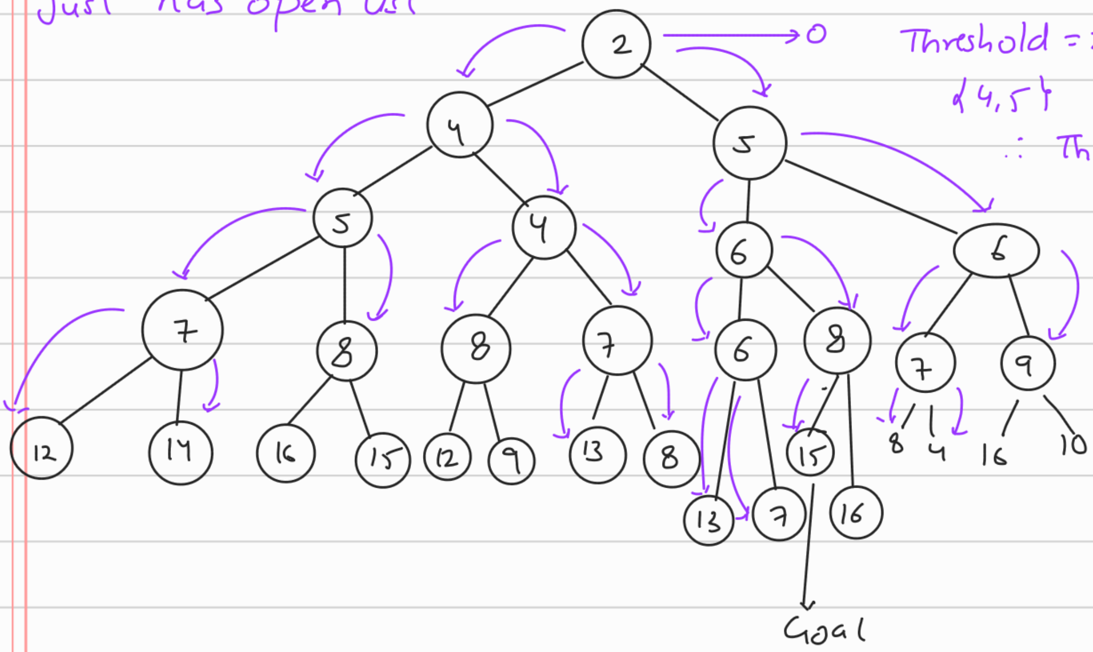
IDA* →

(Iterative deepning A*)

- Combine advantage of Iterative deepning DFS + A*
- Replace the depth with f-score ⇒ $f(n) = g(n) + h(n)$
- We take a threshold value.

Steps →

- Set the root node as the current node & calculate the f-score
- set threshold value.
  - Max f-score value that is allowed for expansion of successor nodes
- Node expansion
  - Expand current node & calculate f-score of successor
- Pruning
  - f score > threshold then we will not expand furthur.
- Goal node found.
  - If found search terminate else move back to step2 with new threshold
- Just has open list



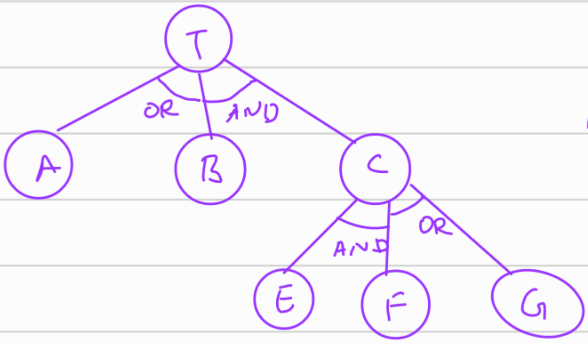
Threshold = 2
{4, 5}
NO Goal
∴ Threshold = min(4, 5)
= 4

S2: {5, 7, 8} ∴ threshold = 5
S3: {7, 8, 6} ∴ threshold = 6
S4: {7, 8, 13, 9} ∴ threshold = 7
S5: {12, 14, 8, 13, 9} ∴ threshold = 8
S6: we reach 15 i.e goal state

- Definately find search
- Avoid disadvantage of DFS
- It also admissible
- More time complex

AO* → (Informed search)

→ Works on idea of and-or tree



A or (B + C)

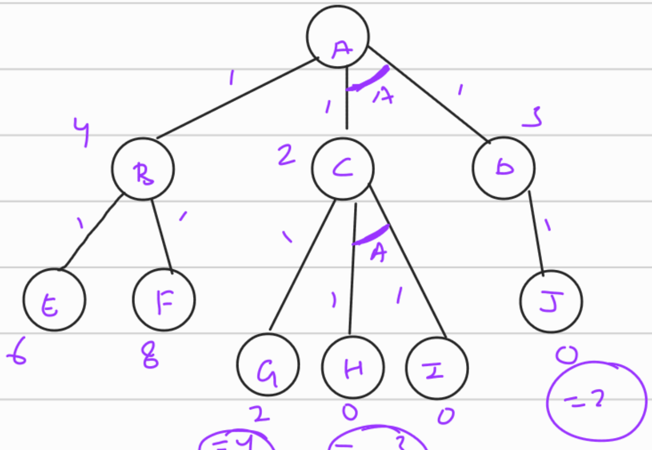
(E + F) or G

- We called $f(n) = g(n) + h(n)$

$g(n)$ → Reaching from start to 'n'

$h(n)$ → estimation of reaching from 'n' to goal.

- DFS till leaf
- Backtrack & update hueristic value



∴ C = 3

∴ D = 2

$f(A, B) = 1 + 4 = \{5\}$

$f(A, (C \cdot D)) = 1 + 2 + 1 + 3 = \{7\}$

$f(\overset{A \downarrow}{B}, E) = 1 + 1 + 6 = 8$

$f(\overset{A \downarrow}{B}, F) = 1 + 1 + 8 = 10$

B → min (8, 10) = 8

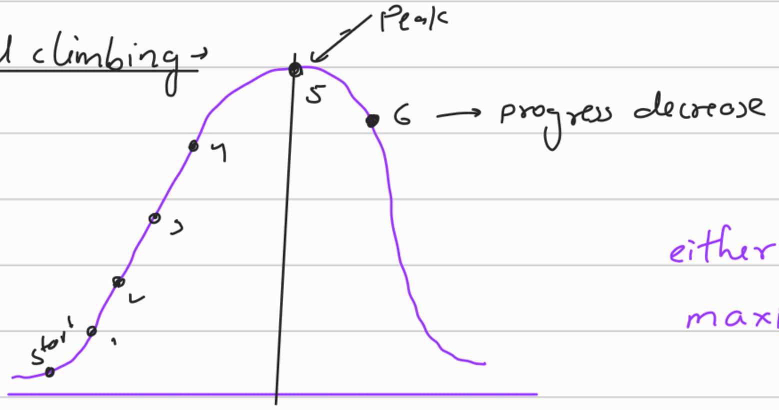
now B = 8, C = 3, D = 2

A → min (8+1, 3+2+1+1) = 7

Local Search →

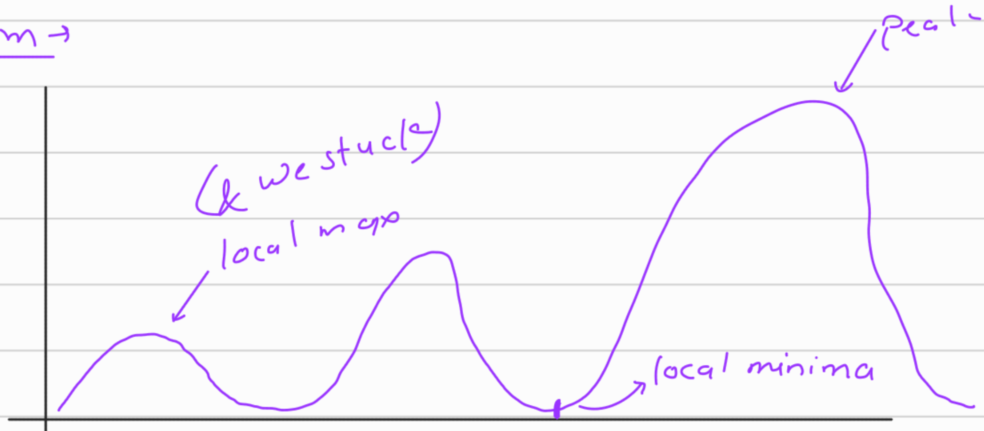
- Only current state in memory.
  - Hill climbing
  - Gradient decent & ascent
  - Simulated annealing.

