Date
Page

Syllabus.

UNIT- 1

Introduction

Introduction to Artificial Intelligence, History of Artificial Intelligence, State of the Art, Risks and Benifits of A.I, Intellegent Agents, Agents & Environment, good Behaivour: Concept of Rationality, Nature of Environments, Structure of Agents.

UNIT- 2

Problem Solving.

Solving Problems by Searching: Problem-solving Agents, Example problems, Search algorithms, Uniformed Search Strategies, Informed (Heuristic) Search Strategies, Heuristic functions, search in Complex Environment, Local Search and Optimization Problems.

UNIT- 3

Date
Page

UNIT- 5

5Ri Adversarial Search & Games.

Game Theory, Optimal Decisions in games, Heuristic Alpha- Beta Tree Search, Monte Carlo Tree Search, Stochastic Games, Partially Observable Games, Limitations of Game Search Algorithms, Constraint Satisfaction Problems (CSP), Constraint Propagation: Inference in CSP's, Backtracking Search for CSP's.

Sudoku } knapsack }

UNIT- 4.

Knowlegde

Logical Agents, Knowledge-Based agents, The Wumpus World Logic, Propositional logic: A very simple Logic, Propositional Theorem Proving, Effective Propositional Model checking, Agents Based on Propositional Logic, First - Order Logic, Representation Revised, Syntax and Semantics of first order Logic, Using first order Logic, Knowledge Engineering in first order logic.

wumpus world problems. 6 M
Satisfiability problem. - 3M
ANN 5M
Decision Tree

Prolog.

Date
Page

UNIT - 5

Internet shopping world.
Backward Chaining, Prolog.
forward Chaining
Resolution

Reasoning.

Inference in First-Order Logic, Propositional vs. first Order Inference, Unification and first-order inference, forward Chaining, Backward Chaining, Resolution, Knowledge Representn, ontological Engineering, Categories and objects, Events, Mental objects and Modal Logic, Reasoning Systems for Categories & Objects, Reasoning with Default Information.

UNIT - 6

Planning.

Automated Planning, Classical Planning, Algorithm for classical planning, Heuristic for Planning, Heirachical Planning & acting in Non-deterministic Domains,

Time, Schedules, and Resources, Analysis of planning Approaches, Limits of AI, Ethics of AI, future of A.I, A.I Components, A.I Architectures.

Strips, block world, S.W, H.W Components

24/7 Entropy $= \sum_{i=0}^{n-1} - p_i \log_2 p_i$

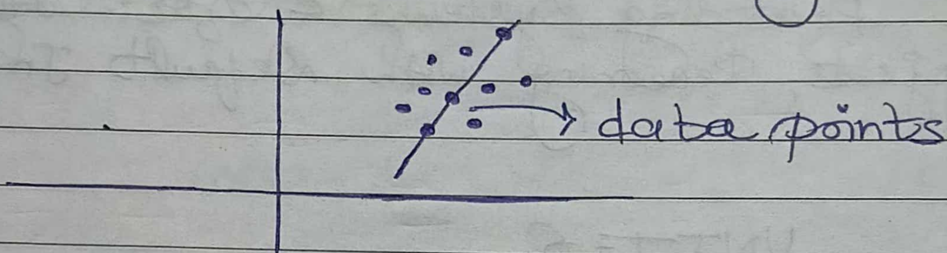
25/7 * Artificial Neural Network.

$$\boxed{\Sigma = mx + c}$$

Canteran Geometry.

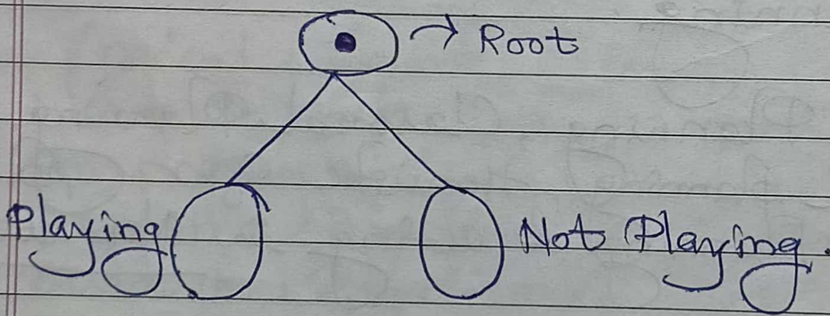
* Root Mean Square Error.
* Line Regression $\longrightarrow$ $\boxed{y = mx + c}$ $\longrightarrow$ function

$\boxed{y = mx + c + \beta}$ $\longrightarrow$ Tuning parameters p a β



26/7

* Data Set, Concept Drift, Entropy



Software* knowlege, knowledge Representation, Machine learning, Natural language Processing, knowkdge Base

further it needs:

① Computer Vision
② Robotics.

} Hardware

Meta → Data - about data.

Date
Page

① Thinking humanly : Congitive Modelling approach.

- get inside the human mind

* knowledge level Through Test, assignment

- → factual, Conceptual, Procedural, Meta-Cognitive knowledge.

③ ways to do : introspection, psychological experiments, brain imaging.

* Cognitive Science : Related to thinking

- Philosophy : ① Can formal rules to draw valid conclusions ?
② How knowledge lead to action ?
③ Where knowledge come from ?
④ How does the mind arise from the physical brain ?

- Mathematics : ① What are the formal rules to draw with conclusions ?
② What can be computed ?
③ How do we reason with uncertain information?

- Economics : ① How should we make decisions as to maximize payoff ?

(2) How should we do this when others may not go along ?

(3) How should we do this when others

- Psycology : How do humans and animals think and act ?

- Computer Engineering :

(1) How can we build a efficient Computer?

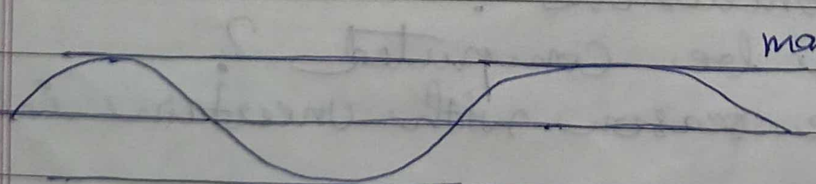
- Intelligent Agents : Agents and environment

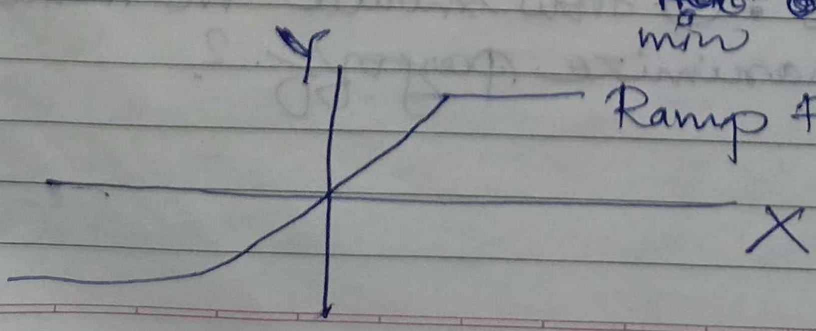
Activation functions:

Liner function : $F_{AN}(net-\theta) = \lambda(net-\theta)$

(2) Step function : $F_{AN}(net-\theta) = \begin{cases} \delta_1, & \text{if } net \geq 0 \\ \delta_2, & \text{if } net \leq 0 \end{cases}$

Ramp function : $F_{AN}(net-\theta) = \begin{cases} \delta & net-\theta \geq \epsilon \\ net-\theta & -\epsilon < net-\theta < \epsilon \\ -\delta & net-\theta \leq \epsilon \end{cases}$

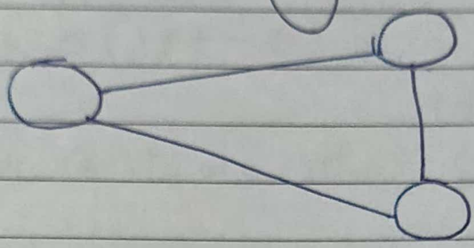




Ramp Function

/8/22 A·I Lab.

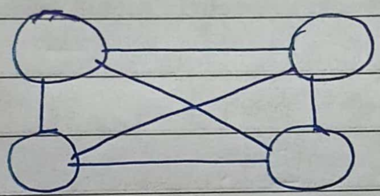
01. # Graph Colouring Problem

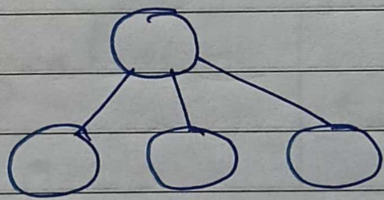


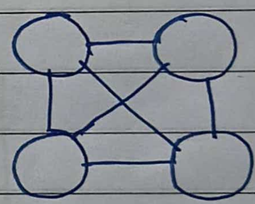
Graph G(V,E)

E - finite edges
V - finite vertices.

Asked to colour this graph.
no two nodes have same color.
with minimum







ad j →

i ↓ $[(x_4)]$ 6×6.

$\{x_1, x_2, x_3, x_4\}$

for (___)
{

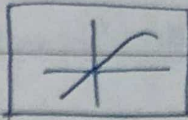
for (___)
{

if (ad [i] [j] == 1)

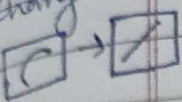
color a[i] [j] = x [k]

4/8/22

Sigmoid Function: ☆



as λ changes curve changes.



$$F_{AN}(net-\theta) = \frac{1}{1+e^{-\lambda(net-\theta)}}$$

When λ controls the steepness generally is 1

Hyperbolic Tangent Function:

$$F_{AN}(net-\theta) = \frac{e^{\lambda(net-\theta)} - e^{-\lambda(net-\theta)}}{e^{\lambda(net-\theta)} + e^{-\lambda(net-\theta)}}$$

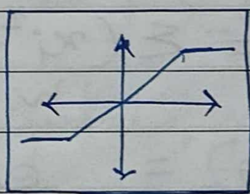
Gaussian function:

$$F_{AN}(net-\theta) = e^{-\frac{\tan\theta^3}{\theta^{\theta}}}$$

8/8/22 Monday.

