CSE366

DS

303

29.01.24 / CSE 366

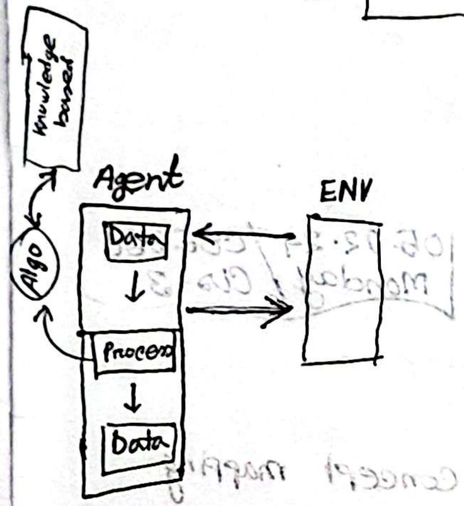
Monday / Cls-1

366

475

Data → Process (AI/Application) → Deploy

Input → Deploy → Output



AI

- → Agent
- → Problem solving using searching
- → Knowledge Base

Knowledge Base ↓ Machine learning → Deep learning

Machine learning → Data → Algorithm (train) → Test

Marks

Mid → 20

Final → 30

Quiz → (3) 10

AH → 05

lab Perf → 10

Project → 25

- → Report
- → Presentation
- → Viva

Large language Model

- → Lama
- → LAVA

LLM

LLAMA — 2 → 7B

LLAMA - Fatory

Gradle

GUI

Fine tuning

wed / Cls-2

Stimuli → Input

Perfect rationality → AGI

Bounded rationality → AI

05.02.24 / CSE366
Monday / Cls-3

Modularity →

Planning horizon

Finite stage

Indefinite stage:

Infinite stage:

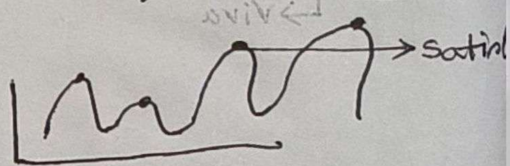
** একটা concept mapping থাকবে xm এ
Human, **

07.02.24 / CSE 366
Wednes / Cls-4

AI → Evaluation function

→ Objective → Max

→ Cost → Min



offline

Agent → ENV

* Problem

High level

↓

Represent

↓

Solution

↓

Quality of Solution

online

Problem Solving using search
- → A*, UCS, heuristics
- → local search (optimization)
  - → Genetic Algorithm
- − Q-Learning

Mid

- → Ad versial search
  - Min-max, α-β, pre
- → CSP

→ Learning in AI (ML)
- → KB
- → ML fundamental
- → DP
- → NLP → LLM → RAG

12.02.24 / CSE366
Monday / Ch-5

// Simple Reflex Agents:

Goal Based agent
Utility Based agent

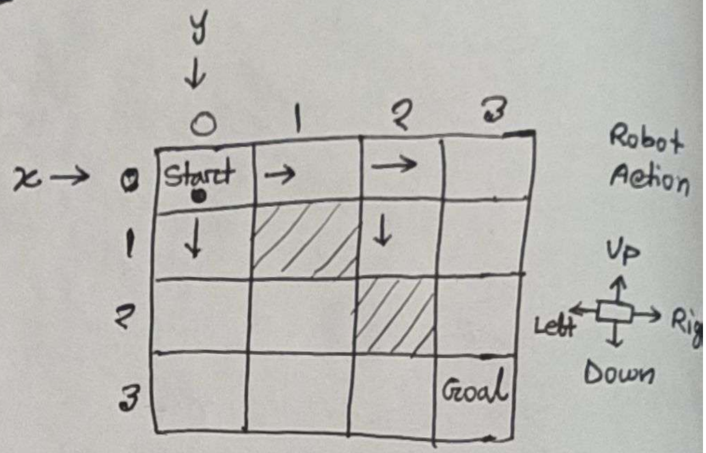
- Simple Reflex → Condition → Action = Rule
- Model Based → State + Condition
- Goal Based → State + Goal
- Utility Based → State + Utility (performance measure)

→ Learning agent → Knowledge Based + State + Goal + Utility

#Search Algorithms in AI :

Problem Solving Agent →

→ Search Space
→ Start State
→ Goal test
→ Search tree



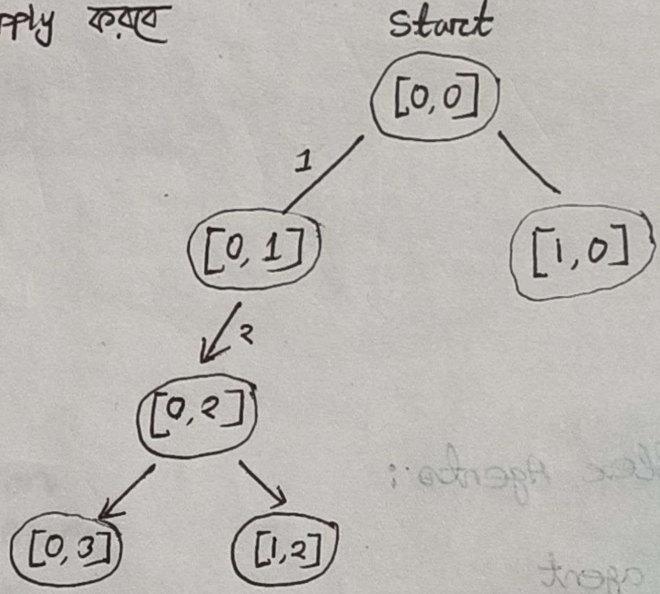
Search space

<u>Action</u>: Robot {Up, down, left, right}

Action → Ability

<u>Transition model</u>: Robot কিভাবে কত ভাবে Action Apply করবে তা।

Search tree



<u>Path Cost</u>:

<u>f(n)</u>

Best solution

Optimal solution

* Completenes : At least 1 ta Solution যেন পাই।
* Optimality :
* Time Complexity :
* Space Complexity :

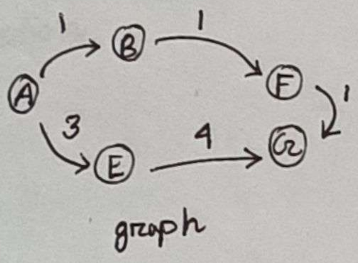
1. Uninformed / Blind Search

* BFS
* DFS
* Uniform Cost Search
* Depth First Search (DFS)
* Bidirectional search
* Interative deepling dept first Search

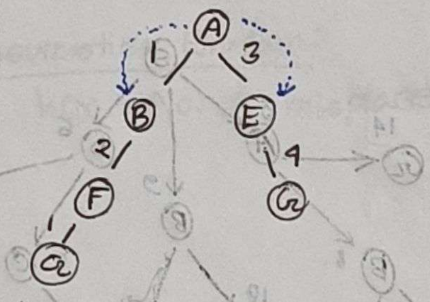
Blind

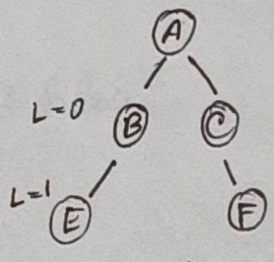
Informed Search

* Best First Search
* A* search



graph





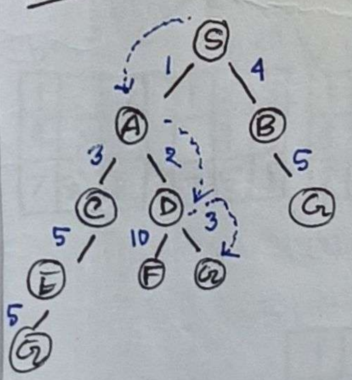
(IDDFS)

Priority Queue —

B	E

E	F

Uniform Cost Search mainly depends on Priority Queue.



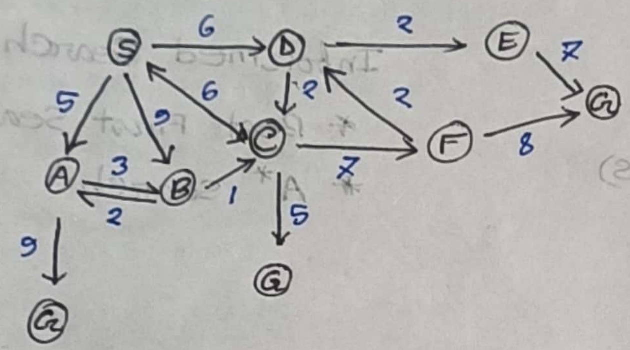
Start : S
Goal : G

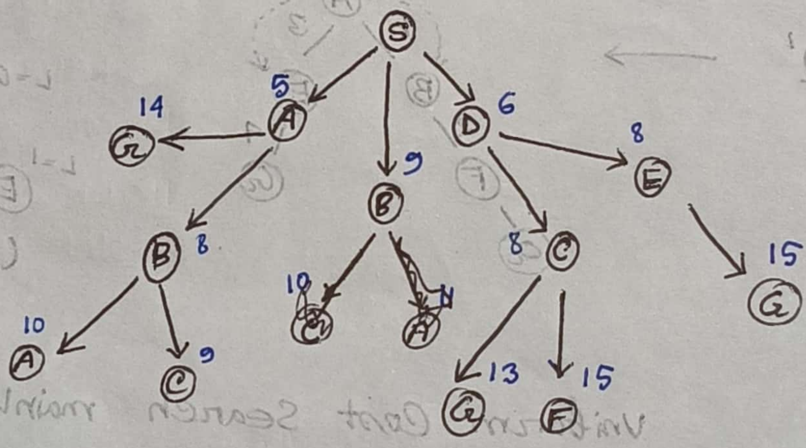
Q : A(1) , B(4)
Select A (1)