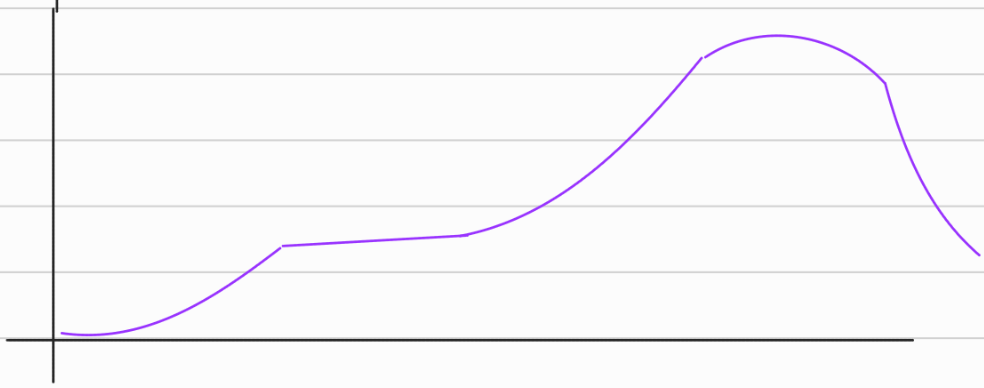
Hill climbing →



either minimization of maximization.

Problem →

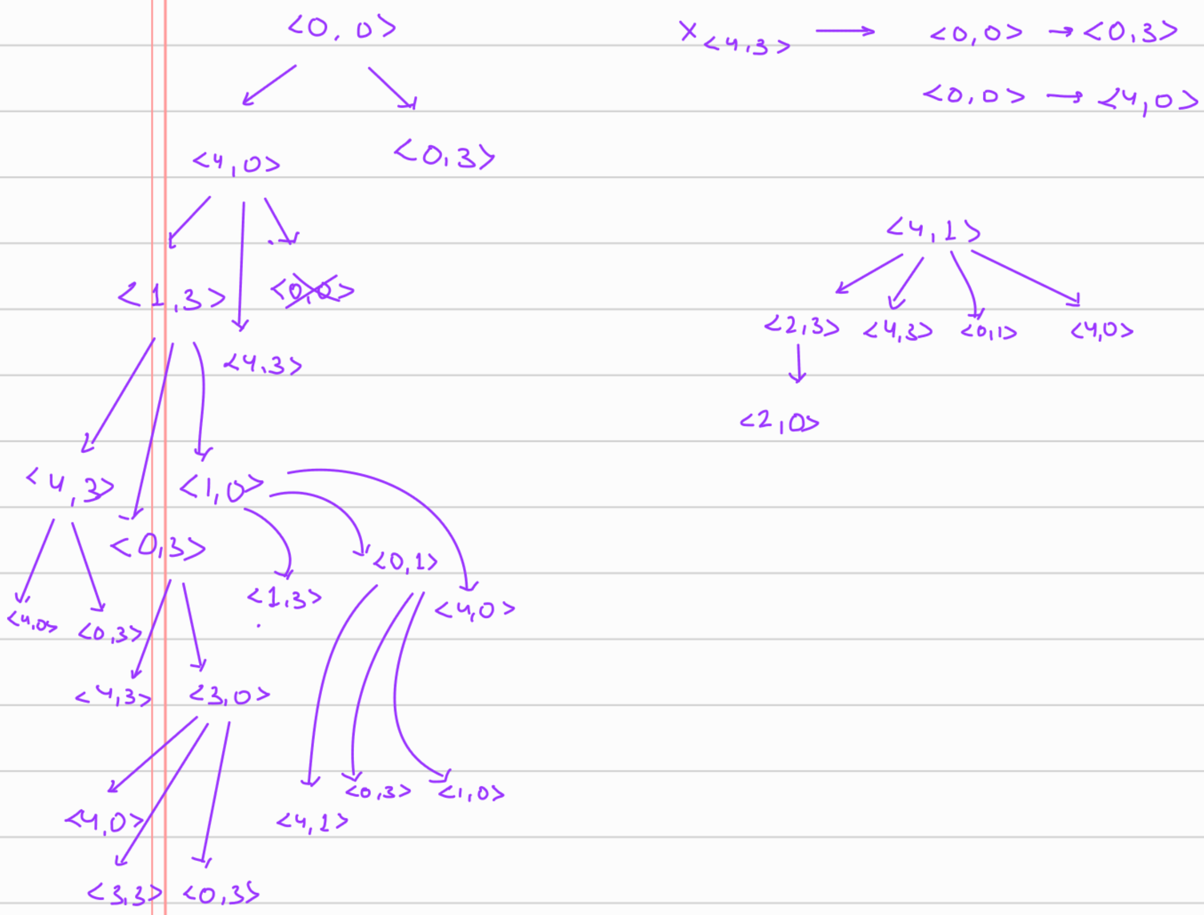




Production system →

Water jug problem →

JUG 4, 3 Goal = 2l in 4l Jug.
$\quad\;\; x \;\; y$



Adversial Search stretegy → (Multi agent Search)

- Applied when ∃ a compitition b/wⁿ agent
- In gameplaying this is generally used
- We can win or reduce chances of other player
- Comp. environment where multiple agent goal conflict which each othe → kind of game

↓

Mini-Max Algorithm →

- Kudh max win ya saamne min win karo
- Similar to DFS + backtracking

GAME - THEORY →

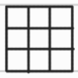
- Maximize chances of our winning symultaneously minimising opponent score

- Nature: Deterministic, turn by turn, perfect info sharing b/w player representation of current state.

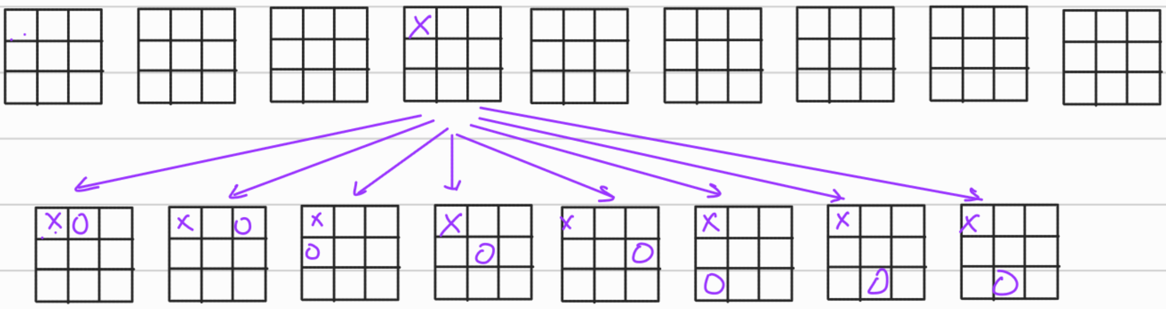
- Opponent will always introduce uncertainity in the game
- Chess is hard to predict since it branching factor is 35
- Assume avg chances to win make 50 move to win chess game
- Total possible state space = 35^{100}

- Always try to make optimal move.