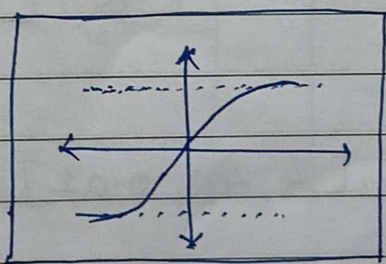
Hyperbolic Tangent function:

• Ramp function:



Apply tanh function becomes hyperbolic funct

tanh



hyperbolic function

$m_1 = e^{\lambda x}$ & $e^{-\lambda x} = m_2$

$$y = \frac{m_1 - m_2}{m_1 + m_2}$$

$\sqrt{S.D} = \sigma$

$Var = E(x^2) - [E(x)^2]^2$

* Steepness (function)

Gaussian function: (for real numbers)

☆ Iris Data: flowers & their classification
3 parts of flower

{mean, median, mode, range, variance, deviation}

$y = mx + c$ independent
↓
dependent

y	x
9	5
12	4
6	8
7	3

$$mean(x) = \frac{5+4+8+3}{4} = \mu.$$

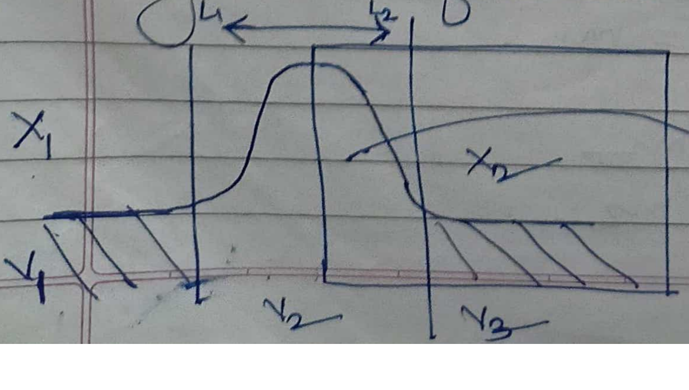
$$\sigma^2 = \frac{\sum (x_i^2 - \mu^2)}{n}$$

$f(net - \theta) = e^{-\frac{(net-\theta)^2}{\sigma^2}}$

$$f(x) = e^{\frac{-x x^2}{\sigma^2}} \qquad x = [x_1, x_2, x_3 \cdots x_n]$$

- Import math
x = np.arrange[4, 40, 0.01]

Gaussian function:



X_1 & X_2 are Xtreme Values.

Concentration of attribute of Values.

17/8/2022

* Intelligent Agents :

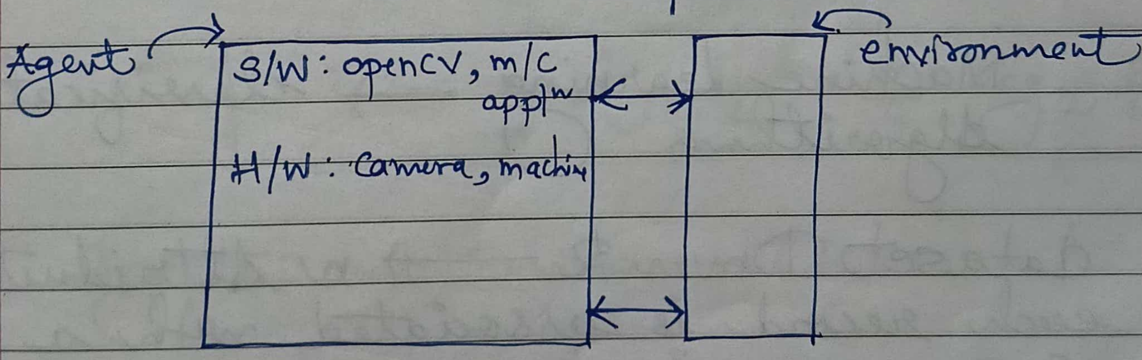
· Agents and Environments :

Example :
Hospital
← Assitive System
← Doctors to take Decisions.

medical history / reports of patients

{Camera, machine}
Agents : {S/W, H/W} → {M/C Learning Component, Predicting disease, OpenCV}

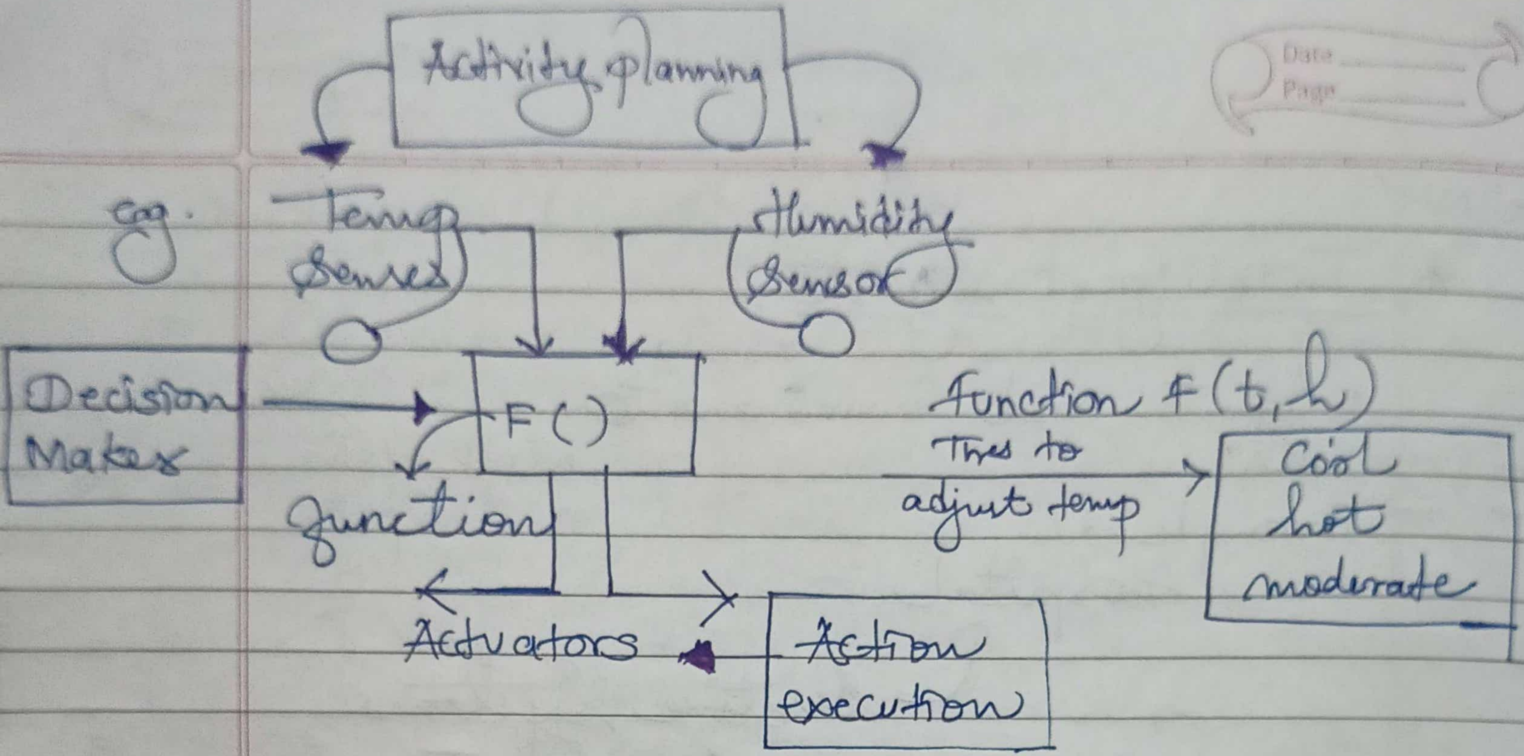
Environment : {Hospital, patient, visitor}



(Sensors) Group 1 : Trying to understand environment.

(Activators) Group 2 : Trying to take the decisions based on what system understood from group 1.

e.g.

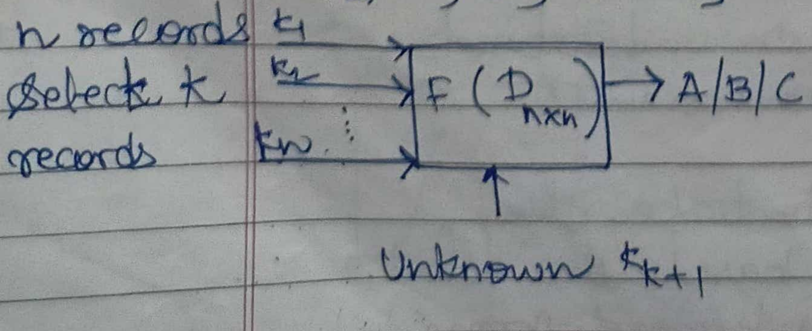


Sensors (I/P) → System
↓
main function (F())
making decisions
↓
Actuators (O/P)

* Realistic Example:

Machine Learning Algorithms → classification

dataset D n×m R → n Attributes
each record is associated with a class
{A, B, C ....}



function F learned from k to kn records
ready to classify the record

Date
Page

Function F(D) ⟵ mathematical operations.

Data set D

Temp	Humidity	Wind	Play
L	L	L	Y
H	L	L	N

10 records (6, 4)

Size of Data set 10 x 4

dataset first → potential

$$\boxed{\text{Entropy} = -\frac{6}{10}\log_2\frac{6}{10} - \frac{4}{10}\log_2\frac{4}{10}}$$

Temperature = low, high, moderate.

4 < 1, 3

$$\boxed{E = -\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{10}}$$

18/8/2022
Thursday.

Agents and Environment.

- Address different types of agents.

* Vacuum Cleaner ⟷

1. Camera
images $\xrightarrow[\text{dust}]{\text{understood}}$ ON/OFF suct section.

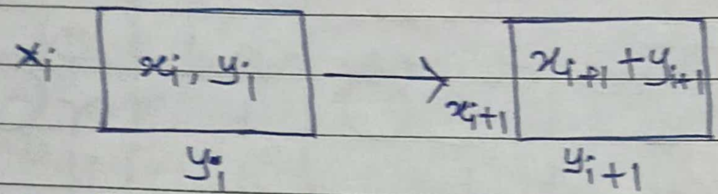
example of ~~reflex~~ simple reflex Agent system.

* Some agents require previous history to do analysis of future and some doesn't.