Q: C(4) , D(3), B(4)
Q: D(3), B(4) , C(4)

Select D (3)

7 Exam





$S \rightarrow D \rightarrow C \rightarrow G$

Informed Search

* Best first Search
* A* Search

Heuristics Function : h(n)

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

Heuristics Cost — Estimated Cost

Admissibility of the heuristics function

8 puzzle Problem

Start

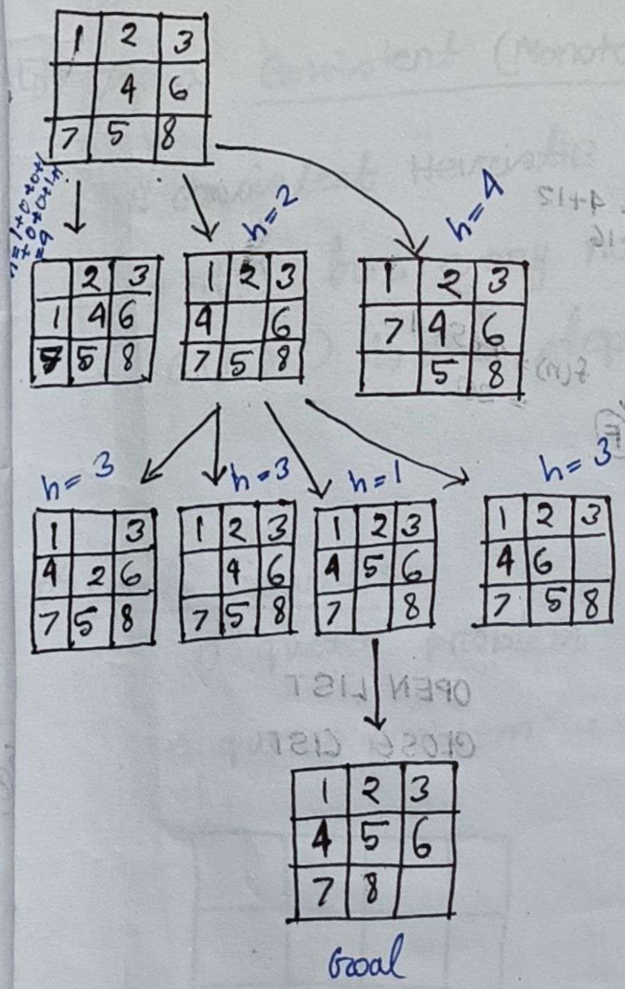
1	2	3
	4	6
7	5	8

Goal

1	2	3
4	5	6
7	8	

heuristics Function :

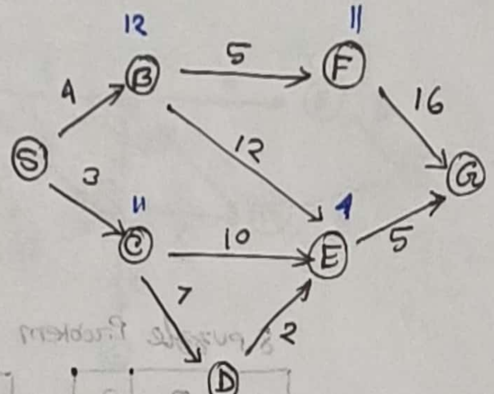
h(n) = No. of misplaced tiles



$A^* = UCS + BFS$

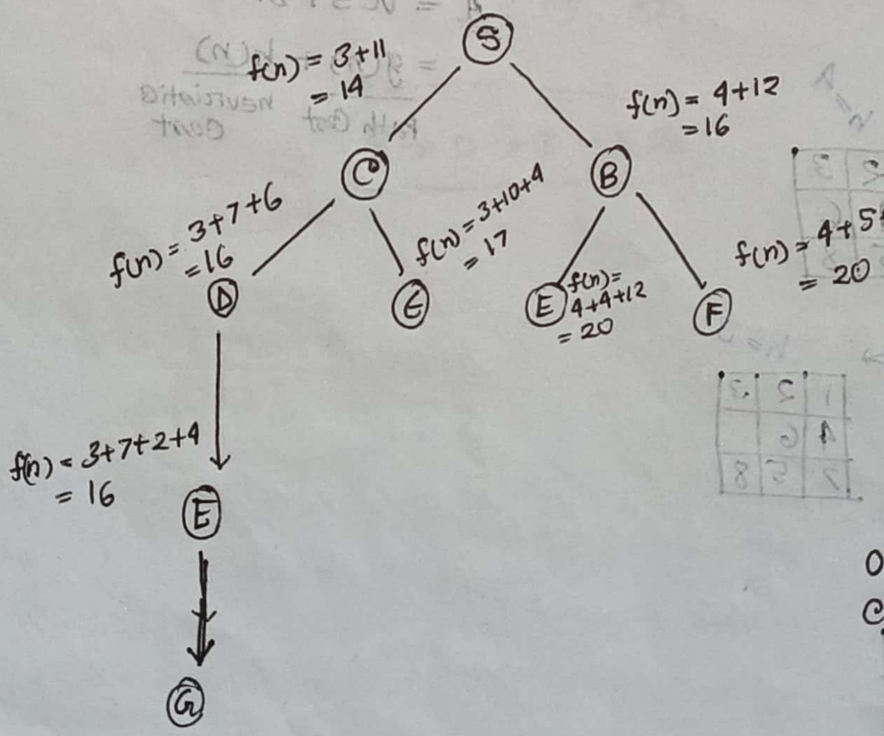
$= g(n) + h(n)$

Path Cost — heuristic cost



Node	G(n)
S	14
B	12
C	11
D	6
E	4
F	11
G	0

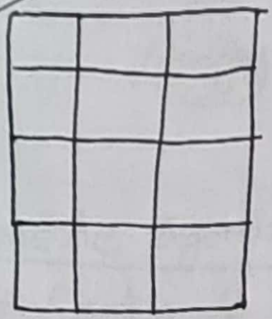
$$f(n) = g(n) + h(n)$$



Solution Path : S → C → D → E → G

Robot battery level to grid সাজাও

10×10



20.02.24 / CSE366
Tuesday / Cls-6

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

$g(n)$ → Path

Admissible Heuristic: $h(n) \leq h^*(n)$ → A^* → optimal

Inadmissible Heuristic: $h(n) > h^*(n)$ → Graph Search এ use করি।

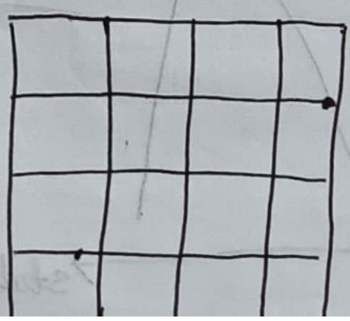
☐ Use a Consistent (Monotonic) Heuristic:

A consistent Heuristic satisfies the condition $h(n) \leq c(n, n') + h(n')$ for every node n and its successor n', where $c(n, n')$ is the step cost from n to n'

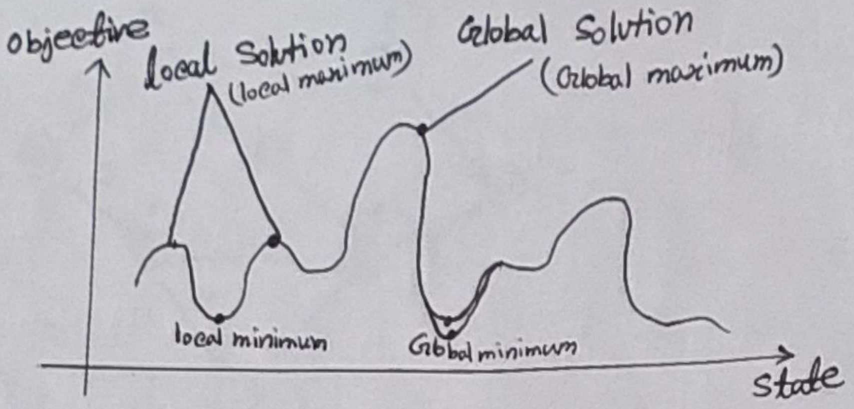
Local Search:

n-queen problem

4 queen problem



Hill Climbing—



→ Hill climbing
→ Simulated Annealing

04·03·24 / CSE03
Monday / Cls-

AI
→ Agent
Path finding
Search
- Informed
- Uninformed

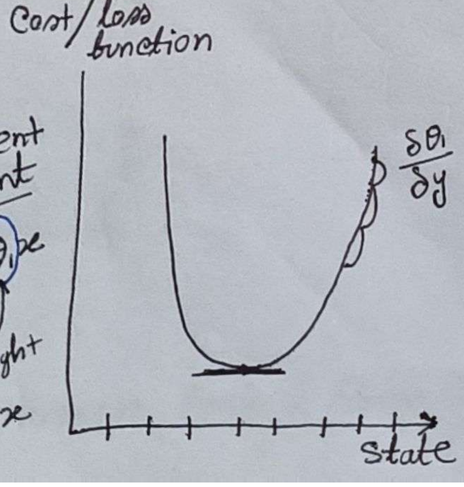
→ Local Search
Optimal Solution}

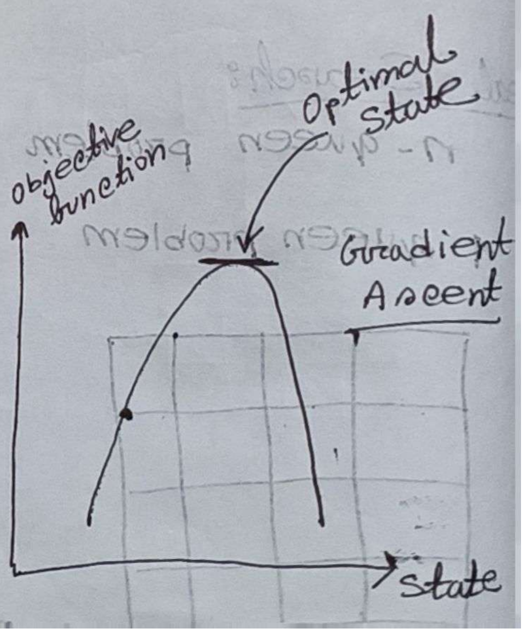
Gradient Descent

$h(x) = \theta_0 + \theta_1 x$

bias
weight

$y = \theta_0 + \theta_1 x$

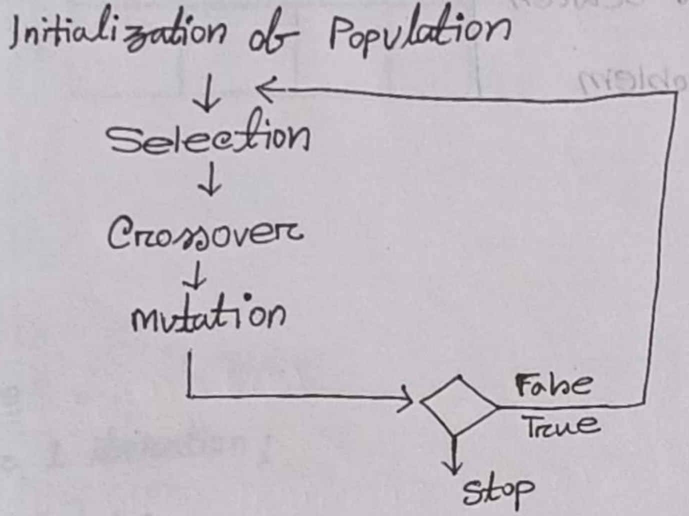




$$\theta_1 = \theta_1 - \alpha \frac{\delta \theta_1}{\delta y} \theta_1$$

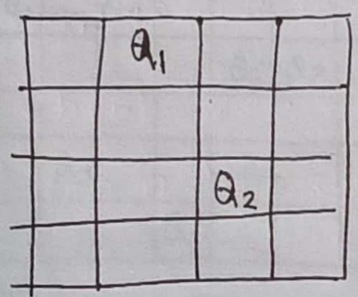
learning rate

Genetic Algorithm:



* Step-1: Initialize population
* Step-2: Selection according to fitness → fitness function
* Step-3: Cross Over
* Step4: Perform mutation
* Step-5: Repeat cycle till the condition of stop is true.

* n queen Problem



AI

Agent and Environment

[Informed and Uninformed Search] ↷ Quiz

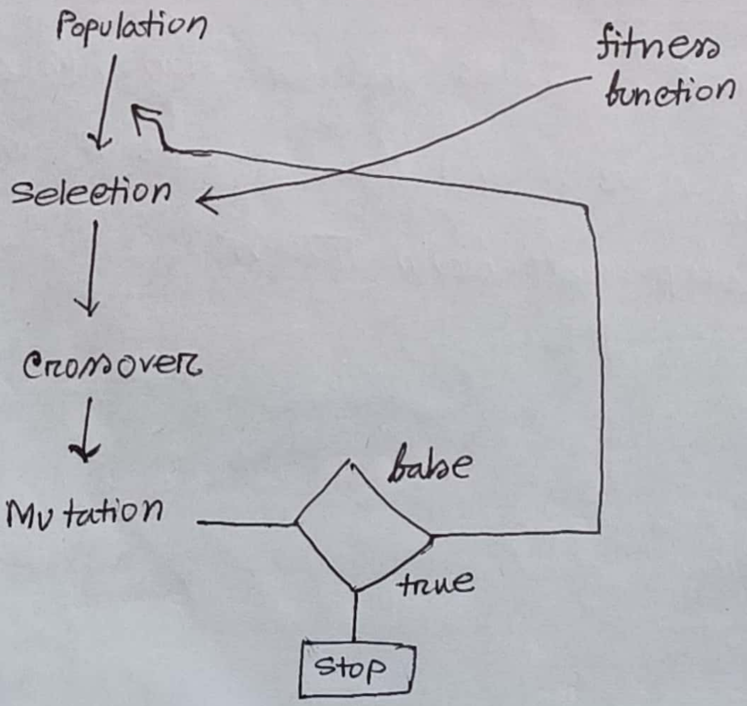
Ch-4 Local Search

Ch-5 Game Playing and Adversarial Search

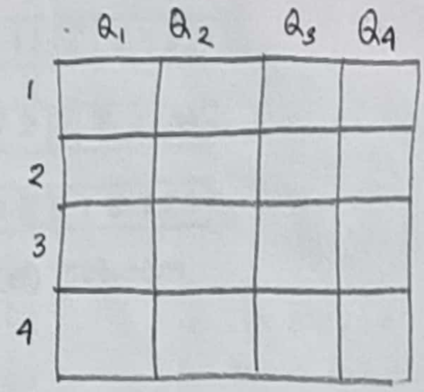
Ch-6 Contraint Satisfaction Problem

- Hill climbing.
- Simulated Anealing.
- Gradiant Descent.
- Genetic Algorithm

\# Initialization of population



8 Queen Problem:



\# Consider the following individual state for 8 queen problem and perform Genetic Algorithm with the threshold of selection 16:

(a) State 1: [2 4 7 4 8 5 5 2]

(b) State 2: [3 2 7 5 2 4 1 1]

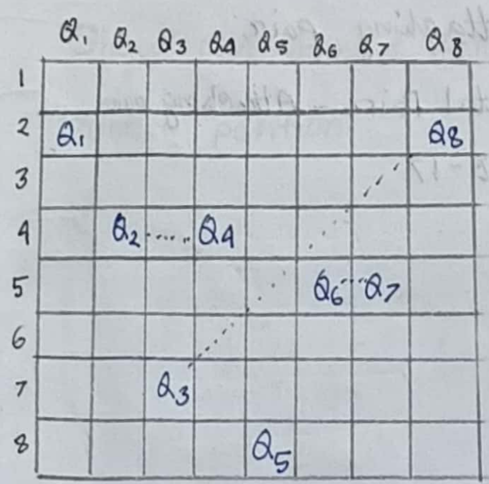
(c) State 3: [2 4 4 1 5 1 2 4]

(d) State 4: [3 2 5 4 3 2 1 3]

Solve

For 1 iteration;

State 1:



Selection:

fitness function: no of non attacking pair

total $= 7\times 8 = 56$

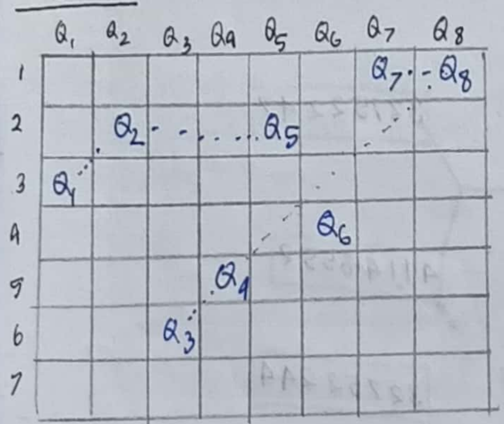
Total pair $= \frac{56}{2} = 28$

State 1 non attacking pair = Total Pair − Attacking Pair

$= 28 - 4$

$= 24$

State :2

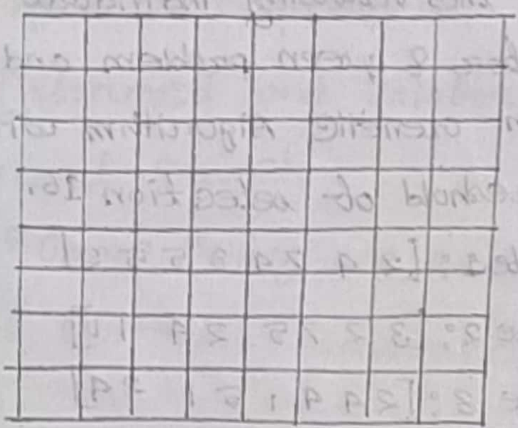


state 2 non attacking pair = Total Pair − Attacking Pair

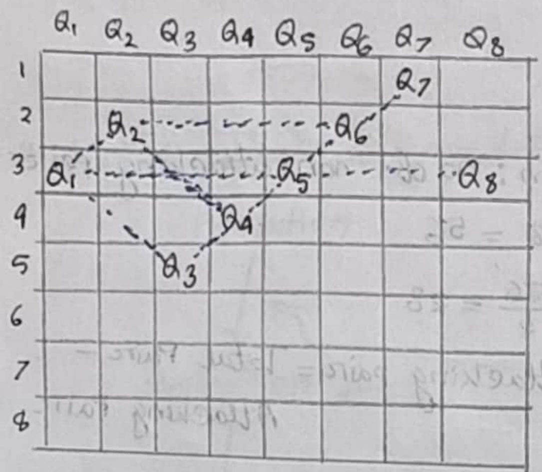
$= 28-5$

$= 23$

State 3:



State : 4



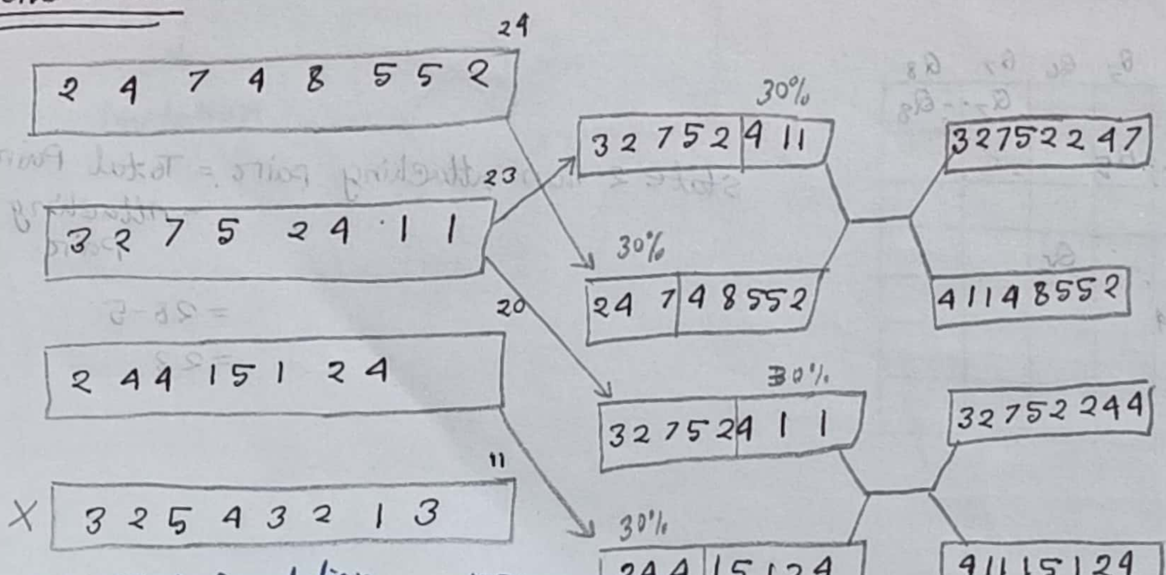
state 4 non attaching Pair

$= \text{Total Pair} - \text{Attaching pair}$

$= 28 - 17$

$= 11$

Solve another



(a) Initial Population b) Fitness Function

3 2 7 5 2 2 $\boxed{5}$ 7

4 1 1 $\boxed{5}$ 8 5 5 2

3 2 $\boxed{8}$ 5 2 2 4 4

4 $\boxed{2}$ 1 1 5 1 2 4

(e) Mutation

bag of words = { Python, Programming, like, I, EWU }

Python ↓ 0, Programming ↓ 1, like ↓ 2, I ↓ 3, EWU ↓ 4

Max Correct sequence = [3, 2 1 0]

State 1: [3 1 0 0] +1+0+0+1 = 2

State 2: [2 3 1 1] +1

Show GA for 1 iteration where fitness function is correct word position

#

Chapter - 5

Game playing and Adversarial Search

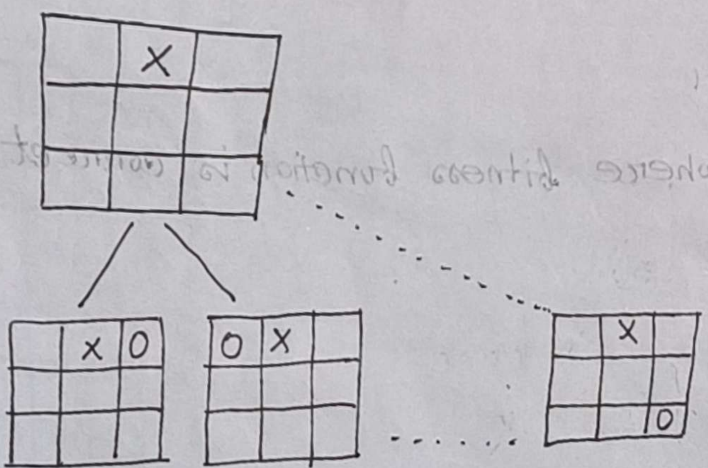
two player   Max gain   Min gain

	Deterministic	chance
Perfect information	Chess, checker, go	monopoly
Imperfect information	battle ship	bridge, poker UNO.

Game as search Problem

* Two player: MAX and
* Formal Definition
  * Initial state
  * Players (s)
  * Action (s)
  * Result
  * Successor function
  * Terminal test
  * Utility / Evaluation function
    - Win : +1
    - loss : -1
    - Draw : 0

X → MAX
O → MIN

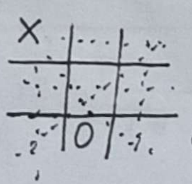


* Heuristic is —

$$E(n) = M(n) - O(n)$$

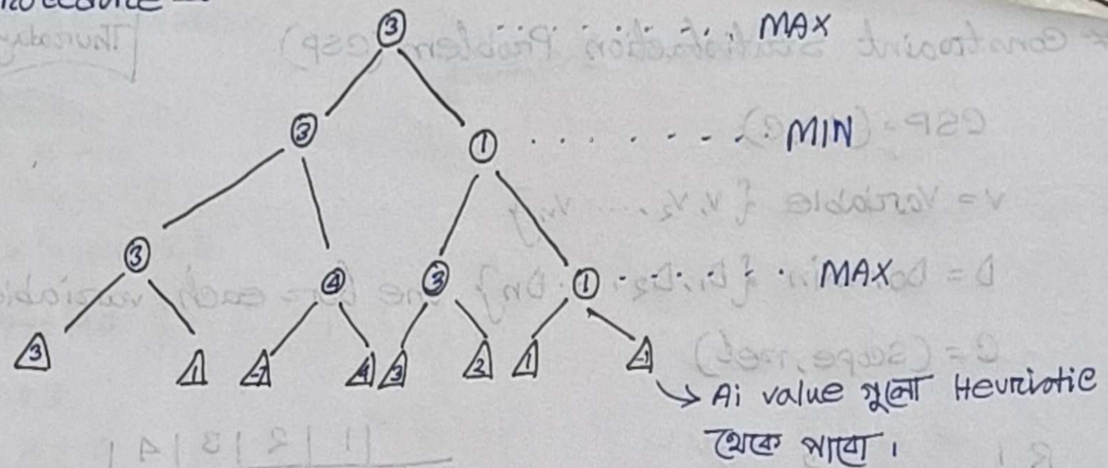
where,

M(n) is the total of any possible winning lines.
O(n) is total of opponent's possible winning lines.



$M(n) = 6$
$O(n) = 5$
$E(n) = 6 - 5 = 1$

* MIN MAX Procedure —



Alpha Beta Prunning –

→ α = value of best choice so far for MAX (highest-value)

→ β = Value of best choice so far for MIN (Lowest-value)

Prune when,

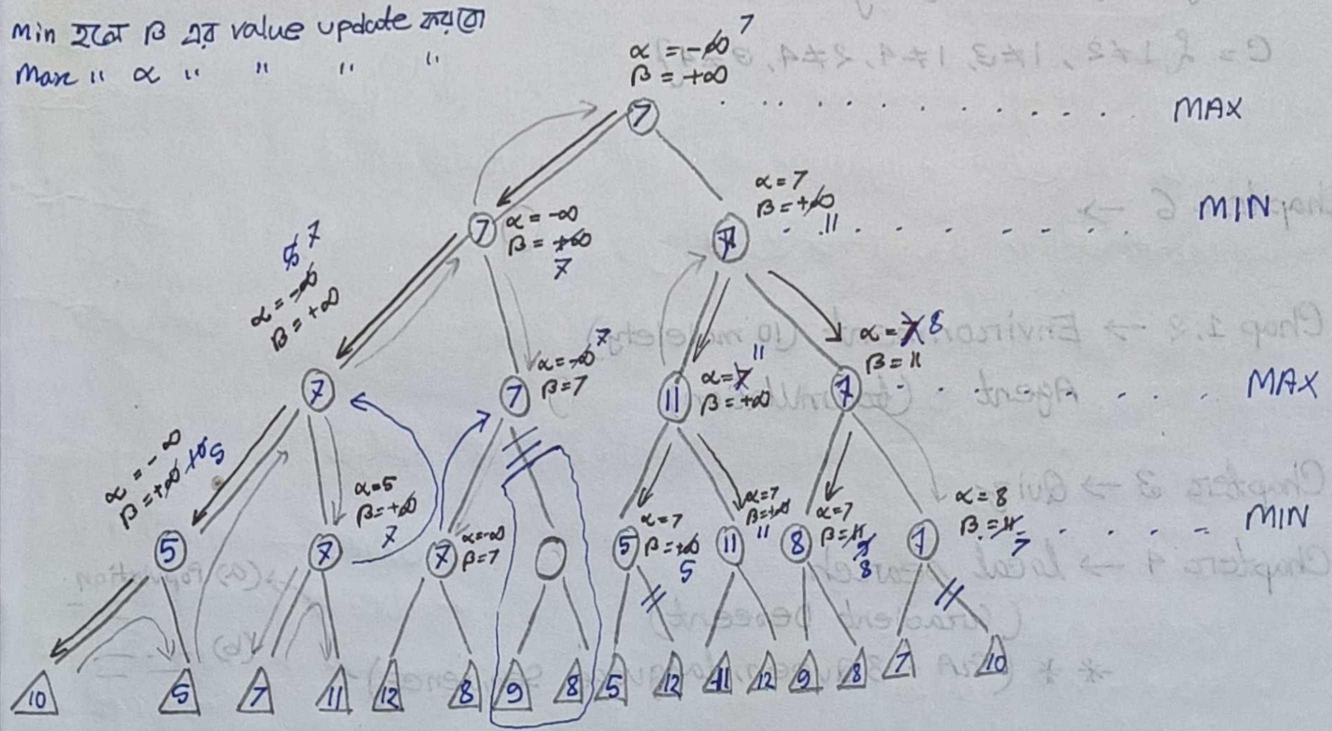
$\alpha \geq \beta$

Initial $\alpha = -\infty$

$\beta = +\infty$

Min হলে β এর value update করবো

Max " α " " " "