Ex



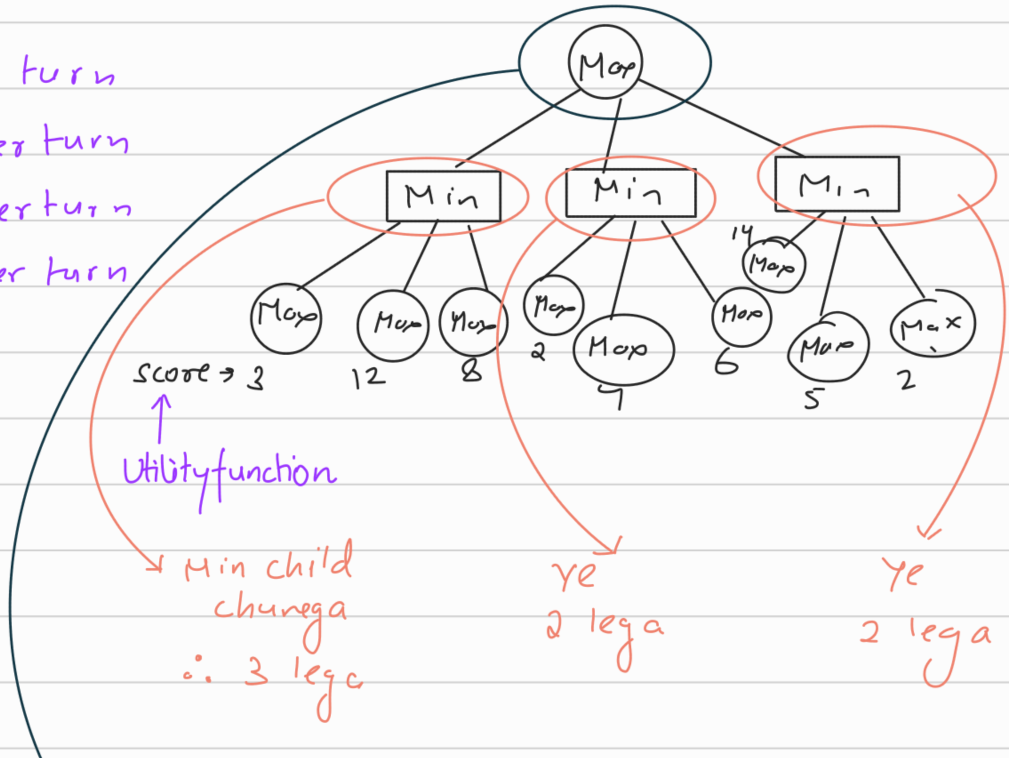
W = +1
L = -1
D = 0



If I win ∴ score = 1
If I lose ∴ Score = -1
If Draw ∴ score = 0

1st = Max player turn
2nd = Min player turn
3rd = Max player turn
4th = Min player turn
⋮
etc

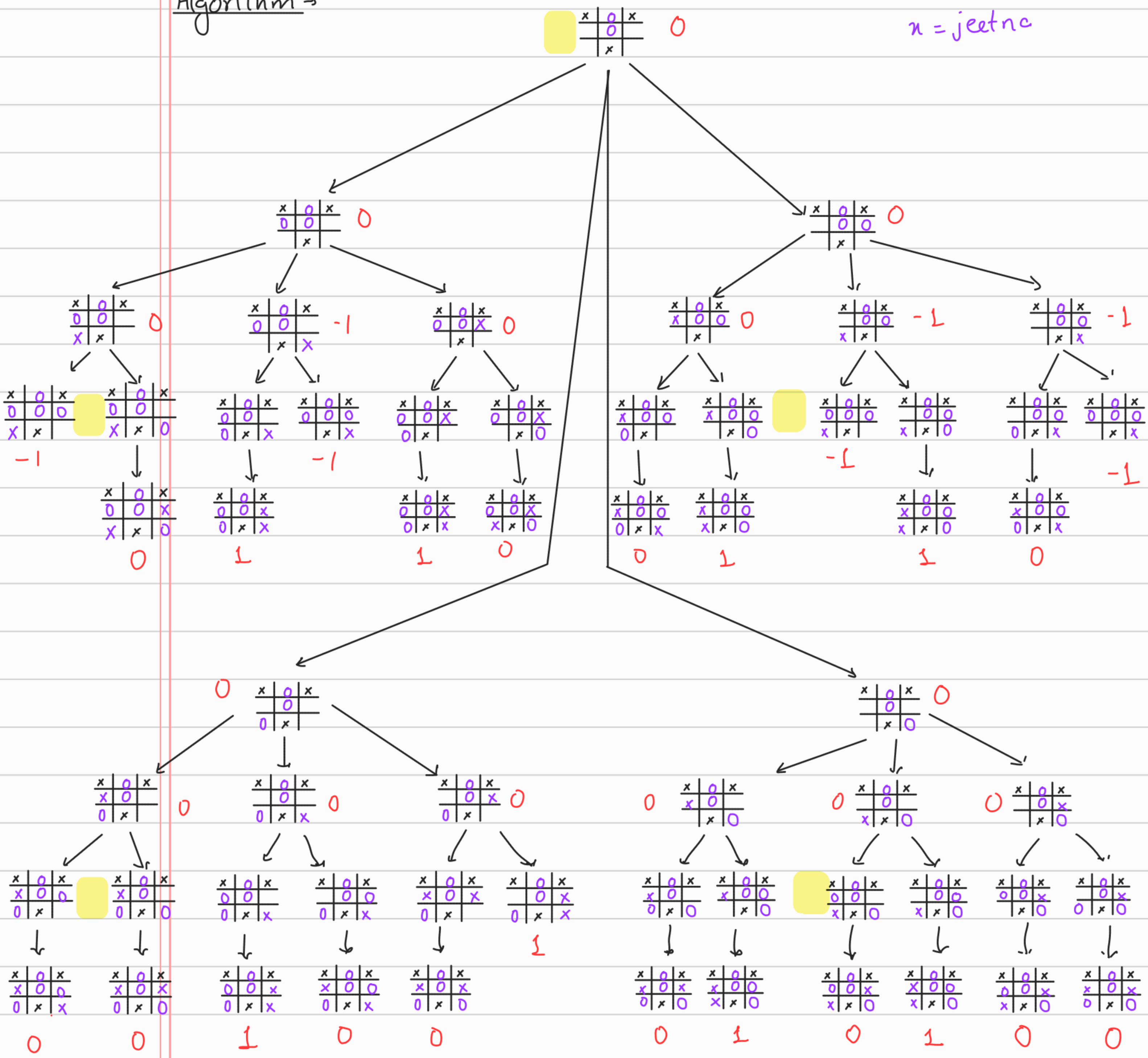
Move = ply



Maxlega (3, 2, 2) = 3

Algorithm →

x = jeetne



Minimax Algo →

function minmax (node, depth, Maximum)
    if (depth = 0 or node is terminal node)
        return value of node

  if Maxplayer
    max_eva = -∞
    for (each succeor of node)
        eva = minmax (succeor, depth+1, false)

max_eva = max (max_eva, eva)

return max_eva

else

min_eva = +∞

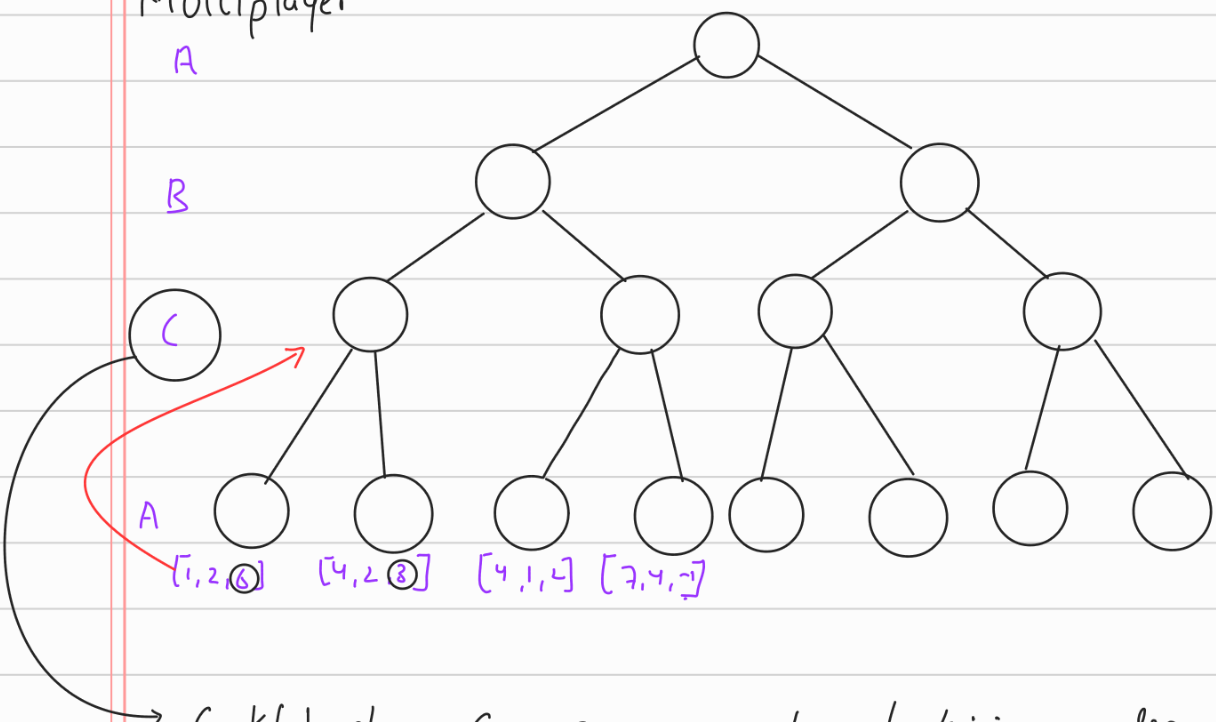
for (each succesor of node)