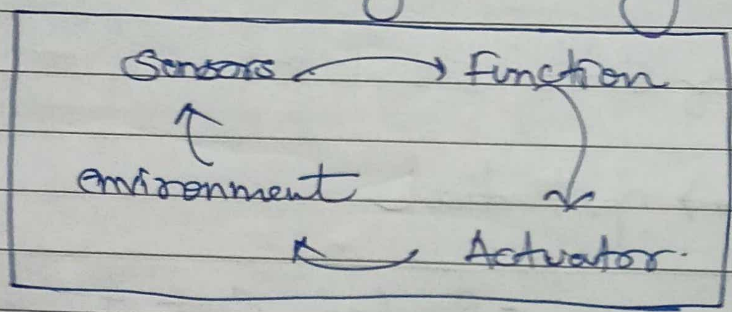
- States ⟶ helps us to take decisions

$\langle x_i, y_i \rangle$

(2)

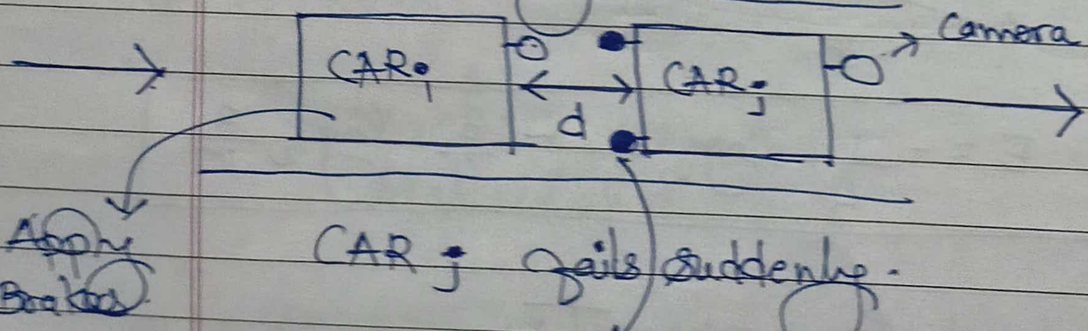


- Simple Reflex System / Agent



⊙ Don't Consider Past history / past perception for making further decisions

- Car's moving on the Road.



CAR j fails suddenly.

Yellow light glows.

example Model Reflex Agent

⊙ History info is taken into Consideration.

* Agents are goal based.

* Goal-based Reflex Agents :

  - specific task.

* Learning Agent :

  -

28/8/2022 Laboratory - Practical.

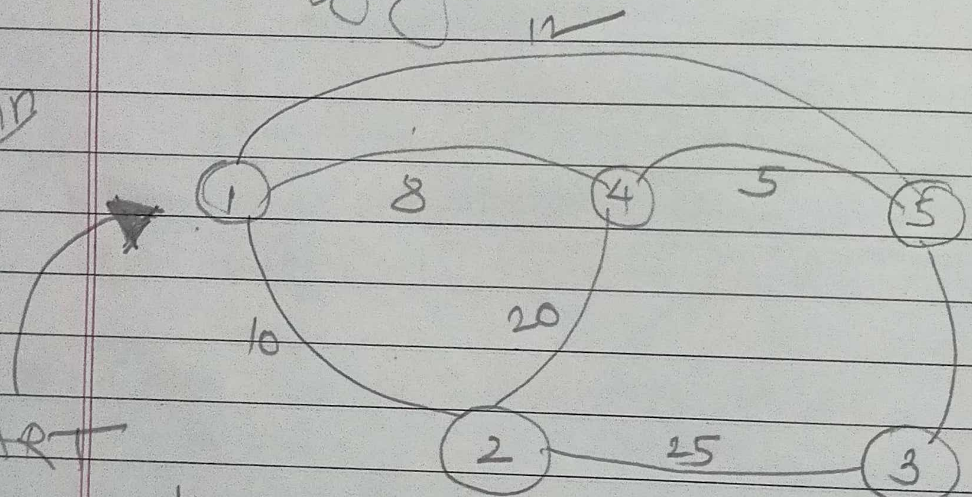
Travelling Salesman Problem.

Branch & Bound. (Optimization)

Claim - person starting a journey with his city & visit all the cities only once & coming back to his native city.

- Suggest shortest distance (cycle) by satisfying a claim.

@dmin



START salesmann city | cycle of min dist

Non polynomial time

$$A = \begin{bmatrix} \infty & 10 & \infty & 8 & 12 \\ 10 & \infty & 25 & 20 & \infty \\ \infty & 25 & \infty & \infty & 6 \\ 8 & 20 & \infty & \infty & 5 \\ 12 & \infty & 6 & 5 & \infty \end{bmatrix} \xrightarrow{\text{reduction}} A'$$

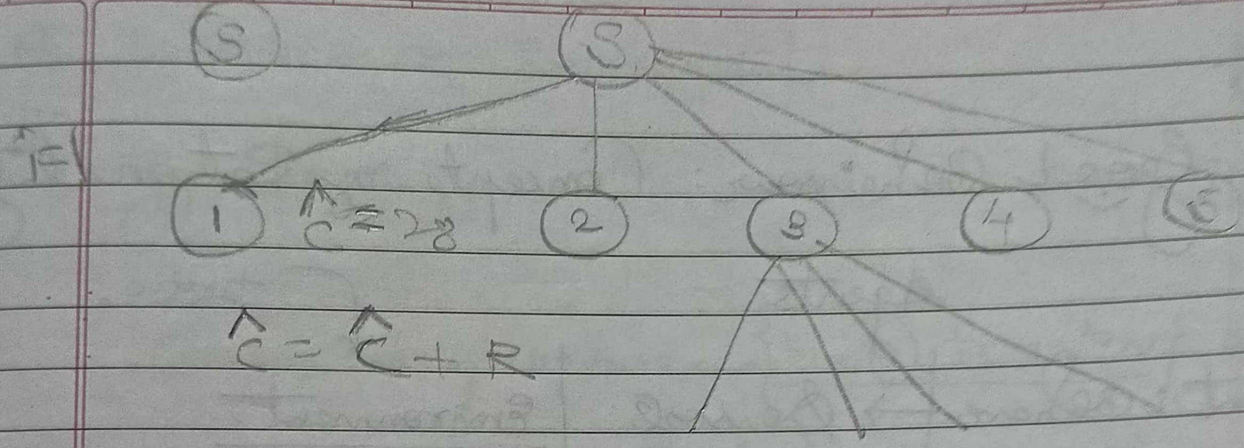
$c_1 + c_2 + c_3 + c_4 \ldots c_i = \hat{c}$ ← Bound (Initiate)

28/8/22

Python: min →
max → assume as infinity

$\hat{C} = 25$

Date
Page



$\hat{C} = \hat{C} + R$

episodic, sequencial

Date
Page

25/8/2022
Thursday

* Omniscience :

i. Omniscientist agent knows actual outcome of its actions & can get act accordingly.
ii. Omniscientist is impossible in reality,
iii. Rationality is not same as perfection. Rationally max-mizes expected performance
iv. Perfection maximizes actual performance.

* The nature of Environment :

Performance, environment, actuators & Sensors.

Properties of task environments :

① Fully Observables, partial observables.
② Single Agent & multiagent.
③ Deterministic, stochastic.

UNIT-2.
Problem Solving.

output measured in terms of actions

Algorithm Agent (model, action, rules)

model
action
rules /
pattern.

{
rule ← Actions (model, rule)

activity ← agent (rule)

model single/multiple
action {a1, a2, a3}

desired actions ← activity

return action
}

Algorithm Simple-Reflex-Agent (percept)
returns an action

Persistent: rules, a set of condition-action rules

State ← Interpret-Input (percept)
rule ← Rule-Match (state, rules)
action ← rule . ACTION.
returns action.

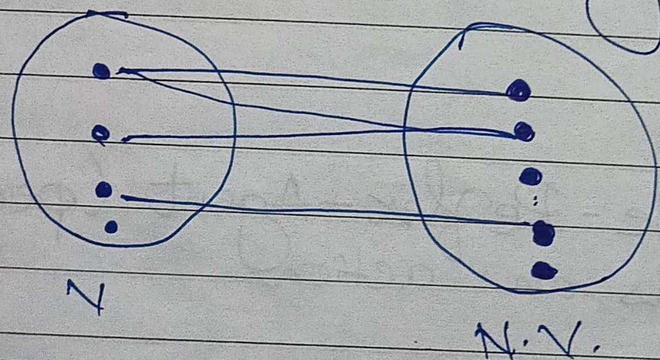
Assignment No. 5.

* Greedy Method :

1. Minimum Spanning Tree : Kruskals, Prims.
2. Dijkstra's (all pair shortest path.)

Optimization :

1. Minimizing Cost
2. Maximizing profit

3. 

select minimum edge amongst all.

Two for loops. (i, j)

Run till (i to maximum no of nodes).

Cost = cost + W[i, j]

Take Two arrays for visited & non-visited.

Date
Page

* Knapsack Problem :

Algorithm :

① for $i=1$ to n — $O(n)$

Calculate P/W ratio

② Sort object in decreasing order by P/W } $\boxed{O(n\log n)}$ overall

③ for $i=1$ to n

M = Knapsack capacity

if $(M > 0$ & $W_i \leq M)$

$M = M - W_i$;

$P = P + P_i$

else break;

if $(M > 0)$

$P = P + P_i \left(\frac{M}{W_i}\right)$

} $O(n)$

unatry method.

Consider Profit/weight ratio

Date ______
Page ______

Obj	Obj 1	Obj 2	Obj 3
Profit	25	24	15
Weight	18	15	10
P/W	1.3	1.6	1.5

Knapsack Capacity = 20

Profit का Greedy हूँ

Beginner चोर

W	Profit
2	→ 3.2
18	→ 25

20 = 5 + 15 ... total profit 28.2

15 → 24

$1 \rightarrow \frac{24}{15}$

$2 \rightarrow \frac{24}{15} \times 2$

2 → 3.2

Weight का Greedy हूँ

Intermediate चोर

10	→ 16
10	→ 15

31

15 → 24

$1 \rightarrow \frac{24}{15}$

$10 \rightarrow \frac{24}{15} \times 10$

10 → 16

अब समझा क्या

Professional चोर कैसा बनेगा।

5	7.5
15	24

31.5

10 → 15

$1 \rightarrow \frac{15}{10}$

$5 \rightarrow \frac{15}{10} \times 5$

Percept : an idea of what something is like that you get from your senses.

2/8/22 Example

Vaccuum Cleaner

- model
- state
- rules
- actions
- <u>Sequence</u>

} perceptron (percept)

- effect is on the state, so rules and action changes as output.