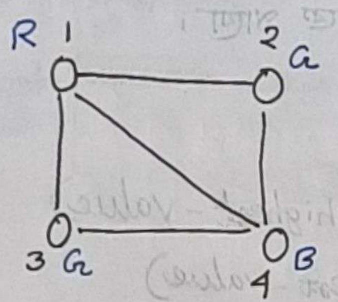
Constraint Statisfaction Problem (CSP)

CSP = (V, D, C)

V = Variable $\{V_1, V_2, \ldots V_n\}$

D = Domain $\{D_1, D_2, \ldots D_n\}$ one for each variable

C = (Scope, rel)



	1	2	3	4
Initial Domain	RGB	RGB	RGB	RGB
1 = R	R	GB	GB	GB
2 = G	R	G	B	B
3 = B	R	G	B	B
3 = G	R	G	G	B
4 = B	R	G	G	B

V = {1, 2, 3, 4}

D = {Red, Green, Blue}

C = {1 ≠ 2, 1 ≠ 3, 1 ≠ 4, 2 ≠ 4, 3 ≠ 4}

Chapter 6 →

Chap 1, 2 → Environment (10 modelety)
Agent - (formulation)

Chapter 3 → Quiz

Chapter 4 → local search
(Gradient Descent)
** (GA - 8 Queen, language sequence) → (a) Popula
(b) _ _

Chapter 5 → MIN MAX, α-β prune } Tic-Tac-Toe, Heuristic

Chapter 6 → Graph coloring problem,

Q → 6.1, 6.2, 6.3, 6.4, 6.7, 6.8, 6.9

CamScanner

Quadratic Function

gradient decent

$\alpha = 0.03$

initial $x = 10$

$f(x) = x^2 + 10x + 25$

$f'(x) = 2x + 10$

$\alpha = 0.03$

initial $x = 10$

1st iteration:

$x = x - \alpha f'(x)$

$= 10 - 0.03 \times 30$

$= 10 - 0.9$

$= 9.1$

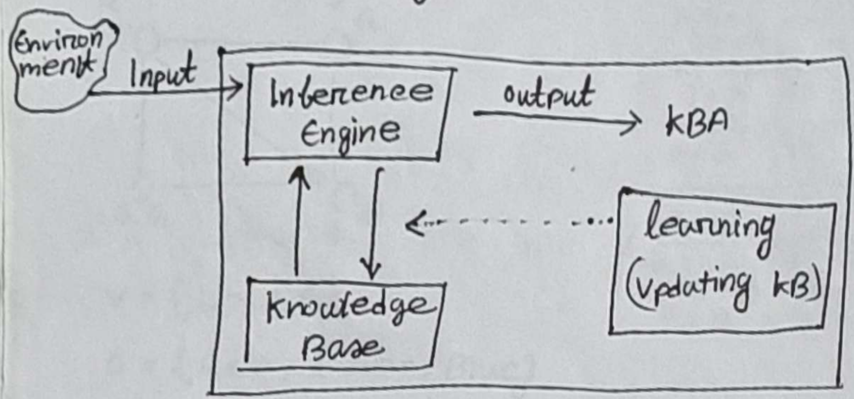
25.03.24/CSE364 Monday/ Cb-

Final

Lecture-9

Knowledge Base Agent in AI

* knowledge-base
* Inference Engine



Operation
- TELL
- ASK
- PERFORM

→ Knowledge level → Task
→ logical level → Inference
→ Implementation level → Code

* Declarative
* Procedural

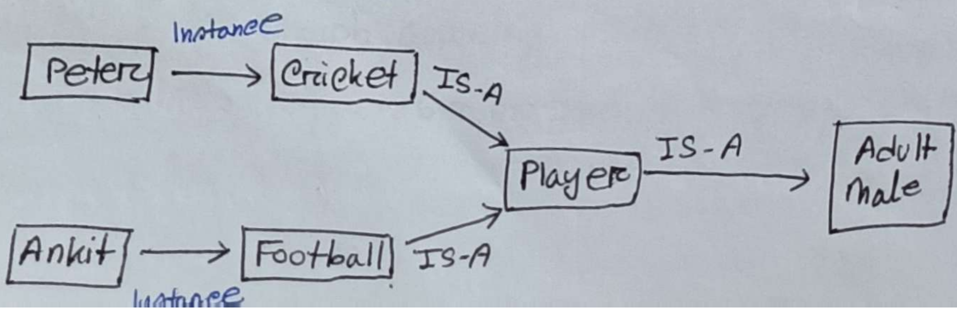
Knowledge Representation –

* Object
* Event
* Performance
* Meta-knowledge
* Fact

* Simple Relational knowledge

* DBMS → relational
* OOP → Inheritable
* Logical → Proposition, Predicate
* Ruled Based → Control Statement

Inheritable knowledge –



Inferential knowledge:

Alve is footballer

$\Rightarrow$ Footballer (Alve)

All footballers are athletes

$\Rightarrow \forall x:$ footballer $(x) \longrightarrow$ Athletes (x)

Procedural knowledge:

If:
else:
} Rules

Techniques:

Logical representation:

Syntax

Semantics

Propositional logic:

Predicate logic:

27.03.24 / CSE 3
Wednesday / Cls

Knowledge Base Agent (KBA)

English
↓
Propositional logic
↓ Quntifier
Predicate

[Basic logical ~~statement~~ statement
[Propositional function

01.04.24 / CSE 3
Monday / Cls-

Logic Representation:
* Syntax
* Semanties
} Rules.

* Propositional logic
Predicate logic

English Statement → Propositional logic statement

* Predicate logic

Universal → ∀ set of ∧ operation

~~Estiential~~ Existentfal → ∃ set of ∨ operation

All student like CSE 366

$\forall x$: likes CSE366 (x)

Some students hate CSE 366

$\exists x$: hate CSE366 (x)

All students likes few CSE course

$\forall x, y$: likes (x, y)

Variable - P, q - Value: True,

Operatop → logic operator
$\wedge, \vee, \neg, \rightarrow, \leftrightarrow$

If $\underbrace{\text{Tom is a eat}}_{P}$ only if it $\underbrace{\text{likes fish}}_{q}$

$P \leftrightarrow q$

$\exists x,y$: like CSE $(x,y) \wedge$ hate GED (x,y)

Inference Rules -

* Modus Ponens:

It is a deductive inference rule in which if A implies B and A is true, then B must also be true.

For example,

$\underbrace{\text{If it is raining}}_{A}$, then $\underbrace{\text{the ground is wet.}}_{B}$

$A \longrightarrow B$

if A is true, than B is true.

Symbolic Notation:

$P \longrightarrow Q\ ,\ P \vdash Q$

The symbol "$\vdash$" means "entails" or "leads to", and is used to indicate that Q can be deduced from the premises $(P \rightarrow Q)$ and P.

* Modus Tollens:

It is a deductive inference rule in which if A implies B is false, then A must also be false.

Example: If it is raining, then the ground is wet.
If it is not raining, the ground is not wet

Symbolic Notation:

$P \longrightarrow Q\ ,\ \neg P \vdash \neg Q$

17.04.24 Wednesday Cls

KR
→ Representation
→ Inference

OOP
DBMS
Rule
logic — Propositional / Predicate

Hypothetical Syllogism-

If A implies B and B implie C then A implies C

$A \rightarrow B$, $B \rightarrow C$

$A \rightarrow C$

Disjunctive Syllogism-

If it is either A or B and A is false then B must be true.

For example: It is either raining or snowing (A ∨

It is not raining ($\neg A$)

Therefore, It is snowing

Addition:

If A is true, then A or B is also true.

$A \vdash (A \vee B)$

Entails (Leads to)

Simplification:

If A and B are true, then A is true

$(A \wedge B) \vdash A$

Resolution.

If A implies B and C implies not B,
then A implies not C

$A \to B$
$C \to \neg B$

$A \to \neg C$

If $\underbrace{\text{it is raining}}_{A}$ $\xrightarrow{\text{then}}$ $\underbrace{\text{the ground is wet}}_{B}$

If $\underbrace{\text{the ground is dry}}_{C}$, $\xrightarrow{\text{then}}$ it is not ~~raining~~ wet. $\neg B$

If it is raining then the ground is not dry
$A \longrightarrow \neg C$

And Elimination Rule -

$(A \land B \land C)$ true

A
B
C } true.