eva = minmax(child, depth+1, true)

max_eva = min(min_eva, eva)

return mineva

Multiplayer



→ C khel rah   C apna max lega baki jaane dega

Property of minimax →

- Complete algo.
- Optimal → each player move optimally
- TC ≈ DFS

- Disadvantage : We need to search complete state space search

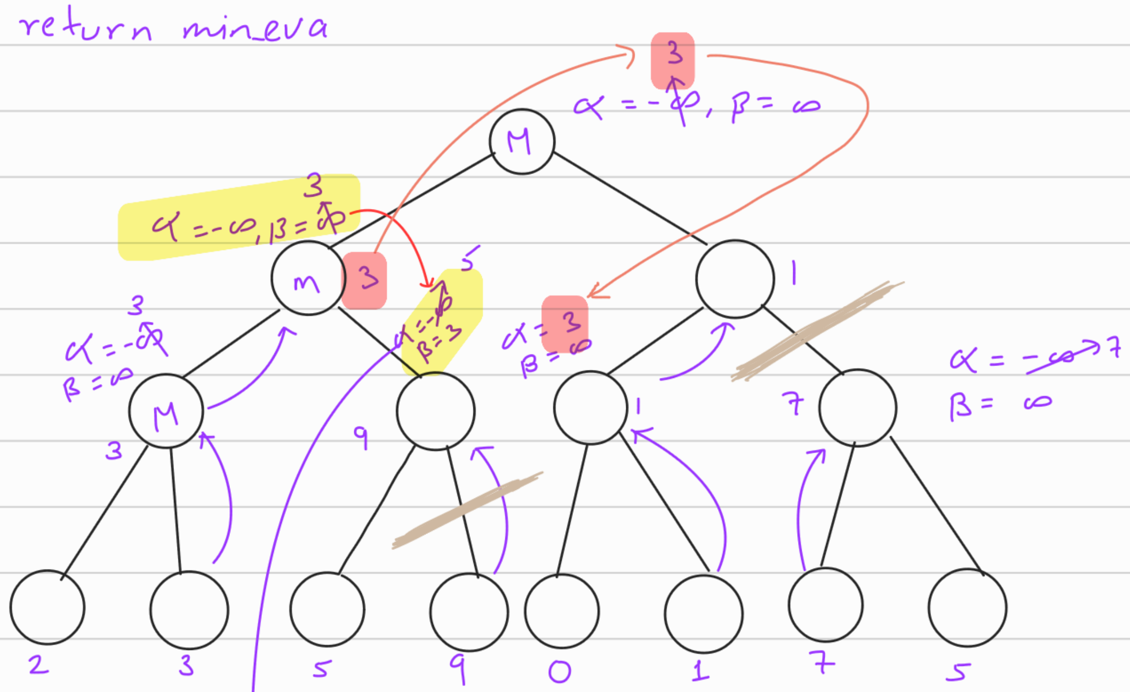
α-β algorithm (To incorporate pruning of node) →

- α - will be calculated only for max player & β for min player
- Modified min-max algo.
- DFS approch

function minmax (node, depth, Maximum, α, β)
    if (depth = 0 or node is terminal node)
        return value of node

if Maxplayer   initial $\alpha = -\infty$
    max_eva = $-\infty$ $\beta = +\infty$
    for (each succeor of node)
        eva = minmax (succeor, depth+1, false, α, β)
        max_eva = max (max_eva, eva), α = max (α, max_eva);
        if ($\beta \le \alpha$)
            break;
    return max_eva

else
    min_eva = $+\infty$
    for (each succesor of node)
        eva = minmax(child, depth+1, true, α, β)
        max_eva = min(min_eva, eva), β = min(β, min_eva)
        if ($\beta \le \alpha$)
            break;
    return mineva



∵ 5 ≥ 3
∴ we dont have to traverse this node

- Two type of ordering

Ex : 14, 5, 2

Worst → Normal move first then best move TC → $O(b^d)$

Best → Best move first then normal move TC → $O(b^{d/2})$

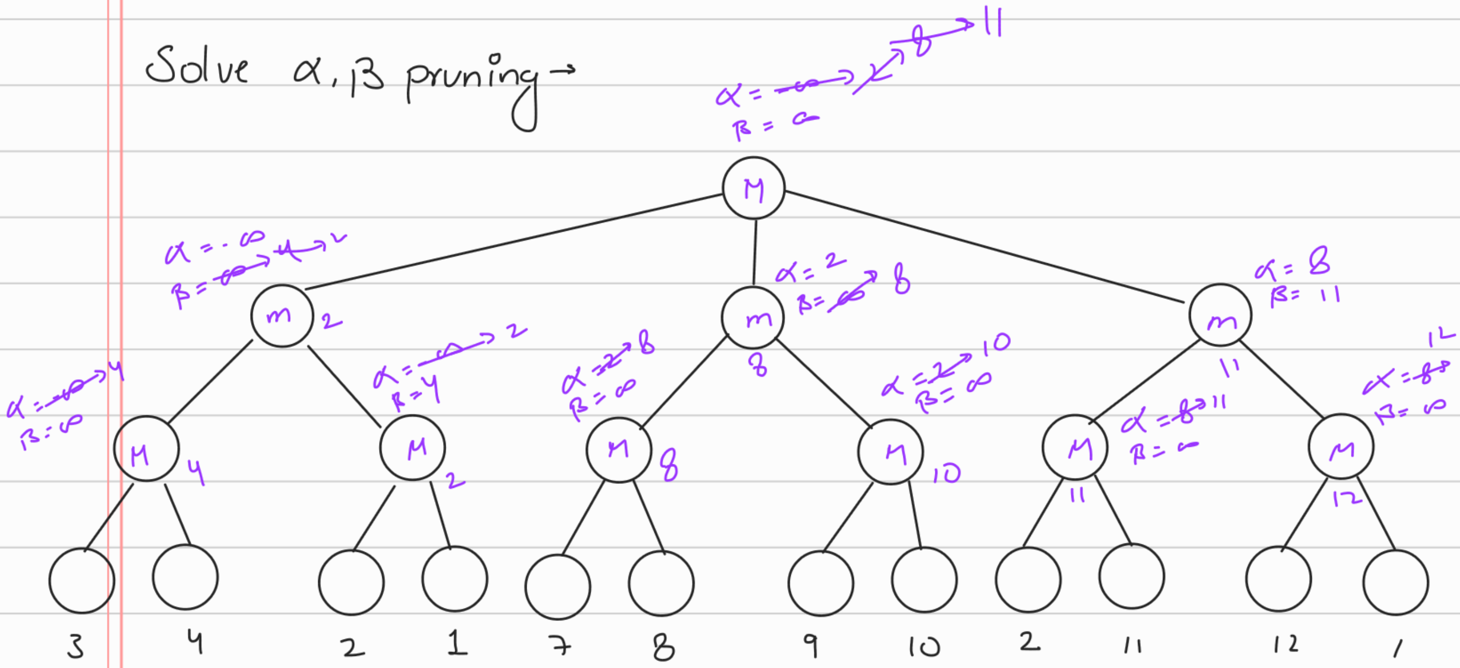
Ex 2, 4, 6

- α → Max value along path of max player

β → Min value along path of min player

- Each node of search tree will be passed on to top bottom.
- breaking condⁿ $\beta \le \alpha$