- Algorithm : Simple_Agent (percept)

{

State ← STATE_UPDATE (model, rules, actions)

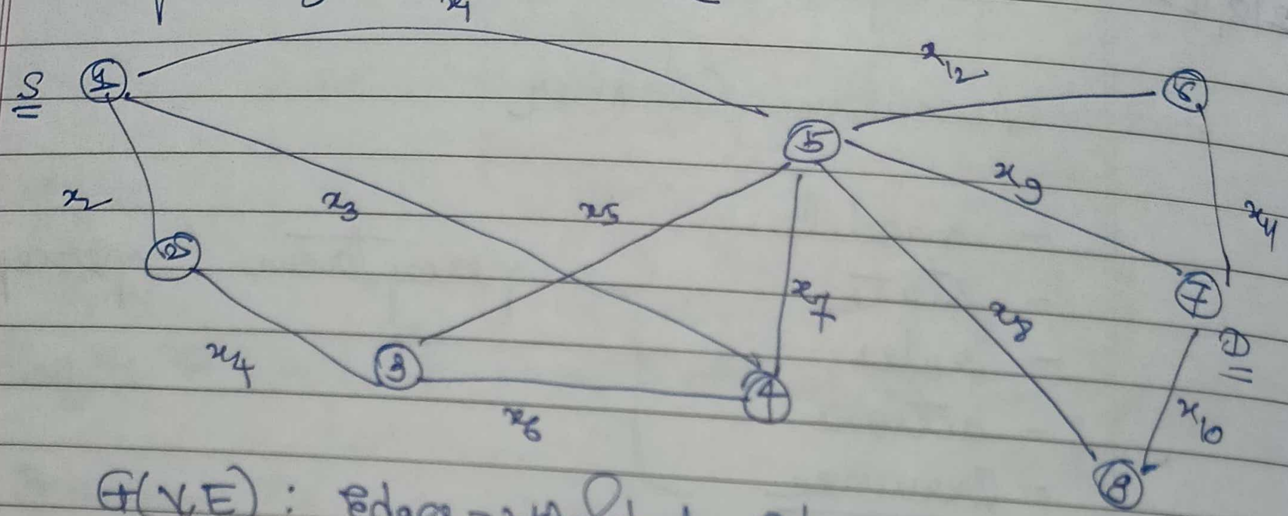
rule ← RULE_CHANGE (model, rules, actions)) previous.

actions ← rules.action

returns action

}

* Example of Tour Planner:



G(V,E) : Edges - weighted Edges.

1. model : Searching a path from a source to destination.

2. State : Nodes.

3. Rules : Branching available leading to minimum distance.

   e.g. if $x_1 > x_2 > x_3$ then select x_3

4. Actions : movement from one node to other nodes.

Sum of Subset Problem :

$L = \{x_1, x_2, x_3 \ldots, x_n\}$

Task is to choose subset S of size k such that

$\sum_{i=0}^{k} S_{\{i\}}$ ← maximum out of all other comb^n of same size.

$L = \{2, 6, 1, 5, 9, 4\}$ $k = 3$

L = | 2 | 6 | 1 | 5 | 9 | 4 |

Tree-Based-Solution.

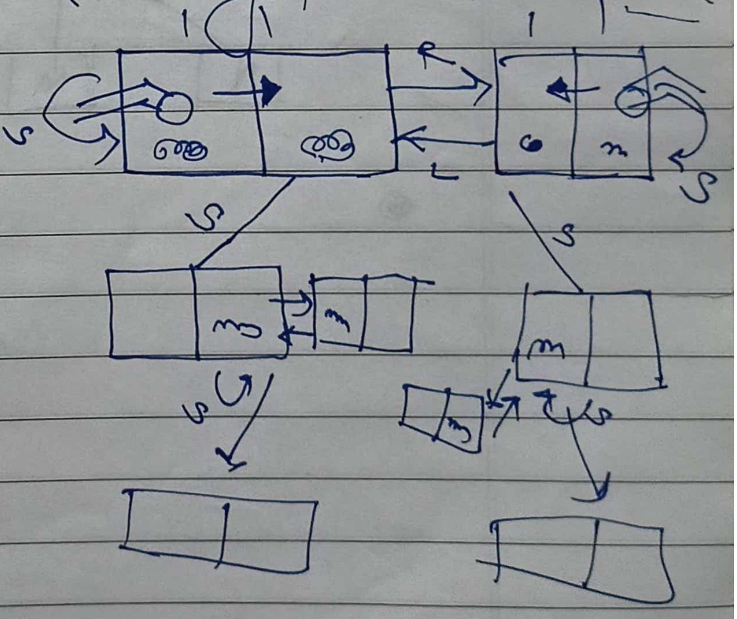
8/9/2022
Thursday

State Space Searching Problems.

① State
② Model
③ Sequence
④ Rule

finding dirt (searching problem)

eg. Vacuum Cleaner



- Working

n

- Non-Working.

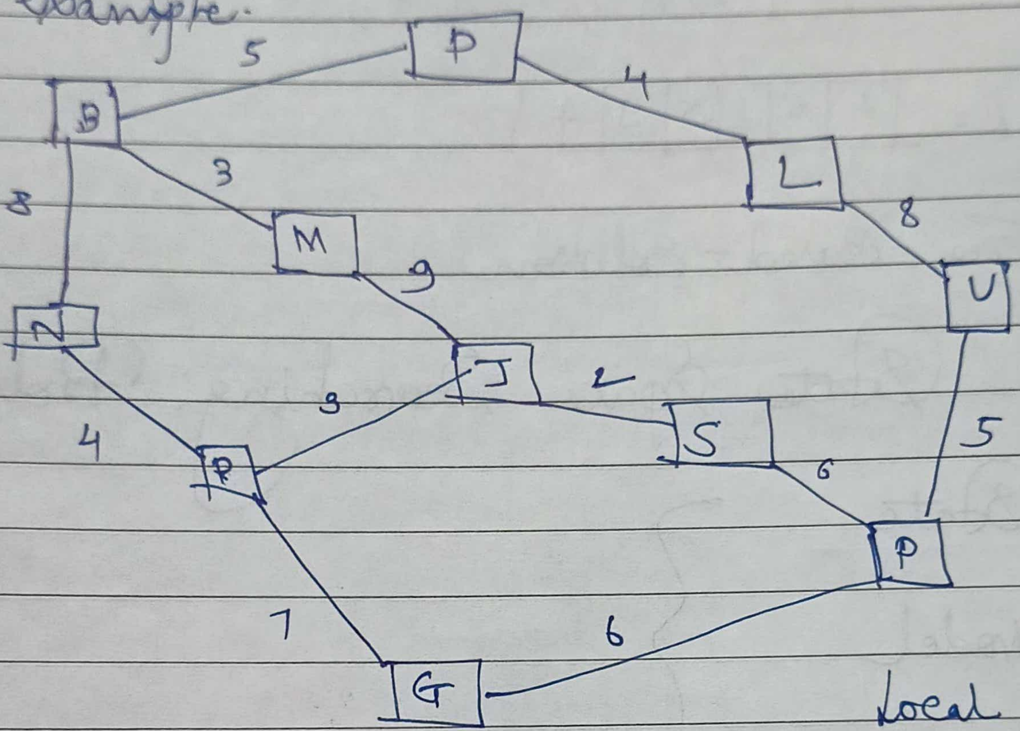
Surface   clean
          dirty

States = 2 states ?, 1

$\therefore 2 \times 2^n$

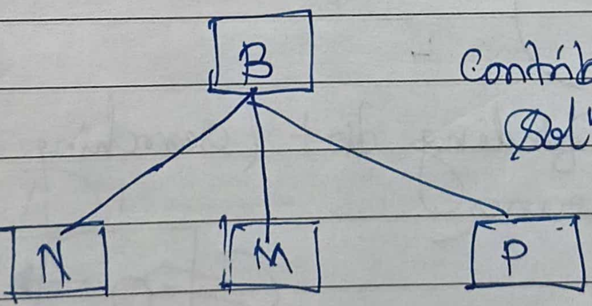
TSP :

example:



Local Optimization

Initial decision, not contributing founding of optimal Sol^n.



Whatever next node we are taking for ~~consideration~~ exploration, it has to contribute minimally while generating final cost.

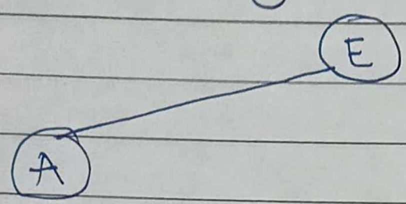
Date ______
Page ______

Assignment No. 3

Depth first & Breadth Search

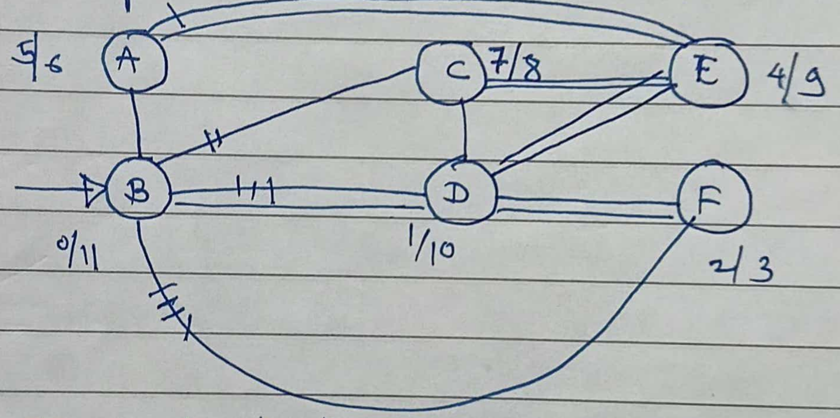
Given a Graph G(V, E)

- undirected Adjacency list, Adjacency matrix
- no weights



12/9/2022
Monday.

Depth First Search & Breadth first Search

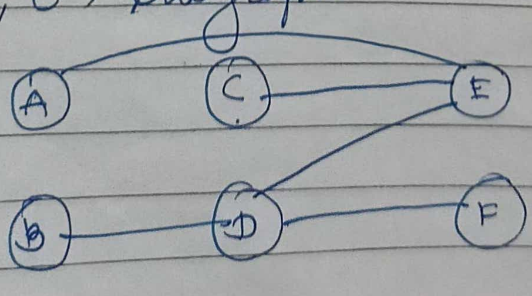


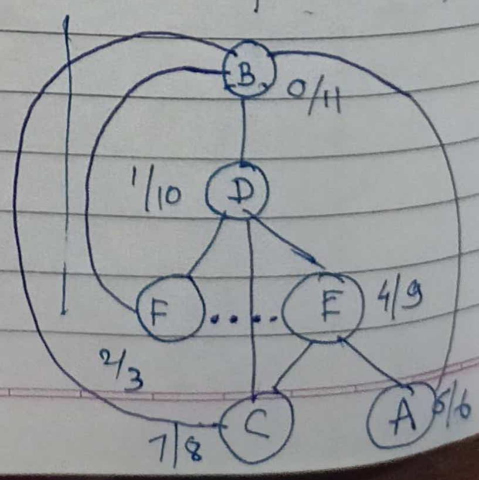
DFS → stack → un-informed Search.