Wampus World in Ai -

Stench		Breeze	
Wampus	Breeze Gold Stench	PIT	Breeze
Stench		Breeze	
Agent	Breeze	PIT	Breeze

	1	2	3	4
4	[1,4]	[2,4] P?	[3,4]	[4,4]
3	[1,3] W_{13} W?	[2,3] G_{23} S, G, B	[3,3] P_{33}	[4,3]
2	[1,2]	[2,2] P?	[3,2]	[4,2]
1	[1,1] $\boxed{A}$	[2,1] ✓	[3,1] P?	[4,1]

$\boxed{A}$ → Agent   B → Breeze
$\boxed{G}$ → Gold
ok → Safe
P → Pit
S → Stench
V → Visited
W → Wampus

Atomic Proposition variable -

* Let $P_{i,j}$ be true if there is a PIT in the room [i,j]

* B_{ij} true rooom [i,j]

  W_{ij} true room [i,j]

  S_{ij} " "

  V_{ij} " " "

  Ok_{ij} " " Safe

Some propositional Rule -

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

$R_2 : \neg S_{21} \rightarrow \neg W_{11} \wedge \neg W_{21} \wedge \neg W_{22} \wedge W_{31}$

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

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

Proof that wampus is in the room (1,3)

Solution

~~$\neg S_1$~~

Apply modus Ponent with $\neg S_{11}$ and R_1

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

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

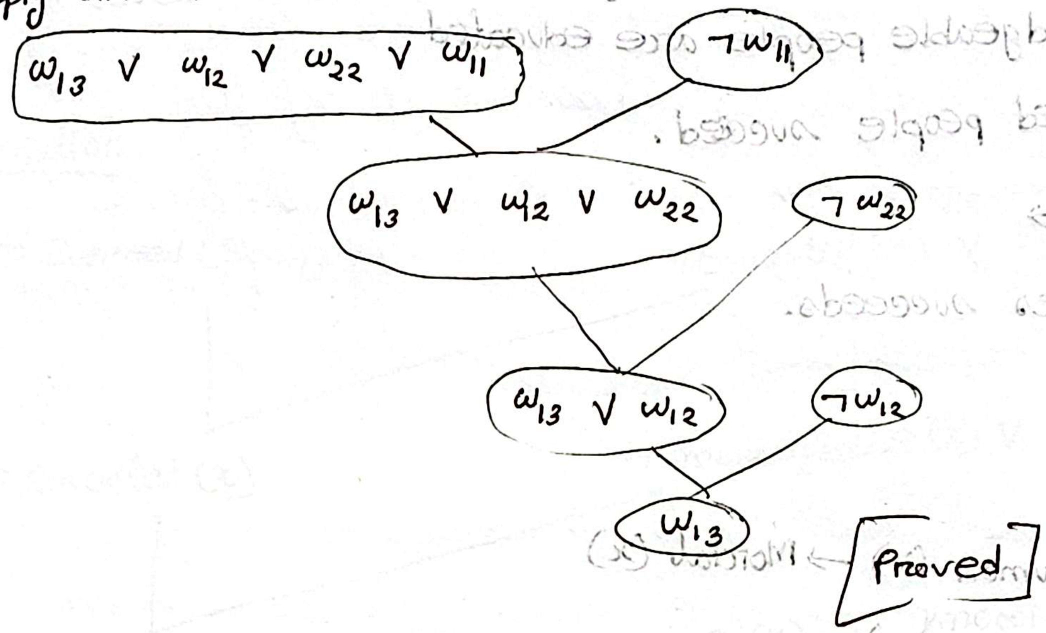
Apply And Elimination, we will get there statement: $\neg W_{11}$, $\neg W_{12}$ a

Apply modus ponent to $\neg S_{21}$ and R_2: $\neg W_{21} \wedge \neg W_{22} \wedge \neg W_{31}$ ^

And elimination: $\neg W_{11}$, $\neg W_{21}$, $\neg W_{22}$, $\neg W_{31}$

Apply modus ponent to S_{12} and R_4: $W_{13} \vee W_{12} \vee W_{22} \vee W_{11}$

Apply unit Resolution -



22.04.28 Monday / CSE 366 Cls-

29.04.24 Monday / CSE 366 Cls-

Resolution in FOL:

1. Conversion of facts into FOL
2. Convert FOL into CNF (Conjunctive Normal Form)
3. Negate the statement which needs to prove (Proof by contradi-ction)
4. Draw resolution graph (Unification)

knowledge Base:

1. All humans are mortal
2. Socrates is a human
3. Some philosophers are human
4. If some one is philosopher, they are wise
5. Socrates is a philosopher

6. Wise individuals are knowledgeable.

7. All knowledgeable people are educated

8. All educated people succeed.

Proof that →

Socrates succeeds.

Solution:

FOL:

1. $\forall x : human(x) \rightarrow Mortal(x)$
2. $human(Socrates)$
3. $\exists x : Philosopher(x) \wedge human(x)$
4. $\forall x : Philosopher(x) \rightarrow wise(x)$
5. $Philosopher(Socrates)$
6. $\forall x : wise(x) \rightarrow knowledgeable(x)$
7. $\forall x : knowledgeable(x) \longrightarrow Educated(x)$
8. $\forall x : Educated(x) \rightarrow Succeed(x)$

Goal:

$succeed(Socrates)$

* Conversion to CNF

1. $\neg human(x) \vee Mortal(x)$
2. $human(Socrates)$
3. $Philosopher(x)$
4. $human(x)$
5. $\neg Philosopher(x) \vee wise(x)$
6. $Philosopher(Socrates)$
7. $\neg wise(x) \vee knowledgeable(x)$
8. $\neg knowledgeable(x) \vee Educated(x)$

9. ¬Educated (x) ∨ succeed (x)

Resolution

¬Succeed (Socrates)　　　¬Educated (x) ∨ Succeed (x)

¬Educated (x)　　　¬knowledgeable (x) ∨ Educated (x)

¬knowledgeable (x)　　　¬wise (x) ∨ knowledgeable (x)

¬wise　　　¬philosopher (x) ∨ wise(x)

¬philosopher　　　philosopher (Socrates)

{}

Inference engine -

* Forward Chaining
* Backward Chaining

Motivation
Methodology
Result
Discussion

wumpus

* The law says that it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy of America has some missiles and all of its missiles were sold it by Colonel West, who is American. Prove that Colonel West is a Criminal.

Solve

* It is a crime for an American to sell weapons to hostile nations.

R1: $\forall x, y, z$: American $(x) \wedge$ weapon $(y) \wedge$ ~~Hostile (z) ⇒ A~~ Sell $(x, y, z) \wedge$ Hostile $(z) \Rightarrow$ Criminal (x)

* Nono has some missiles.

R2: $\exists x$ Owns (Nono, x) $\wedge$ Missile (x)

R2: Owns (Nono, x)

R3: Missile (x)

* .... all of its missiles were sold it by Cornel West.

R4: Missile $(x) \wedge$ Owns (Nono, x) $\Rightarrow$ Sells (West, x, Nono)

* Misiles are weapons.

R5: Missiles $(x) \Rightarrow$ weapon (x)

* An enemy of America counts as "hostile": ~~R6~~.

R6: Enemy (x, America) $\Rightarrow$ Hostile (x)

* West, who is American . . .

R7: American (West)

* The country Nono, an enemy of America

R8: Enemy (Nono, America)

Goal: Criminal (West)

CNF Form:

R_1: ¬American(x) ∨ ¬weapon(y) ∨ ¬Sells(x,y,z) ∨ ¬Hostile(z) ∨ Criminal(x)

R_2: Owns(Nono, x)

R_3: Missile(x)

R_4: ¬Missile(x) ∨ ¬Owns(Nono, x) ∨ Sells(West, x, Nono)

R_5: ¬Missile(x) ∨ weapon(x)

R_6: ¬Enemy(x, America) ∨ Hostile(x)

R_7: American(West)

R_8: Enemy(Nono, America)

Resolution graph

¬American(x) ∨ ¬weapon(y) ∨ ¬sells(x,y,z) ∨ ¬Hostile(z) ∨ Criminal(x)   ¬Criminal(west)

¬American(x) ∨ ¬weapon(y) ∨ ¬sell(x,y,z) ∨ ¬Hostile(z)   American(west)

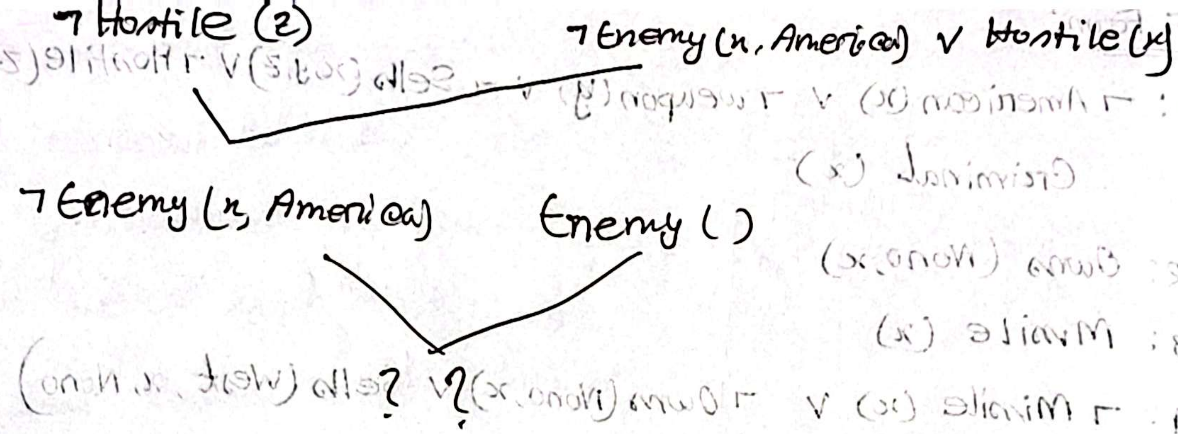
¬weapon(y) ∨ ¬sell(x,y,z) ∨ ¬Hostile(z)   ¬Missile(x) ∨ Weapon(x)

¬Sell(x,y,z) ∨ ¬Hostile(z) ∨ ¬Missile(x)   Missile(x)

¬Sell(x,y,z) ∨ ¬Hostile(z)   ¬Missile(x) ∨ ¬Owns(Nono,x) ∨ Sells(west, x, Nono)

¬Hostile(z) ∨ ¬Missile(x) ∨ ¬Own(Nono, x)   Missile(x)

¬Hostile(z) ∨ ¬Owns(Nono, x)   Owns(Nono, x)



Inference Engine:

* Forward Chain
* Backward Chain.