Solve α, β pruning →



Constraints satisfaction processing (CSP) →

→ Set of variable $X = \{x_1, x_2, \ldots x_n\}$

→ Set of Domain $D = \{d_1, d_2, \ldots d_n\}$

→ Constraint $C_i = \{S_i, R_i\}$ $\forall$ $C = C_i$ $\forall$ $i \in [1, n]$

scope Relation

$$S_i \subseteq \{x_{i1}, x_{i2}, \ldots, x_{in}\}$$
$$R_i \subseteq \{d_{i1}, d_{i2}, \ldots, d_{i3}\}$$

- Solⁿ is assignement of value from domain to variable such tha

All constraints satisfies.

Ex 4- Queen problem

Set of var: {Q_1, Q_2, Q_3, Q_4}

Block (4x4) 16 are domain

Constain: row, col, dig block

Ex Graph coloring problem

Constrain: No neighbour country have same color

Ex Sudoku problem.

Ex SAT problem

bool exp with some var. & exp. & we have to assign 0 & 1

In such a way that whole outcome of exp is true or false

Ex Cript arithematic problem →

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

Can we assign value from 0-9 such that no two var. has same value

1 ∴ T=2 G=8 ~~G=9~~ ⇒ U=1

$$\begin{array}{r} c_4\,c_3\,c_2\,c_1\ \ \\ S\,E\,N\,D \\ +\ M\,O\,R\,E \\ \hline M\,O\,N\,E\,Y \end{array}$$

2 ways →

→ Generate & Test

→ Backtracking

S+M ≥ 10

S ≥ 9

S	E	N	D	M	O	Y
9	=N			1	0	

(If C_3=0)

wrong ∴ consider carry

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

Chk for all

C_1: 1 C_2: 1 C_3: 0 C_4: 1

$$
\begin{array}{cccccc}
 & c_4 & c_3 & c_2 & c_1 & \\
 & B & A & S & E \\
+ & B & A & L & L \\
\hline
G & A & M & E & S
\end{array}
$$

B : 7

A : 4

S : 8

E : 3

c_1 :

c_3 : 0

L : 5

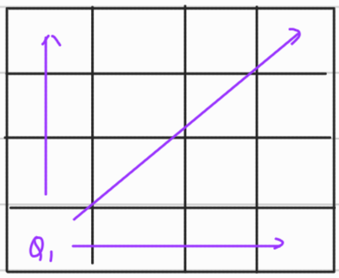
G : 1

M : 9

c_2 : 1

c_4 : 1

Matching graph →



R_{14}

$R_{12} = \{(1,3), (1,4), (2,4), (3,1), (4,1), (4,2)\}$

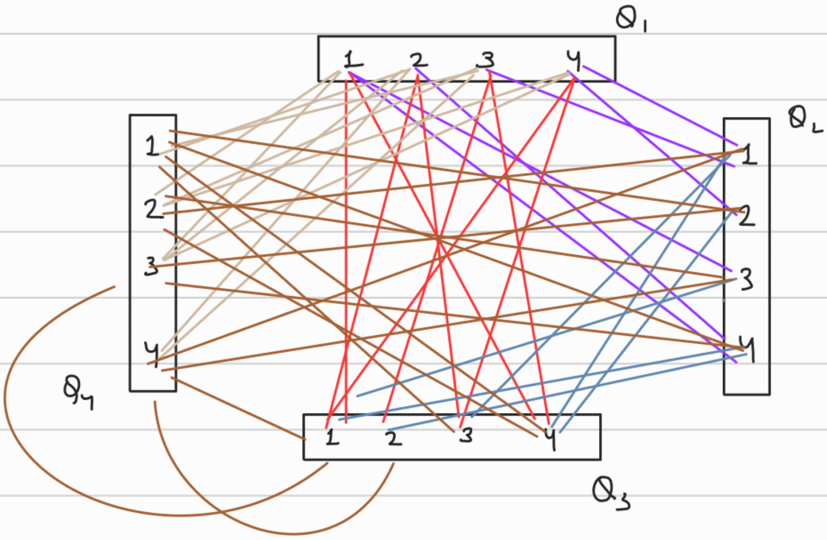
$R_{13} = \{(1,2), (1,4), (2,1), (2,3), (3,2), (3,4), (4,1), (4,3)\}$

$R_{14} = \{(4,2), (4,3), (1,2), (1,3), (2,1), (2,3), (2,4), (3,1) (3,2) (3,4)\}$

$R_{23} = \{(1,3), (1,4), (2,4), (3,1), (4,1), (4,2)\}$

$R_{24} = \{(1,2), (1,4), (2,1), (2,3), (3,2), (3,4), (4,1), (4,3)\}$

$R_{34} = \{(1,3), (1,4), (2,4), (3,1), (4,1), (4,2)\}$



Q. Consider the following class room scheduling problem there are 4 classes C_1, C_2, C_3, C_4 & 3 class rooms r_1, r_2 & r_3 the following tables shows the class schedule

Class	Time
C1	8 - 10:30 am
C2	9 - 11:30 am
C3	10 - 12:30 am
C4	11 - 1:30 an

In addition to this ∃ following restriction

- Each class must use one of 3 rooms r_1, r_2, r_3
- r_3 is too small for C_3
- r_2 & r_3 are too small for C_4

(i) Show initial values for C_1, C_2, C_3, C_4

(ii) Express given problem in term of constraint satisfaction problem

(iii) Find the solution using arc consistensy.

$C_1 = \{R_1, R_2, R_3\}$

$C_2 = \{R_1, R_2, R_3\}$

$C_3 = \{R_1, R_2\}$

$C_4 = \{R_1\}$

$C_1 \neq C_2$

$C_2 \neq C_3$

$C_3 \neq C_4$

$C_3 \neq R_3$

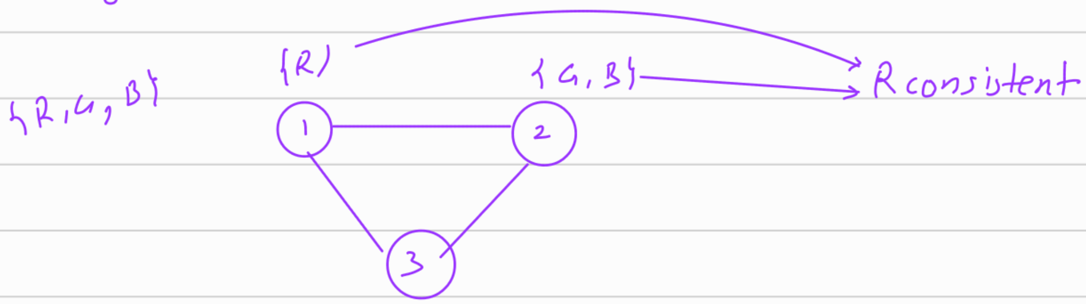
$C_4 \neq R_2$

$C_4 \neq R_3$

$C_1 \neq C_3$

Set of variables $= \{C_1, C_2, C_3, C_4\}$

Set of domains $= \{R_1, R_2, R_3\}$



Making consistent

$C_1 = \{R_1, \cancel{R_2}, \cancel{R_3}\}$

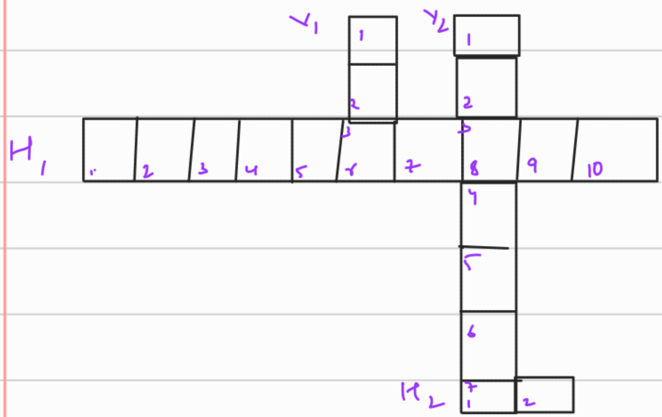
$C_2 = \{\cancel{R_1}, \cancel{R_2}, R_3\}$

$C_3 = \{\cancel{R_1}, R_2\}$

$C_4 = \{R_1\}$

Q. Consider a varient of crossword puzzle problem in this varient we assume that we have a set of words $w_1, w_2, -- w_n$ and a crossword puzzle grid our goal is to fill the crossword grid with the words such that letters of intersecting words matches. Identify the variables to which values are to be assigned specify the domain from which variable can take there values define the constrainst that must be hold between variables and find the solution.