X { E }

Associate every node with time ~~span~~ stamp.

G'(V, E') Subgraph

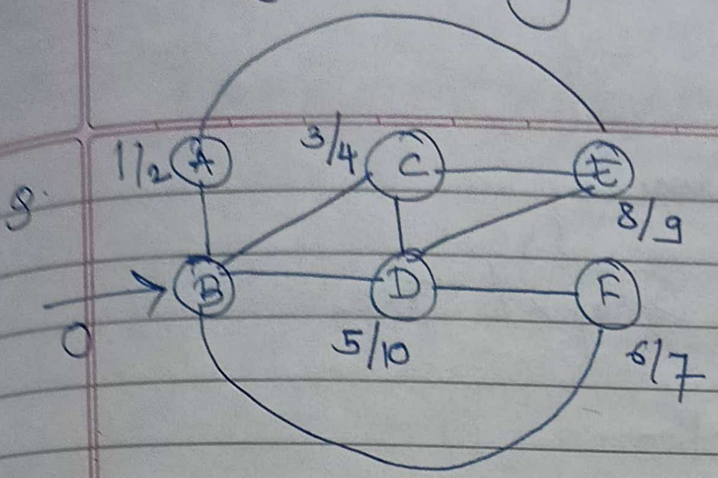




V = 6

E = n-1 } Front
          last

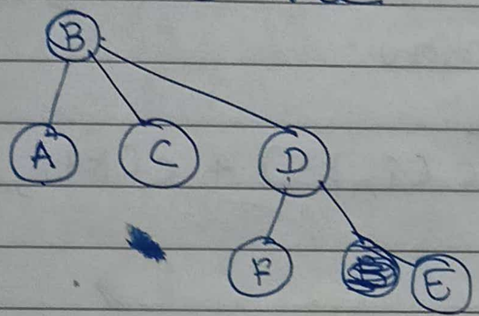
Tree edge → twicely traversed



Apply BFS

BFS →

BFS tree.



14/9/2022
Wednesday.

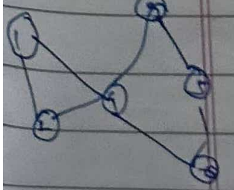
Search Optimization.

uninformed Search → BFS, DFS

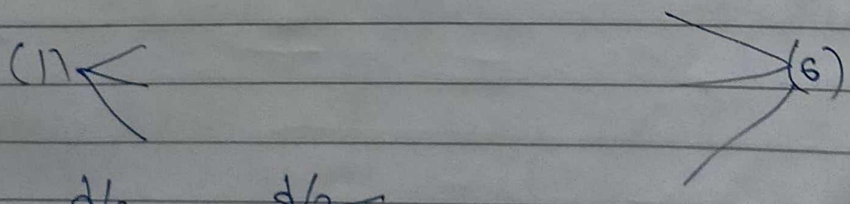
Goal, seq, states, actions, rules → percept

optimization

TSP → find tour of minimum length/cost.



Source destination



$2^{d/2} + 2^{d/2}$

2^{d}.

Date ______
Page ______

Best first approach

* Every node in a graph / Tree.

equipped with function

- OBST, TSP

Associated function with each node.

OBST

$$c[i,j] = \min_{i \le k \le j} \{ c(i, k-1) + c[k+1, j] + \sum_{i=1}^{j} p_i$$

$\cancel{c[i,j]} =$

first node

$$c[i, v-i] = \min \{ c(i,k) + \min \{ c[k, \{v-i-$$

(minimax)

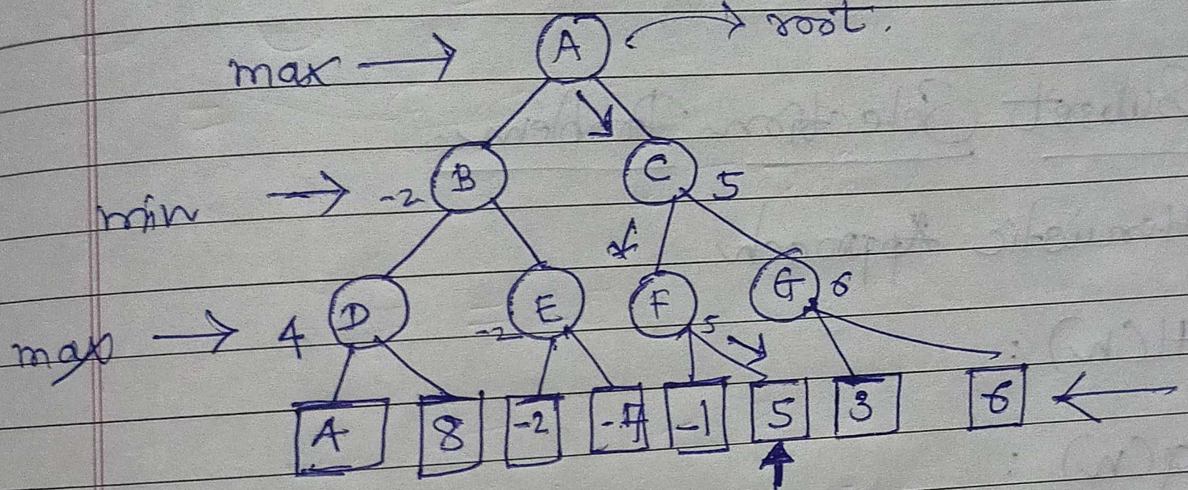
Min Max Searching Problem

22/9/2022 Wednesday

You and Opponent playing a game.

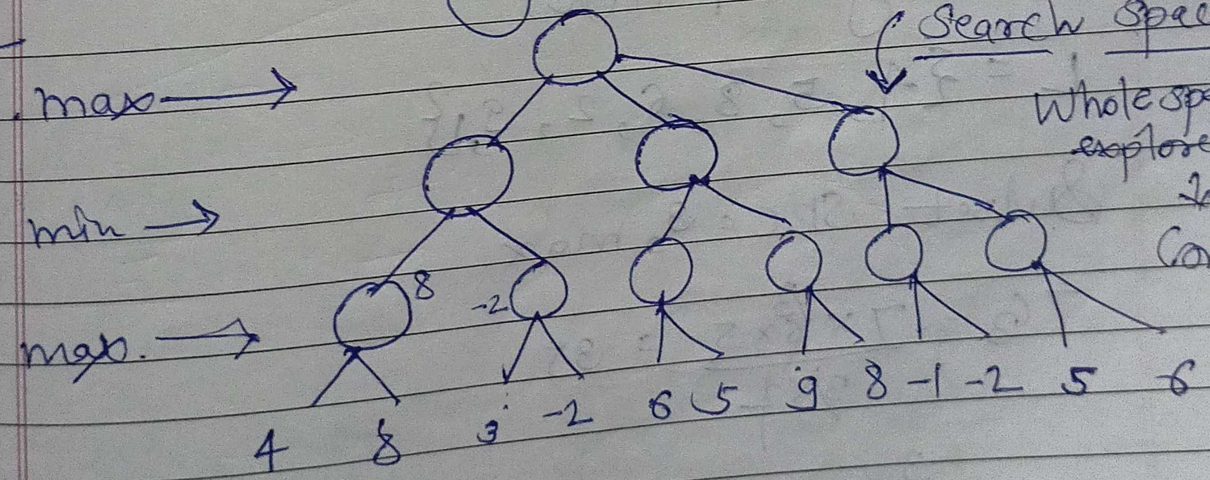
G_1 X G_2

tries to get max score — tries to minimize the runs strategy.



* α - β Pruning Algorithm

30/9/2022 Thursday



Search Space

Whole space is explored.

↓

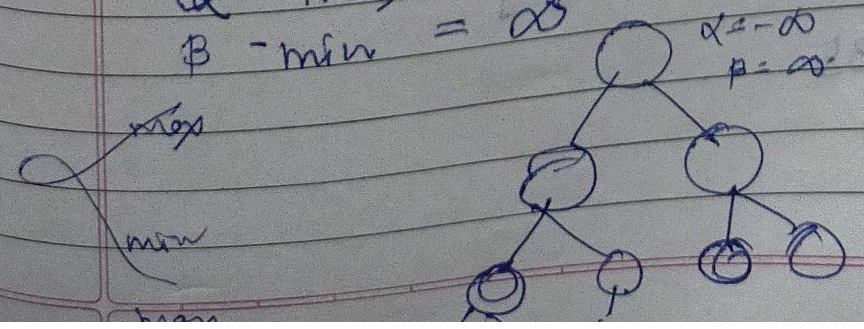
Complexity

↓

exponential time

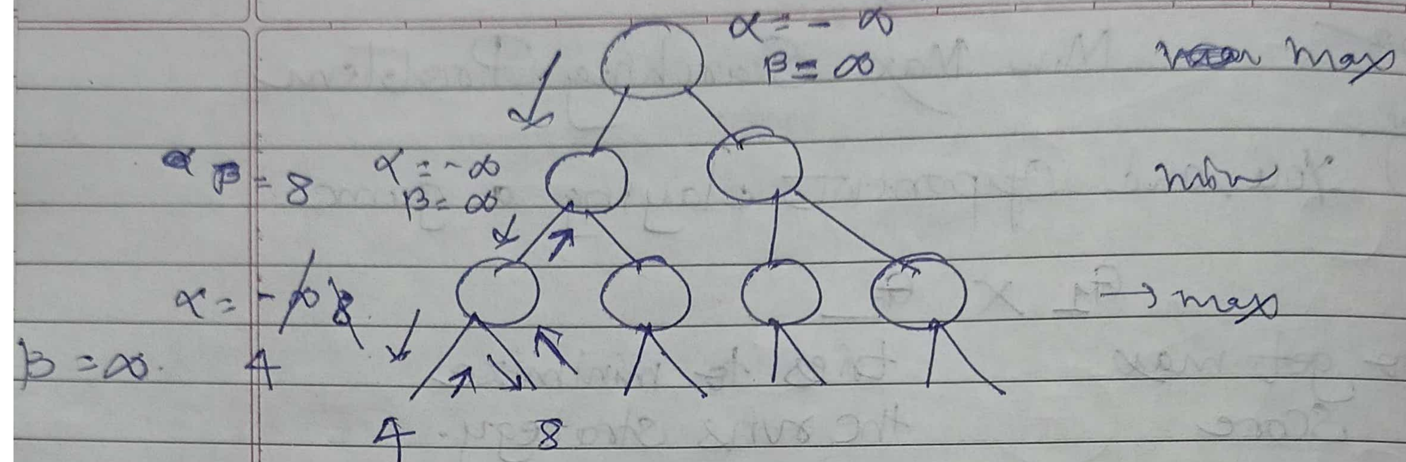
α - max = $-\infty$

β - min = ∞



min $\begin{cases} \alpha = 8 \\ \beta = \infty \\ B = \infty \end{cases}$

Condition ($\alpha \geq \beta$ then pruning)
Condition satisfied → pruning
Condition not satisfied → explore



- <u>Subset Selection Problem.</u>

Heuristic Approach

$H(n)$:

$g(n)$:

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

$L = \{4, 5, 8, 6, 2, 9, 1\}$

Subset SBC = 3, max.