Propositional logic

$P \Rightarrow Q$

$L \wedge M \Rightarrow P$

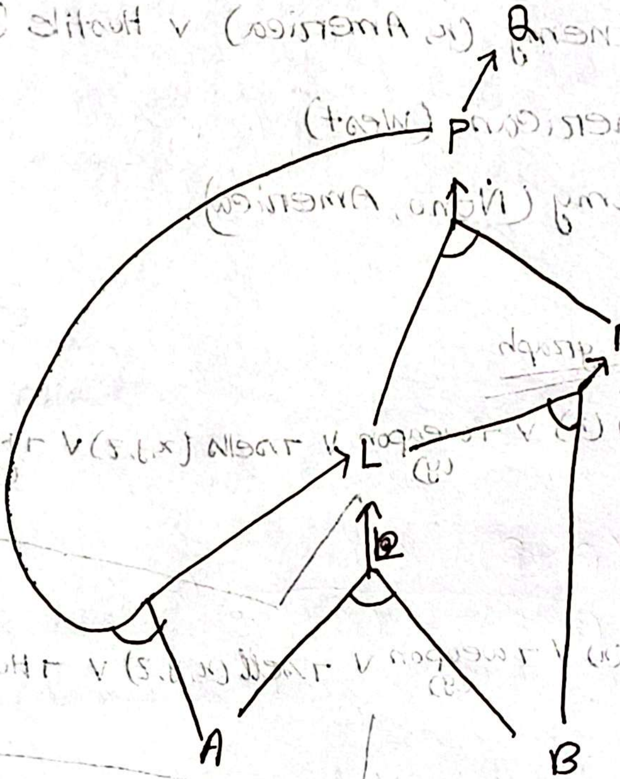
$B \wedge L \Rightarrow M$

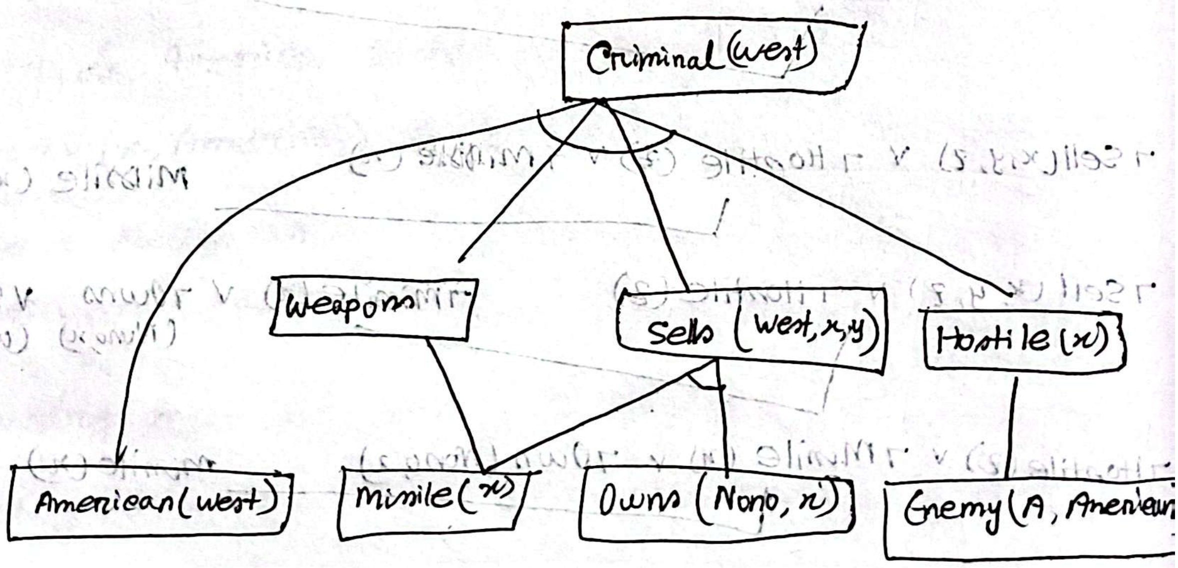
$A \wedge P \Rightarrow L$

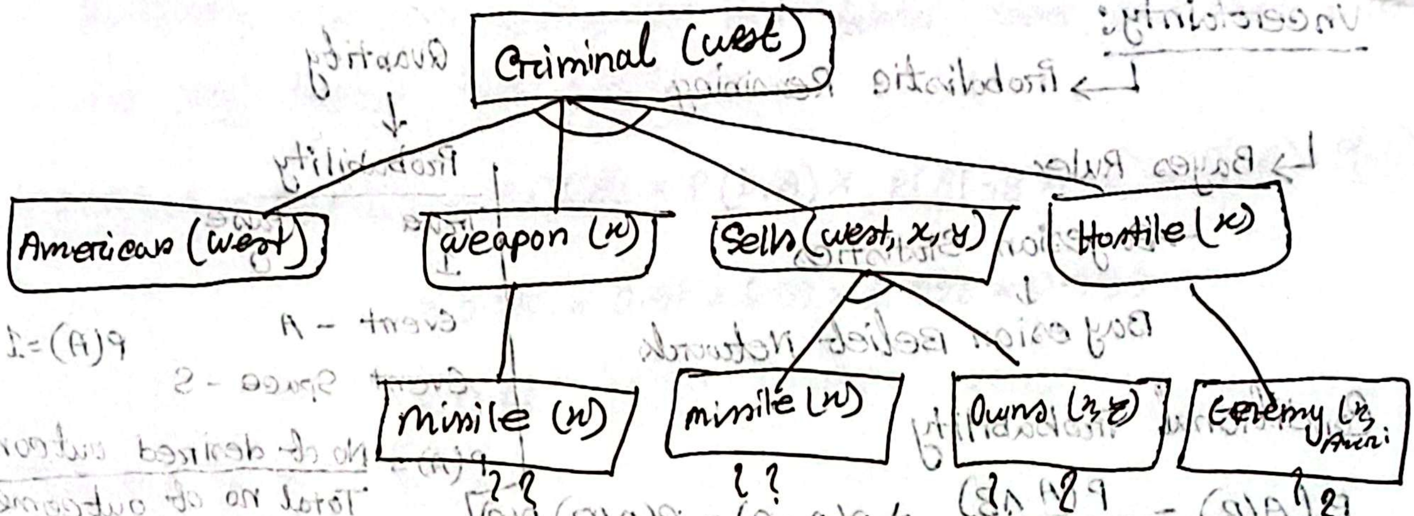
$A \wedge B \Rightarrow L$

A

B







A → Direct line

And थरमता A हटा।

Probabilistic Reasoning in AI

Uncertainty:

↳ Probalistic Reasining

↳ Bayes Rule

↳ Bayesian Statistics

↓

Bayesian Belief Network

Quantify

↓

Probability

True	False
1	0

Event - A $\quad P(A)=1$

Event Space - S

$$P(A) = \frac{\text{No of desired outcome}}{\text{Total no of outcome}}$$

$$\Rightarrow P(A) + P(\neg A) = 1$$

Conditional Probability

$$P(A/B) = \frac{P(A \wedge B)}{P(B)} \Rightarrow P(A \wedge B) = P(A/B)\, P(B)$$

$$P(B/A) = \frac{P(A \wedge B)}{P(A)} \Rightarrow P(A \wedge B) = P(B/A)\, P(A)$$

$$\Rightarrow P(A/B)\, P(B) = P(B/A)\, P(A)$$

Bays Rules

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

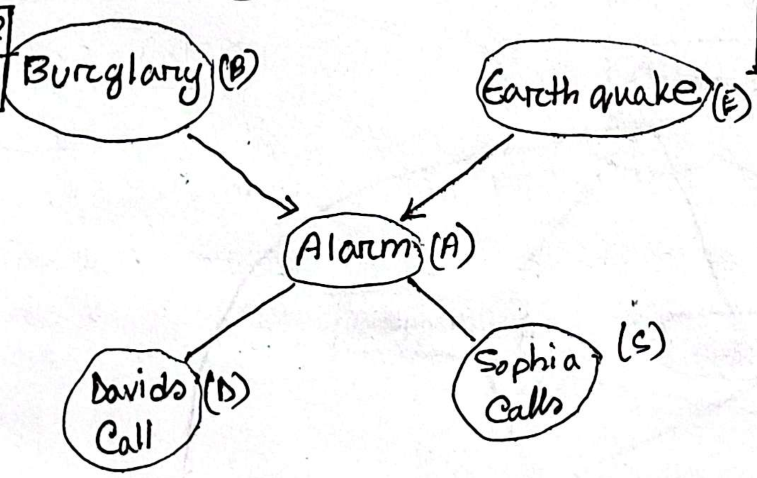
$A = [A_1, A_2 \cdot \ldots A_i]$ $\quad$ So, $P(A_i/B) = \dfrac{P(A_i) \times P(B/A_i)}{\sum_{i=1}^{k} P(A_i) \times P(B/A_i)}$

Set of mutually exclusive.