THE, UNIVERSITY, OF, CHICAGO ⟶ Domain



Constraint →

$H_1(10)$	$H_1(5) = V_1(3)$
$H_2(2)$	$H_1(8) = V_2(3)$
$V_1(3)$	$H_2(1) = V_2(7)$
$V_2(7)$	

Q.

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
R_1									
R_2									
R_3									
R_4									
R_5									
R_6									
R_7									
R_8									
R_9									

All diff $= \{(R_1, C_1), (R_1, C_2), (R_1, C_3)$
$(R_2, C_1), (R_2, C_2), (R_2, C_3)$
$(R_3, C_1), (R_3, C_2), (R_3, C_3)\}$

All diff $= \{(R_1, C_1), (R_1, C_2), (R_1, C_3), (R_1, C_4) \ldots \ldots (R_1, C_9)\}$

$\vdots$

All diff $= \{(R_9, C_1) \ldots \ldots (R_9, C_9)\}$

All diff $= \{(C_1, R_1), (C_1, R_2) \ldots \ldots (C_1, R_9)\}$

$\vdots$

$\{(C_9, R_1), (C_9, R_2) \ldots \ldots (C_9, R_9)\}$

Cryptarithmetic →

```
        2
      A P D
    ×   A D
    -------
    R P A D
  D D C D
  ---------
  D P C E D
```

$a \cdot b = kb$	$a \cdot b = ka$
1.5	2.6
3.5	4.6
7.5	8.6
9.5	

even + even = even

odd + even = odd

odd + odd = even

D = 5, P = 9, A = 7, R = 3, E = 2, C = 6

```
        R O Z
      * M U G
  ------------
      T A U A
    T O Y Z
  T U G A
  ------------
  R A T T T A
```

Z = 5, A = 0, G = 6, U = 7

T = 2, R = 3, O = 4, Y = 1

M = 8

Knowledge representation →

- Understanding of subject area
- Machine can acquire this knowledge

Concepts & facts

Relationship b/w concepts — Knowledge — properties of concepts

Ability to combine this knowledge to solve problem

Data V/s Info V/s Knowledge →

Data	Info	Knowledge
Premettive facts	Processed data	Processed Info

- Acqusition gathering
- Storage
- Retrival
- Reasoning

Categories of knowledge →

1. Procedural knowledge → Rules, Strategies & procedures
   → Also called imperitive knowledge
   → How to do things.
   → directly applied to task

2. Declaritive → Concepts, objects, facts
   → knowing about something
   → More generalized than procedure

3. Meta knowledge → Knowledge about knowledge.
   → Useful in reveling pattern (Ex Catalog)

4. Heuristic knowledge → Educated guess
   → i.e. based on some fact or logic

5. Structural knowledge → Rule set, Concept & relationship
   ↓
   kind of, part of, is a

6. Uncertain knowledge → Probabilistic guess.

7. Common sense knowledge → Default propozitions

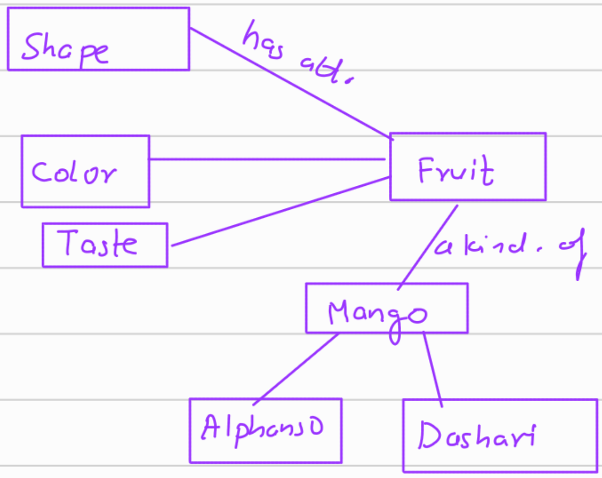
8. Ontological knowledge → Relationship b/w constraints
   → It overlap with procedural & Declaritive knowledge

Representation →

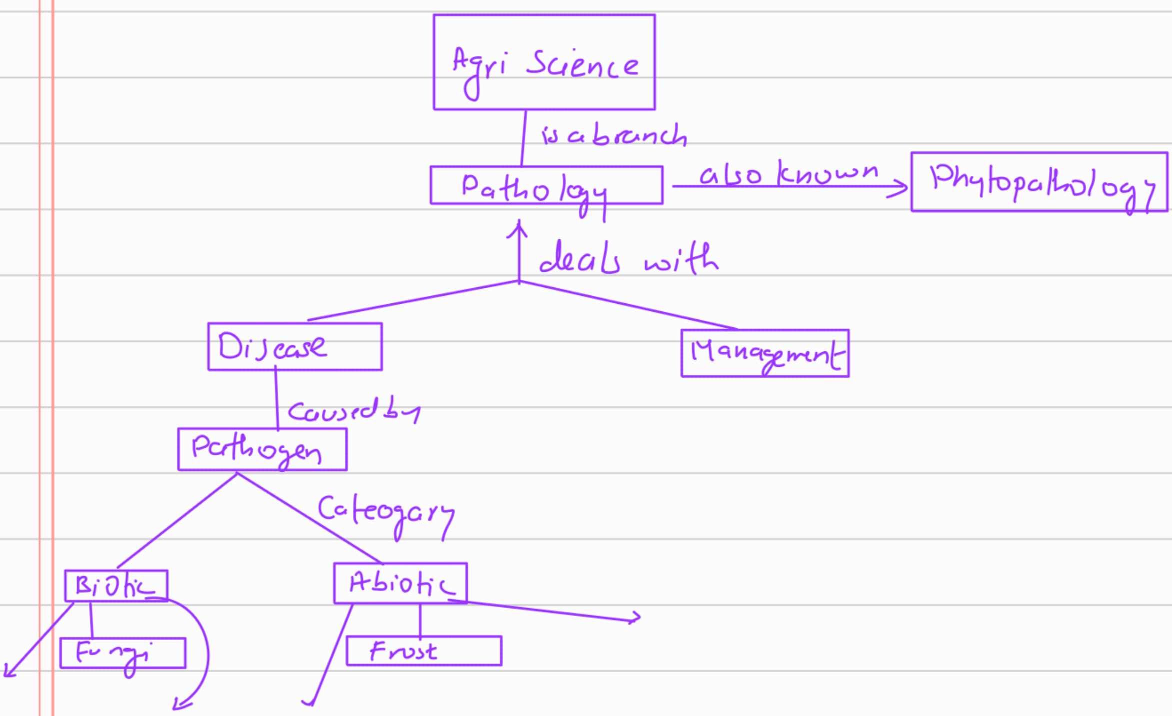
- Object Attribute value (triplet) →

$\text{Object} \xrightarrow{\text{Attribute}} \text{Value}$

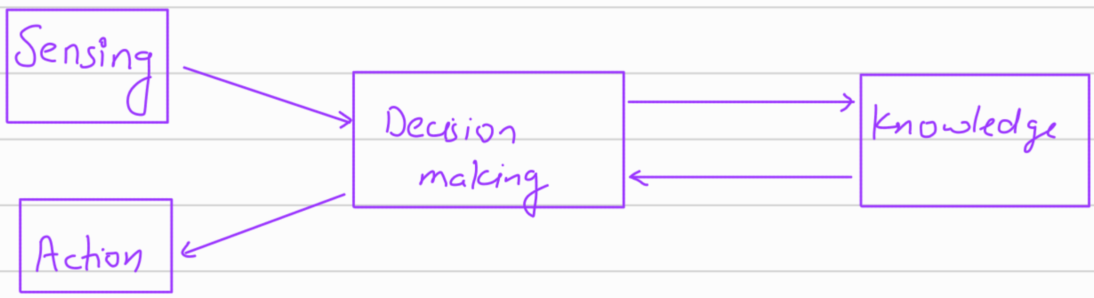
- Sementic N/w^k or concept map
  - ↳ Visual representation of concepts
  - ↳ Rectangles and Arc/Line