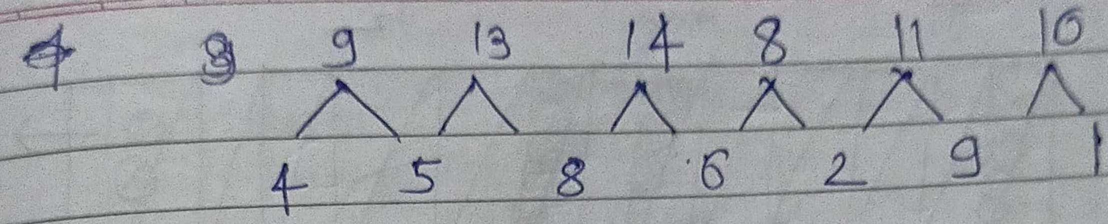
$^7C_3 = \frac{7 \times \cancel{6}^{3} \times 5}{\cancel{3} \times \cancel{2}} = 35$

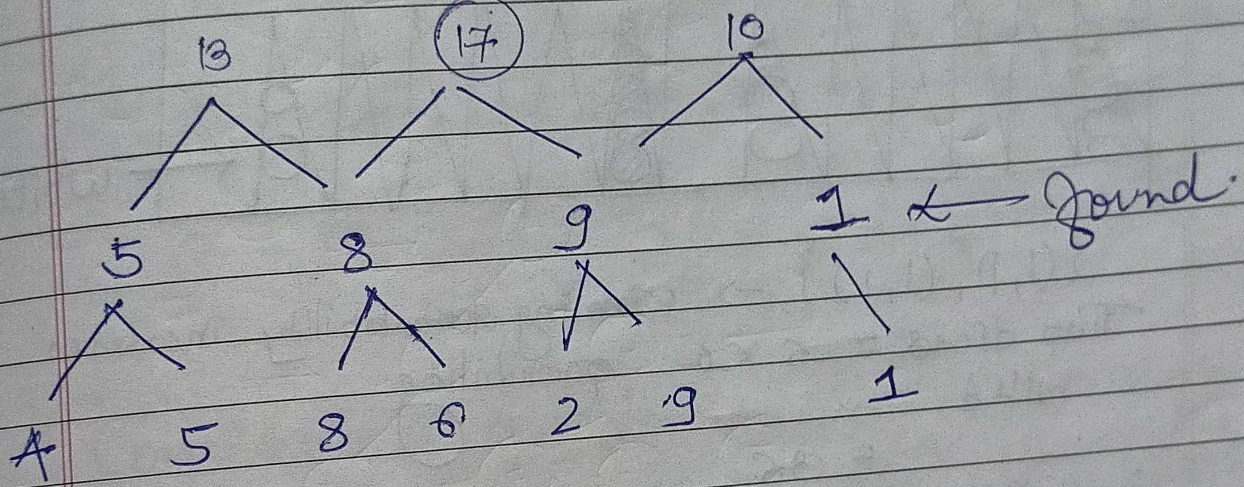
$\max \sum_{i=1}^{3} (\text{subset}[i])_j$

$1 \leq j \leq 35$

Huffman Tree A* 8 puzzle implementation.
α-β pruning 8 puzzle implementation.

checkers



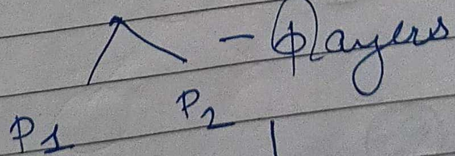


{8, 9}

17/10/2022 Monday. Game Search

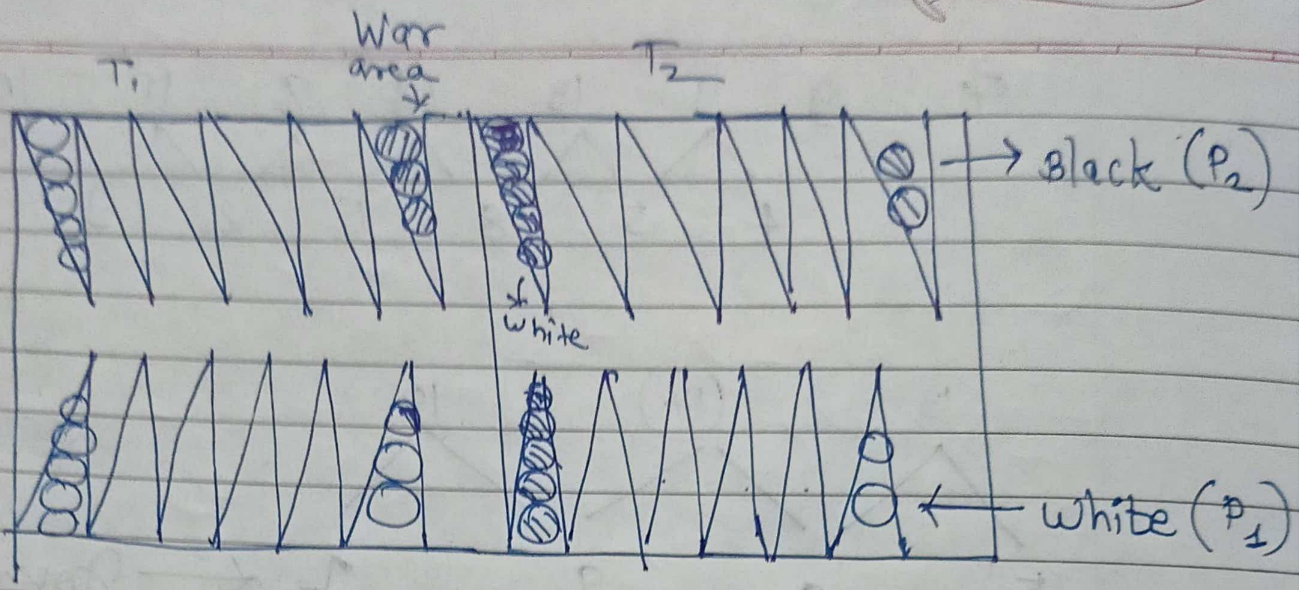
Stochastic Game Search.

* Backgammon



24 triangles | each 15 coins
white | Black

move + (luck) E(S) :



{(1,1), (1,1)} → winning probability increases.
Two dices = 6 × 6 possibility
rolled
= 36.
{ (1,1) (2,2) (3,3) (4,4) (5,5) (6,6) }

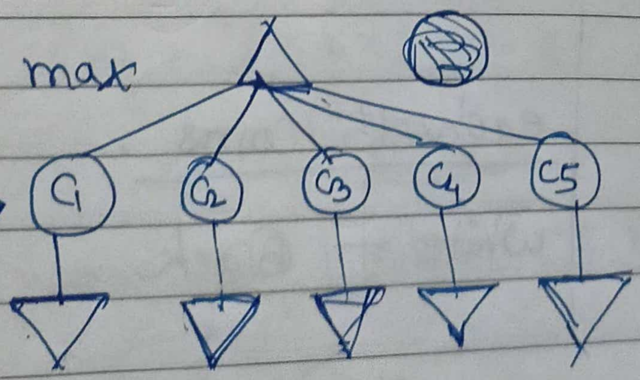
8) (1/18)

36 - 6 = 30.

$\boxed{(1,2)\ (2,1)}$ closed loop.
→ $P_1 = P_2$

(1, 2] (2, 1]

=6
dge=5



ACM digital library.
UCI Machine Learning Repository.
KAAL dataset repositary.

sci kit learn SVM.
decision tree

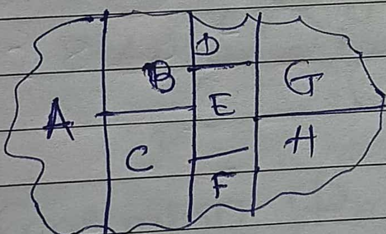
18/10/2022
Wednesday.

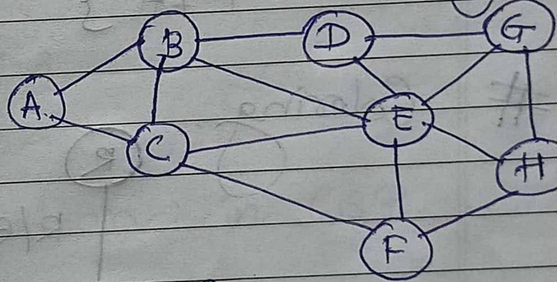
Constraint Satisfaction Problems (CSP).

X - Variables. $(x_1, x_2, x_3 \ldots, x_n)$
D - Domain $(d_1, d_2, d_3 \ldots d_n)$ ← one or more variables

C - Constraints (verified Constraints $C_1, C_2 \ldots C_n$)

Consider a problem: Graph colouring $G(V, E)$





Has edge but different color.