Q A pathalogy also known as phytopathology is a branch of agricultur al science it deals with plant desease and then management plant desease are caused by pathogen, pathogen may be biotic or abiotic, biotic pathogen includes funi, bacteria & prematods abiotic pathogen include frost, air pollutants and toxicants draw sementic network.



- Knowledge & intelligence →



- We need a knowledge to interect with machine
- Based on data it must enfrence some concept or knowledge

Q. In 5 houses each with different color live 5 person of different nationalities each of whom prefers a diff. brand of candy a diff. drink and & diff pet and with following constraint, the english man lives in the red house, the sponion owns a dog the norwanion lives in the first house on the left, the green house is immediately to the right of ivory house the man who eats harshey bars lives in the house next to the man with a fox, kitkats are eatens in the yellow house the norwanian lives next to the blue house the smarties eatens owns a snail. the sneaker eater drinks orange juice the eukranian drinks tea, the japanese eats milky way. kit kats are eaten in a house next to the house where horse is kept coffe is drunk in green house, milk is drunk in middle house

(i) Where does the zebra lives & in which house they drink water
(ii) Represent problem as constrainst satisfaction problem.

Colors = {Red, ivary, green, yellow, blue}
Nationality = {English, Spanish, norwanion, eukranion, japanese}
pet = {Dog, fox, snail, horse, zebra}
snacks = {harsheys, kitkats, smarties, srealcar, milky way}
Juices = {Orange Juice, tea, coffe, milk, water}

Nationality	Color	Pets	Snacks	Juice
Spanish	~~R~~ I G Y B	D ~~F~~ ~~S~~ ~~H~~ ~~Z~~	H K S Sn ~~M~~	O ~~T~~ C M W
English	R ~~I~~ ~~G~~ ~~Y~~ ~~B~~	~~D~~ F S H Z	H K S Sn ~~M~~	O ~~T~~ C M W
Norwanion	~~R~~ I G Y B	~~D~~ F S H Z	H K S Sn ~~M~~	O ~~T~~ C M W
Eukranion	~~R~~ I G Y B	~~D~~ F S H Z	H K S Sn ~~M~~	~~O~~ T ~~C~~ ~~M~~ ~~W~~
Japnese	~~R~~ I G Y B	~~D~~ F S H Z	~~H~~ ~~K~~ ~~S~~ ~~Sn~~ M	O ~~T~~ C M W

H1 H2 H3 H4 H5

Language to represent knowledge → Syntax → How to write a sentence
→ Semantic → Meaning of sentence

1. Preposition logic → facts   T/F/unknown
2. First order logic → facts objects relation   "
3. Temporal logic → time   "
4. Probability theory → prob. of being true or false   (0 → 1)
5. Fuzzy logic → How true/false   Degree of believe (0-1)

Preposition logic →
- Can be represented using grammer
- sentence → Atom/complex sentence
- Atom → true/false/Another preposition
- Complex Sentence → sentence/ sentence connective Sentence / sentenses
- connective → $\wedge / \vee / \rightarrow / \leftrightarrow$

let P & Q is proposition
1. $-P$ → is true if P is false in model
2. $P \wedge Q$ → is true if $P \in$ True & $Q \in$ true
3. $P \vee Q$ → is true if $P \in$ true or $Q \in$ true
4. $P \rightarrow Q$ → is true if P is true & Q is false
5. $P \leftrightarrow Q$ → is true if P & Q are true or P & Q both false

$(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$

$(\alpha \vee \beta) \equiv (\beta \vee \alpha)$

$(\alpha \wedge \beta) \wedge \gamma \equiv \alpha \wedge (\beta \wedge \gamma)$

$(\alpha \vee \beta) \vee \gamma \equiv \alpha \vee (\beta \vee \gamma)$

$-(-\alpha) \equiv \alpha$

$\alpha \rightarrow \beta \equiv (-\beta) \rightarrow (-\alpha)$ Contrapositive

$\alpha \rightarrow \beta \equiv (-\alpha \vee \beta)$

$\alpha \leftrightarrow \beta \equiv (\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$

$-(\alpha \wedge \beta) \equiv (-\alpha) \vee (-\beta)$ Demorgans law

→ Entailment $\alpha \models \beta$ ie α logically follows β

α entails β

→ Satisfibility
- Tautology (All true)
- fallacy or Contradiction (All false)

Wumpus World problem →

	1	2	3	4
4	stanch		Breeze	Pit
3	Wumpus	Breeze stench Gold	Pit	Breeze
2	stanch		Breeze	
1		Breeze	pit	Breeze

- represented in form of PEAS → Performance, Environment, Actuator, Sensors
- If you enter pit or wampus games end
- To win reach blak with gold

P Up, down, left, right
E: ?
A: ?
S: Stench, Breeze, Gold

[1,1] → ok no breeze, stinch
[1,2] → Breeze → no conclusion of having pit ∴ move back ∴ at [1,1]
[2,1] → Stench, there is wampus around
[2,2] → Ok
[2,3] → Gold

P_{xy} is preposition pit is (x,y)
W_{xy} is —"— wumpus in (x,y)
B_{xy} is —"— Breeze in (x,y)
S_{xy} is —"— Stench in (x,y)

derive the preposition for $-P_{21}$

$B_{12} \to P_{22} \vee P_{13} \vee P_{11}$

$-P_{11}$, $-B_{11}$, $-S_{11}$, $-B_{21}$, $-W_{21}$, $-P_{21}$

B_{12} , S_{21} , $-W_{21}$, $-W_{12}$,

$B_{11} \leftrightarrow P_{12} \vee P_{21}$

$B_{11} \to (P_{12} \vee P_{21}) \wedge (P_{12} \vee P_{21}) \to B_{11}$

By applying and elemination

$P_{12} \vee P_{21} \to B_{11} \quad \Rightarrow \quad B_{11} \to P_{12} \vee P_{21}$

$-B_{11} \to -(P_{12} \vee P_{21})$

then there is a law model ponens

$$\frac{\alpha \to \beta, \alpha}{\beta}$$

$\neg(P_{12} \vee P_{21}) = \neg P_{12} \wedge \neg P_{21}$

Deduce the preposition P_{31}

Infrence rules for preposition logic

premise	conclusion	
$\alpha \to \beta, \alpha$	β	Modus Ponens
$\alpha_1 \wedge \alpha_2 \cdots \wedge \alpha_n$	α	And reduction
$\alpha_1, \alpha_2, \cdots \alpha_n$	$\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \cdots \wedge \alpha_n$	And introduction
$\neg(\neg \alpha)$	α	double negation
$\alpha \to \beta, \neg \beta$	α	Unit resolution
$\alpha \vee \beta \to \beta \vee \gamma$	$\alpha \vee \gamma$	Resolution
$-\alpha \to \beta, \beta \to \gamma$	$-\alpha \to \gamma$	Resolution