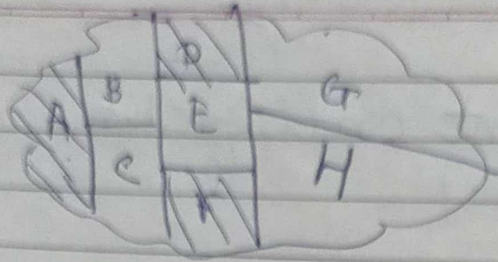
$X = \{A, B, C, D, E, F, G, H\}$

$D = \{RED, GREEN, BLUE\}$

$d_1 = \{A, D, F\}$ RED

$d_2 = \{B, H\}$ BLUE

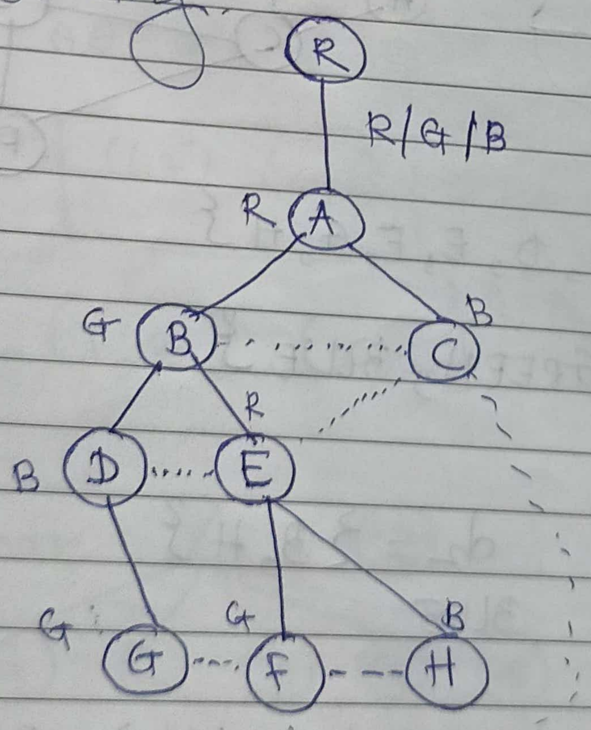
Constraints $\{A \neq B, A \neq C, \cancel{A \neq H}, B \neq C, B \neq D, B \neq E, C \neq F, C \neq E, D \neq G, E \neq H, F \neq H, G \neq H\}$



d_1 RED $= \{A, D, F\}$

$\{A \neq B,\ A \neq C$
$D \neq E,\ D \neq H$
$D \neq G,$
$C \neq F,\ E \neq F$
$F \neq H\}$

Coloring



$X = \{A, B, C, D, E, F, G, H\}$

R G B B R G G B

Knapsack Problem :

$$P = \{ \overset{I_1}{20}, \overset{I_2}{15}, \overset{I_3}{25}, \overset{I_4}{40} \} \Big\} \text{ variables } X$$

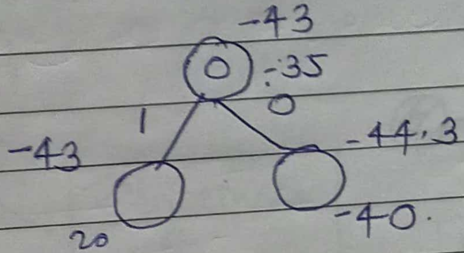
$\frac{P}{W}$ $W = \{ 5, 8, 6, 9 \}$

$m = 15 \longrightarrow$ domain D

Constraint C

maximizing $\sum_{i=1}^{n} p_i x_i$ | $x \in (0,1)$

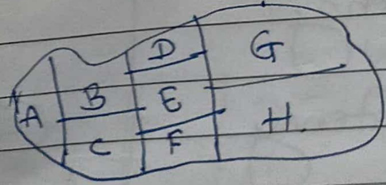
Constraint $\sum_{i=1}^{n} w_i x_i \leq m$.



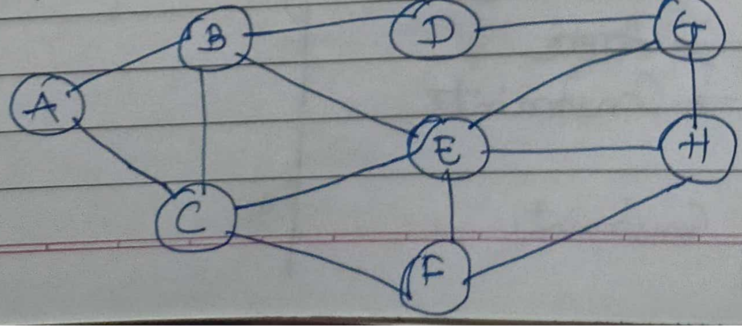
31/10/2022 Monday — Constraint Propagation : in CSP.

Sudoko game

1. Inference   2. Consistency.



Graph $G(V, E)$



Date
Page

local sdh

$D = \{ \text{Red, Green, Blue} \}$

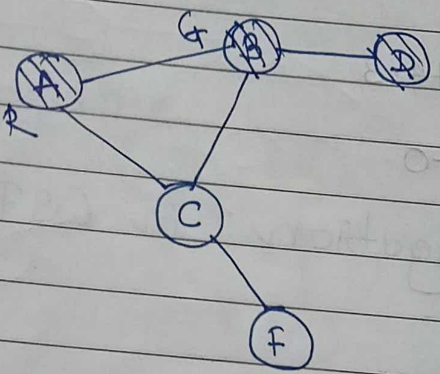
$E \rightarrow (\text{Green})$ × (E region green not allowed)

$D' = \{ \text{Red, Blue} \}$

$X = \{ A, B, C, D, E, F, G, H \}$

$D = \{ \text{Red, Green, Blue} \}$

$C = \{ E \rightarrow \{ \text{Green} \} \} \rightarrow D' = \{ \text{Red, Blue} \}$

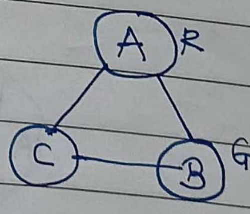


1. Node Consistency
2. Arc Consistency.
3. Path Consistency
4. K. Consistency

nodes → Area
Arc → Constraints

Arc carry Constraint

2 {Red, Green}



{R, G}

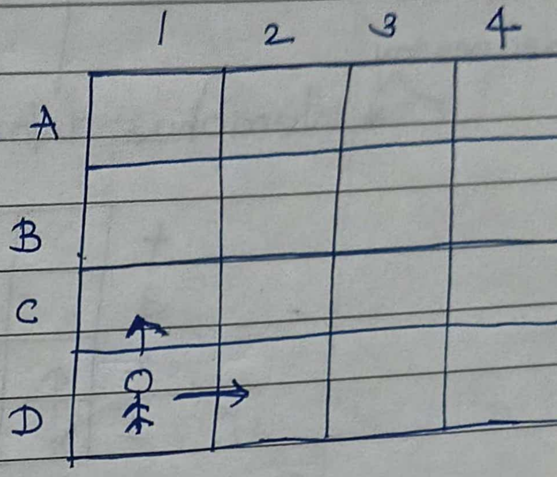
{G}

ϕ

Date
Page

11/10/2022
Wednesday

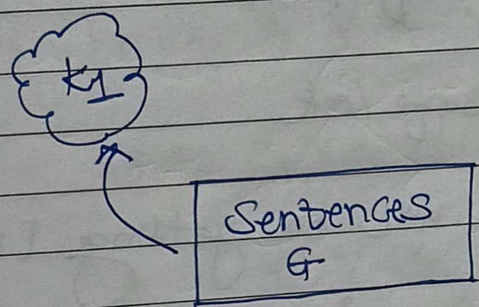
Knowledge Based Agent.



Agent ⟶ Basic knowledge

Representⁿ:
- → Starting with no knowledge
- → Starting with some knowledge.

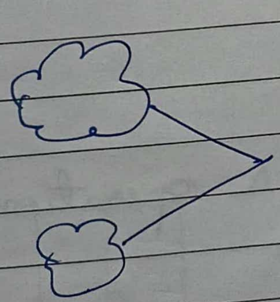
Sentences.



Natural Language Sentence

$G = \{ s_1, s_2, s_3 \ldots\ldots s_n \}$

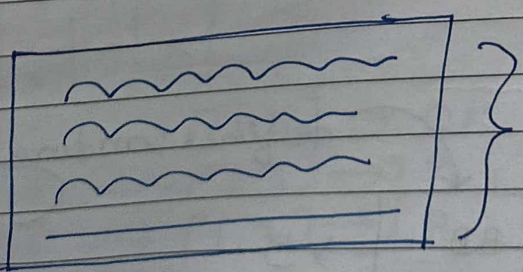
Technical terms.



Rules { Grammer }

Sentences → paragraphs

Words



- meaning
- Connect, Stop words.

if, else, a, an, the

① PIT
② Breeze
③ Stench
④ Gold.

$\boxed{\text{Goal : Geeting Gold.}}$

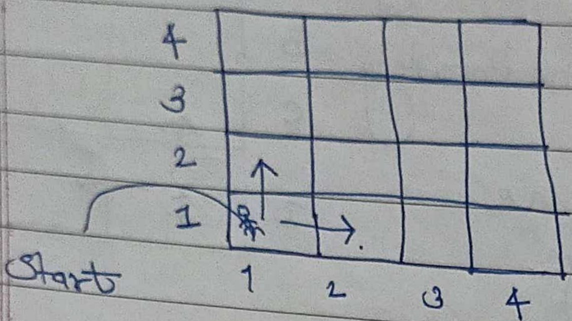
Date
Page

$\boxed{\text{Pit : Dangerous, Probability of loosing}}$

Knowledge

2/10/2022
Wednesday

* Wumphus World



16 places
[1, 1] → start

15 : probabilities 1/15

Changes according to learning opinions.

1/15 → [2, 1] → safe/ok

1/15 → [1, 2] → ~~safe~~/ok

How many times we will be getting [2, 1] ok

100 {40, 20, 20, 20}

Probabilistic answer

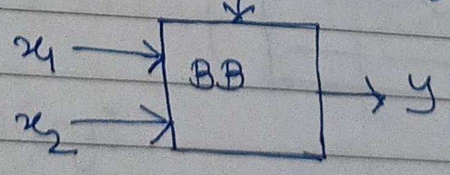
* Relational Algebra. ¬, ∨, ∧ operations
⇒ ⟺

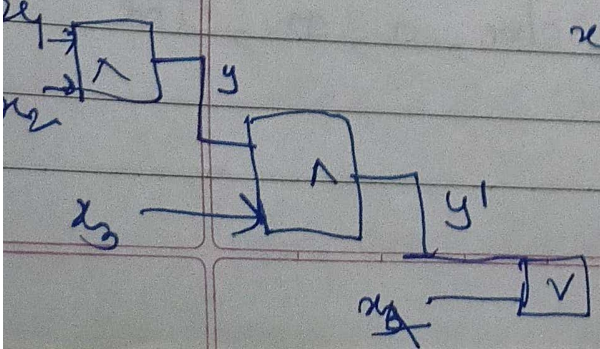
$\boxed{(x_1 \wedge x_2) \vee x_3} \rightarrow$ clause

Anding

System

defines anding.





$\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$

Date

Page

0/2022

...sday

clauses

Logic

Using these Connection : logic { clauses }

PP Formula $F : C_1 \wedge C_2 \wedge C_3 \wedge C_4$

Clauses Contains 2n variables

$\{x_1, x_2, x_3 \dots x_n\}$ $\{\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4, \bar{x}_5 \dots \bar{x}_n\}$

$$\left.\begin{aligned} C_1 &= \{ x_1 \vee x_2 \vee \bar{x}_3 \} \\ C_2 &= \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3 \vee x_5 \\ C_3 &= x_1 \vee x_2 \vee x_3 \\ C_4 &= \bar{x}_1 \vee x_2 \vee \bar{x}_3 \vee x_4 \end{aligned}\right\}$$

PP_1 $\neg x_1$

Atomic

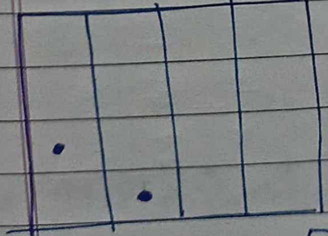
KB

Atomic Sentence

Complex Representation.

molecular statement

molecular Sentences.



$\rightarrow [1,1] \rightarrow$ OK

$\left.\begin{aligned} &\rightarrow [2,1] \rightarrow \neg P_{2,1} \\ &\rightarrow [1,2] \rightarrow P_{1,2} \end{aligned}\right\}$ Atomic / moolakule

Trimfey Manohar

B → Breeze.

Date

Page

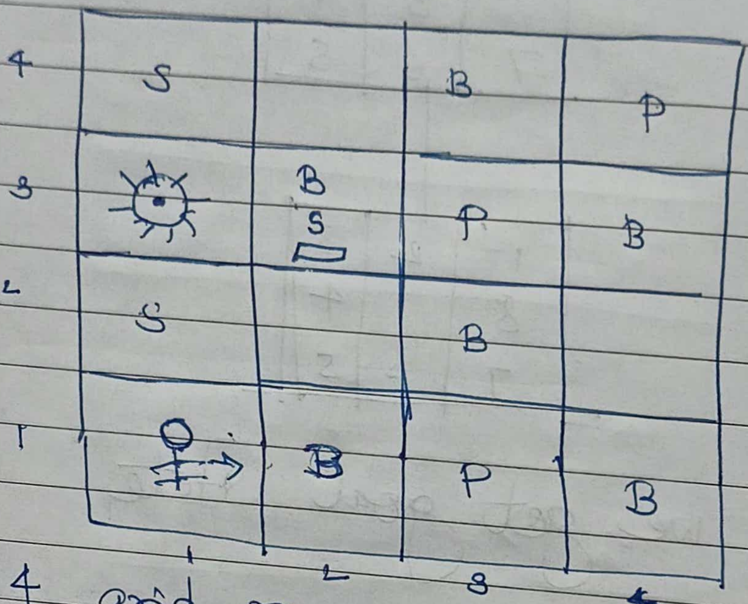
$m_1 = \{P_{2,1} = True, P_{1,2} = False, P_{3,1} = True\}$

$F_1 = P_{21} \vee P_{12} \vee P_{31} = T \vee F \vee T \rightarrow T$

$G = (P_{21} \vee P_{12}) \wedge P_{31} \leftrightarrow P_B$ } safe/ok

* SAT Problem (Satisfability problem)
- NP Complete problem.

* WUMPUS world problem :



(1) • 4×4 grid room

(2) • There will be an agent whose goal is to find the gold and bring it back to the start without getting killed.

(3) • The agents always starts in [1, 1], facing to the right.

(4) The agent has only arrow.

(5) ⇒ The agent perceives

(1) A stench ~~in the~~ in the block containing the wumpus and in adjacent squares not diagonally.

(2) A Breeze in the squares adjacent to a pit.

③ ~~A~~ Glitter in the square / block where gold is present.

④ Bump, if it walks to the wall.

⑤ Scream everywhere in the cave, if Wumpus killed.

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2	2,2	3,2	4,2
1,1 A ok	2,1	3,1	4,1

No Stench No Breeze

1,3			
1,2	P?	3,2	4,2
1,1 ok	2,1 A	3,1 P?	4,1

Breeze so go back

S → Stench, W → Wumpus, B → Breeze, A → Agent

V → Visited

(R_1) $\neg S_{11} \rightarrow \neg W_{11} \wedge \neg W_{12} \wedge \neg W_{21}$

(R_2) $\neg S_{21} \rightarrow \neg W_{11} \wedge \neg W_{21} \wedge \neg W_{31}$

(R_3) $\neg S_{12} \rightarrow \neg W_{11} \wedge \neg W_{22} \wedge \neg W_{12} \wedge \neg W_{13}$

(R_4) $S_{12} \rightarrow W_{13} \vee W_{12} \vee W_{22} \vee W_{11}$

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 A ok S	22	3,2	4,2
1,1	2,1 B ok V	3,1 P?	4,1

1,4	2,4 P?		
1,3	2,3 A S B G	P?	
1,2 S V ok	2,2 ok V		
1,1 V ok	2,1 B ok V		

$P \to$ True

$P \to Q$. the Q ~~need~~ is true.

Date
Page

>> Apply Modex Ponenx with $\neg S_{11}$ and R_1:

(R1) $\neg S_{11} \to \neg W_{11} \wedge \neg W_{12} \wedge \neg W_{21}$ $\quad \neg S_{11}$

$$\boxed{\neg W_{11} \wedge \neg W_{12} \wedge \neg W_{21}}$$

Apply And-Elimination to this we get:

$\neg W_{11}, \neg W_{12}, \neg W_{21}$

>> Apply Modux Ponnex to $\neg S_{21}$ and R_2:

(R2) $\neg S_{21} \to \neg W_{11} \wedge \neg W_{21} \wedge \neg W_{22} \wedge \neg W_{31}$ $\quad \neg S_{21}$

$$\boxed{\neg W_{11} \wedge \neg W_{21} \wedge \neg W_{22} \wedge \neg W_{31}}$$

Apply ~~Modux Ponnex~~ AND-elimination to this we get:

$\neg W_{21}, \neg W_{22}, \neg W_{31}$ ($\neg W_{11}$ already written above)

>> Apply Modux Ponnex to S_{22} and R_4

(R4) $S_{12} \to W_{13} \vee W_{12} \vee W_{22} \vee W_{11}$ $\quad S_{12}$

$$\boxed{W_{13} \vee W_{12} \vee W_{22} \vee W_{11}}$$

>> Apply unit resolution ~~with~~ on $W_{13} \vee W_{12} \vee W_{22}$ 4

$\neg W_{22}$

Unit Resolution W_{11} & $\neg W_{11}$ cancel out.

Date
Page

$\boxed{W_{13} \vee W_{12} \vee W_{22}}$

Apply unit resolution $W_{13} \vee W_{12} \vee W_{22}$ & $\neg W_{22}$

$\boxed{W_{13} \vee W_{12}}$

Again apply unit resolution with $W_{13} \vee W_{12}$ & $\neg W_{12}$

$\boxed{W_{13}}$

Proved.

Propositional Logic.

Syntax : What we see

Semantics : What we experience

$$\boxed{f : C_1 \wedge C_2 \wedge C_3 \wedge C_4}$$

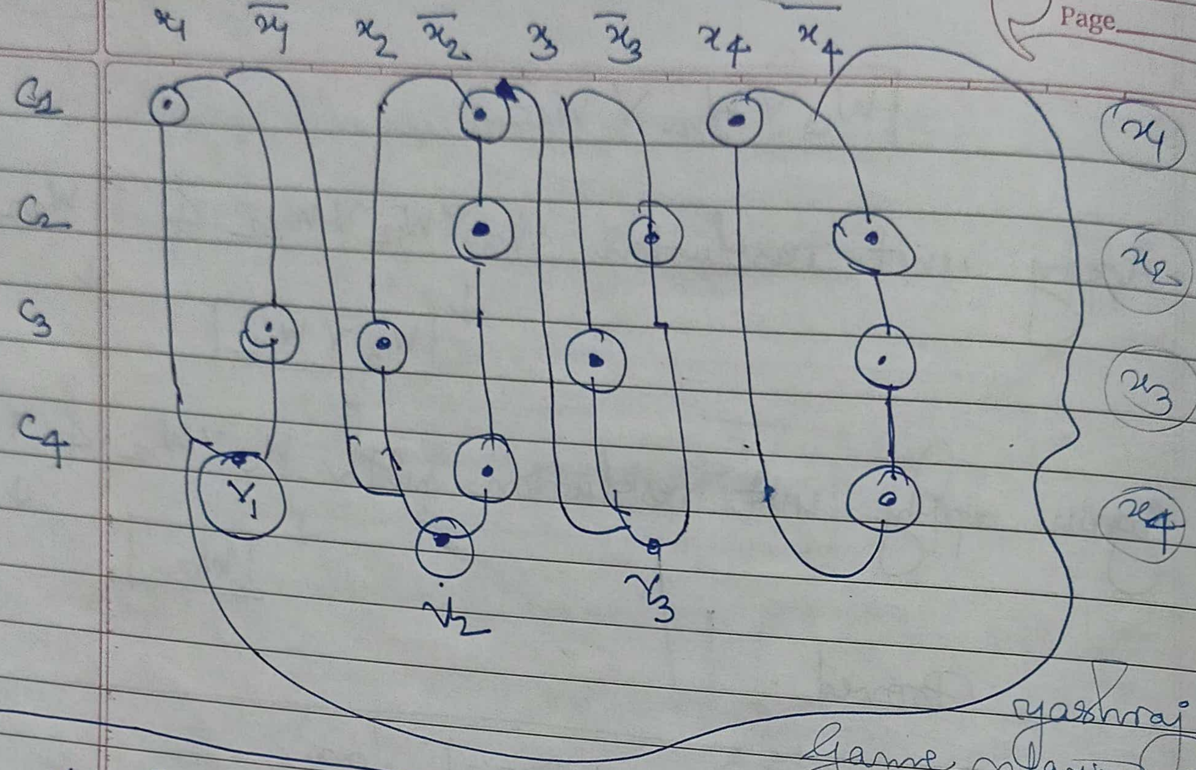
$$C_1 = x_1 \vee \overline{x_2} \vee x_4$$

$$C_2 = \overline{x_2} \vee \overline{x_3} \vee \overline{x_4}$$

$$C_3 = \overline{x_1} \vee x_3 \vee \overline{x_4} \vee x_2$$

$$C_4 = \overline{x_2} \vee \overline{x_4}$$

Jerts Jerrat.



yashraj
Game playing Of

* Game Playing

always

1. There is always an unpredictable opponent
2. Opponents also wants to win
3. Opponent create uncertainty.
4. Time limit.

Fully Observable : When agent is able to know the state of the world from the stimuli is fully observable.

When agent is able to sense using sensor the complete environment is F.O.g.

e.g. Chess,