CNF → Conjugate normal form $(P_1\ P_2\ P_3)$

$$(P_1 \vee P_2) \wedge (P_2 \vee P_3)$$

DNF → Disjunctive normal form

$$(P_1 \wedge P_2) \vee (P_2 \wedge P_3)$$

$$B_{11} \leftrightarrow P_{12} \vee P_{21}$$

$$(B_{11} \longrightarrow P_{12} \vee P_{21}) \wedge (P_{12} \vee P_{21} \longrightarrow B_{11})$$

$$(\neg B_{11} \vee (P_{12} \vee P_{21})) \wedge (\neg (P_{12} \vee P_{21}) \vee B_{11})$$

$$(\neg B_{11} \vee (P_{12} \vee P_{21})) \wedge (((\neg P_{12}) \wedge (\neg P_{21})) \vee B_{11})$$

Q. When asked about the ages of her 3 children ms. baker says that alice is her youngest child if bill his not her youngest child & Alice is not her youngest child if carl is not her youngest child write down the knowledge base describe in puzzle & show that bill is her youngest child?

let $A \rightarrow$ Alice is youngest
$B \rightarrow$ Bill ——"——
$C \rightarrow$ Carl is ——"——

$\Rightarrow A \vee B \vee C$ (One of them is youngest)
$\neg A \vee \neg B$, $\neg A \vee \neg C$, $\neg B \vee \neg C$ (Only 1 among them is youngest)

$\neg B \longrightarrow A \Rightarrow B \vee A$

$\neg C \longrightarrow \neg A \Rightarrow C \vee (\neg A)$

Our assumption is bill is not youngest ($\neg B$)

$\neg B \wedge (\neg B \rightarrow A) \Rightarrow A$

$A \perp -A$

$\neg B \wedge (-A \vee \neg C) \Rightarrow \neg A$

$\because$ Contradiction

B is youngest

Q. A boy & girl is talking I am a boy said a child with black hare, i am a girl said a child with white hair, at least one of them is lying write down the knowledge base & show that both of them are lying

$W_t \rightarrow$ White haired boy who is talking truth
$W_b \rightarrow$ White hair is a boy
$B_t \rightarrow$ Black hair boy who is talking truth
$B_b \rightarrow$ Black hair is boy

1. $B_b \vee W_b$
2. $\neg B_b \vee \neg W_b$
3. $B_t \longrightarrow B_b$
4. $\neg B_t \longrightarrow \neg B_b$
5. $W_t \longrightarrow W_b$
6. $\neg W_t \longrightarrow W_b$
7. $\neg B_t \vee \neg W_t$
8. $B_t \vee W_t$

3 & 8 ⇒ 9
5 & 9 ⇒ 10
1 & 10 ⇒ 11
2 & 10 ⇒ 12
4 & 11 ⇒ 13
6 & 12 ⇒ 14

7 & 13 ⇒ 15
14 & 15 ⇒ Contradiction

Forward reasoning & Chaining →

Ex→

$P \to Q$

$L \wedge M \to P$

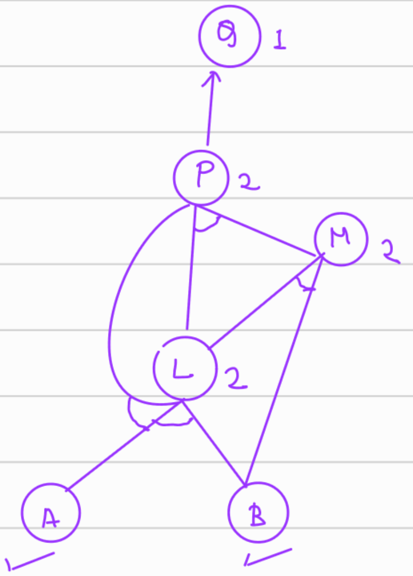
$B \wedge L \to M$

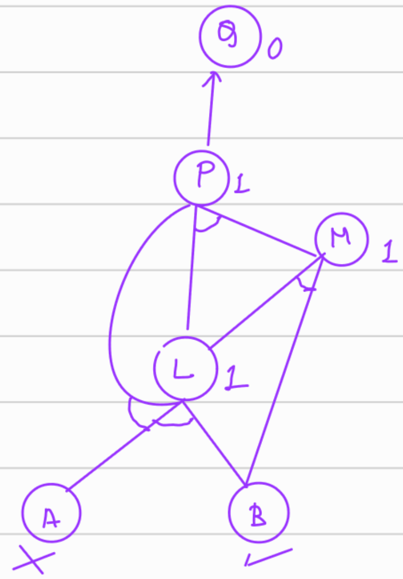
$A \wedge P \to L$

$A \wedge B \to L$

A

B





Ex-

$A \wedge C \to F$

$A \wedge E \to G$

$B \to E$

$G \to D$

Conclude $A \wedge B \longrightarrow D$

∴ In beg. we have A & B

dB : A, B

↓ 3

dB = A, B, E

↓ 2

dB = A, B, E, G

↓ 4

dB = A, B, E, G, D

Backward Chaining →

$F \wedge B \rightarrow Z$

$C \wedge D \rightarrow F$

$A \rightarrow D$

We have conclusion & we need to find knowledge base

Clonclude z

DB

A , B E , C

this is true

↓

A , B , E , C , D

↓

A , B , E , C , D , F

↓

A , B , E , C , D , F , Z

Predicate logic / First order logic →

- Objects , relations & Functions

Ex Studies (A , MANIT)

predicate or relation

Objects

- We also use quantifiers in it ie $\forall$ (for all) $\exists$ (their exist)
- It also has constants , variables
- Connections $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\neg$

determines — terminal 1, terminal 2 → sentence

Ex Brother (x, y)

predicate — variable

Sentence $\longrightarrow$ terminal 1 / terminal 2

terminal $\longrightarrow$ variable

variable $\longrightarrow$ x/y

Ex $\boxed{\text{Predicate}\{\text{function}_1(\text{function}(x,y)), \text{function}_2(\text{function}(y,z))\}}$

sentence

variable

Complex sentence

Complex sentence

sentence

Ex $\underset{\text{(Pred)}}{\text{Sibling}}(\underset{v1}{\text{Ram}}, \underset{v2}{\text{Shyam}}) \Rightarrow \underset{\text{(Pred)}}{\text{Sibling}}(\text{Shyam}, \text{Ram})$

Ex. Everyone who studies at MANIT is Smart — predicate logic ??

Quantifier

$\forall_x$ Studies $(x, MANIT) \longrightarrow$ Smart(x) ✓

$\forall_x$ Studies $(x, MANIT) \wedge$ Smart(x) ✗

Ex Someone studies at Manit is smart

$\exists_x$ Studies $(x, Manit) \longrightarrow$ Smart(x) ✗

$\exists_x$ Studies $(x, Manit) \wedge$ Smart(x) ✓

Note ⇒ $\forall x \forall y \equiv \forall y \forall x$

$\exists x \exists y \equiv \exists y \exists x$

$\exists y \forall x \not\equiv \forall x \exists y$

Ex. There is a person who loves everyone the world

$\exists_x \forall_y \; loves(x, y)$

Ex. Everyone has a mother

$\forall_x \exists_y \; mother(x, y)$

$\Rightarrow$ $\forall_x \; loves(x, icecream) \equiv \neg\exists_x \; \neg loves(x, icecream)$

Ex. brothers are sibling

$\forall_x \forall_y \; Brother(x, y) \longrightarrow Sibling(x, y)$

$\Rightarrow$ $\forall_{x,y} \; Sibling(x, y) \longleftrightarrow Sibling(y, x)$

Ex. - One's mother is one's female parent
- A first cousin is a child of a parents sibling.

$\forall_{x,y} \; (mother(x, y) \longleftrightarrow female(x) \wedge Parent(x, y))$

Q. The law says that it is a crime for an american to sell weapons to the hostile nations, the country nano an enemy of america has some missiles & all of its missiles were sold to it by Kernal west who is an american proof that kernal west is a criminal

American (West)

enemy (nano)

sell (wepons, enemy, west) $\longrightarrow$ crime

crime (west) $\longrightarrow$ criminal

weapon (missile)