Bayesion Network:

T	0.02
F	0.998

T	0.001
F	0.999



B	E	P(A=T)	P(A=F)
T	T	0.94	0.06
T	F	0.95	0.04
F	T	0.69	0.69
F	F	0.999	0.999

A	P(D=T)	P(D=F)
T	0.91	0.09
F	0.05	0.95

A	P(S=T)	P(S=F)
T	0.75	0.25
F	0.02	0.98

* Determine the likelihood that the alarm went off but neigher a burglary nor an earthquake had taken place, and that both David and Sophia had phoned.

$$P(S, D, A, \neg B, \neg E) = P(S|A) \times P(D|A) \times P(A|\neg B \wedge \neg E) \times P(\neg B) \times P(\neg E)$$

$$= 0.75 \times 0.91 \times 0.99 \times 0.998 \times 0.999$$

$$= 0.68$$

15.05.24

AI - Topics

Searching

KB → Representation

KB → Inferencing

KB → Prob. reasoning → Learning

* INferencing
* Propositional logic wampus
* Resolution → FOL
* Backward and Forward chain → FOL
* Unification
* Prob. reasoning
* Belief Network

HMM (Hidden Markov Model)

* Math
* Reinforcement learning

Stochastic Process Model → Discrete → D. Time → Memory less
→ C Time
→ Continuous.

* Markov and Markov Hidden Model

handle Data which can be represented as 'sequence' of Observation over time.

Markov two Assumption.

1. Limited Horizon Assumption

2. Prob. state at a time (t) depends only on the state at the time (t-1).

$$P(Z_t, Z_{t-1}, Z_{t-2}, \ldots, Z_1) = P\left(\frac{Z_t}{Z_{t-1}}\right)$$

2. Stationary Process Assumption:

$$P(Z_t \mid Z_{t-1}) = P(Z_t \mid Z_{t-1}, Z_{t-2}, \ldots Z_1, Z_0)$$

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

Markov Models

* Set of States: $\{S_1, S_2, \ldots, S_n\}$

* Process moves from one state to another generating a sequence of states: $S_{i1}, S_{i2}, \ldots, S_{ix} \ldots$

* Markov chain property: Prob. of each subsequent state depends only on what was the prev state:

$$P(s_{i_k} \mid s_{i_1}, s_{i_2}, \ldots, s_{i_{k-1}}) = P(s_{i_k} \mid s_{i_{k-1}})$$

* To define Markov model, the following prob have to be specified :

Transition matrix

	S_0	S_{hot}	S_{cold}	S_{rain}
S_0		0.33	0.33	0.34
S_{hot}	0	0.8	0.1	0.1
S_{cold}	0	0.2	0.6	0.2
S_{rain}	0	0.1	0.2	0.7

$P(S_{hot} / S_0) = 0.33$

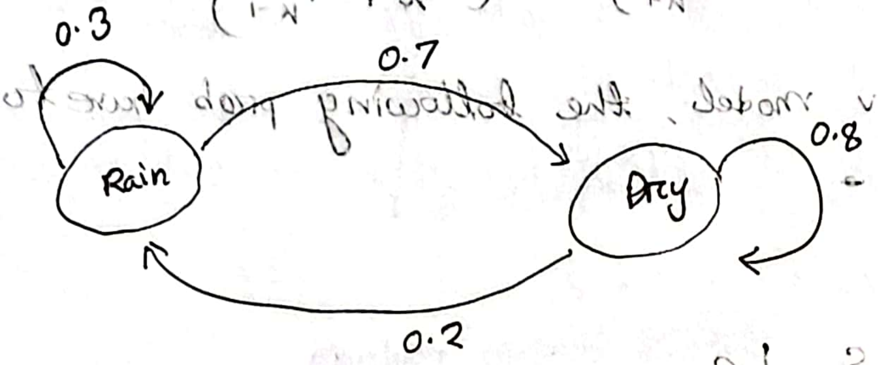
$P(S_{cold} / S_{hot}) = 0.1$

Transition Prob. $a_{ij} = P(S_i \mid S_j)$

initial Prob, $\pi_i = P(S_i)$

Markov Models

	System state is fully observable	System state is partially observable
System autonomus	Markov Chain	Hidden Markov model
Controlled	Markov Decision process	Partially observable Markov dec. process



* Two states: Rain and Dry

	Rain	Dry
Rain	0.3	0.7
Dry	0.2	0.8

P (Rain/Rain) = 0.3

* Initial Prob

P(Rain) = 0.4
P(Dry) = 0.6

Calculation of sequence Prob:

Markov chain Property

$$P(s_{i_1}, s_{i_2}, \ldots, s_{i_k}) = P(s_{i_k} \mid s_{i_1}, \ldots, s_{i_{k-1}})\, P(s_{i_1}, \ldots, s_{i_{k-1}})$$

$$= P(s_{i_k} \mid s_{i_{k-1}})\, P(s_{i_1}, s_{i_2}, \ldots, s_{i_{k-1}})$$

$$= P(s_{i_k} \mid s_{i_{k-1}})\, P(s_{i_{k-1}} \mid s_{i_{k-2}}) \ldots$$

$$\ldots P(s_{i_2} \mid s_{i_1})\, P(s_{i_1})$$

Suppose we want to calculate a prob. of a sequence of states.

$\{Dry, Dry, Rain, Rain\}$

$P(\{Dry, Dry, Rain, Rain\})$

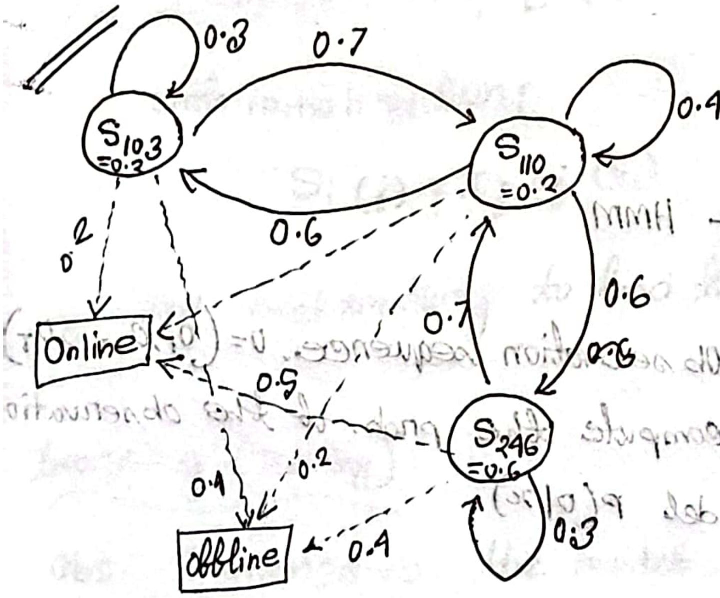
$\cancel{= P(Dry)\ P(Dry \mid Dry)}$

$= P(Rain \mid Rain)\ P(Rain \mid Dry)\ P(Dry \mid Dry)\ P(Dry)$

$= 0.3 \times 0.2 \times 0.8 \times 0.6$

$= 0.0288$

20.05.24 / CSE366



Find the sequence Prob.

$\{S_{103}, S_{103}, S_{110}, S_{110}, S_{246}\}$

Hidden Markov Model –

Components :

States : $\{S_1, S_2, \ldots, S_n\}$

Observation : $V = \{V_1, V_2, \ldots, V_m\}$

Transition Prob : State i to state j,

$A = \{a_{ij}\}$, where :

$a_{ij} = P(S_{k+1} = j \mid S_k = i)$

i/j	$S_{\omega 3}$	

Emission Prob: The Prob of observing V_K from state j is given by

$B = \{ b_j(k) \}$

$b_j(k) = P(O_t = V_k \mid S_t = j)$

Initial State Prob:

$\pi = \{\pi_i\}$

$\pi_i = P(S_1 = i)$

Three Fundamental Problems of HMM

1. Evaluation (likelihood): Observation sequence, $O = (O_1, O_2, \ldots, O_T)$

The model $\{ \lambda = (A, B, \pi)$, compute the prob. of the observation sequence given the model $P(O|\lambda)$.

Forward Algorithm:

$$\alpha_t(j) = P(O_1, O_2, \ldots . O_t, S_t = j \mid \lambda)$$

with the recursive relation:

$$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) \, a_{ij} \right] b_j(O_t)$$

where the initialization is

$$\alpha_1(j) = \pi_j \, b_j(O_1)$$

$$P(O|\lambda) = \sum_{j=1}^{N} \alpha_T(j)$$

The fundamental Problems of HMM

Decoding (most likely sequence)

Observation sequence O and the model λ, find the most likely sequence of states that produced the observation.

Viterbi Algorithm:

$$\delta_t(j) = \max_i \left[\delta_{t-1}(i) \cdot a_{ij}\right] \cdot b_j(O_t)$$

with initialization:

$$\delta_1(j) = \pi_j \, b_j(O_1)$$

and backtracking to find the state sequence.

Learning (Training)

Obs. Sequence = O the model, $\lambda = (A, B, \pi)$ to maximize $P(O/\lambda)$

~~Because~~ Baum-welch Algorithm, an Expectation-Maximization (EM) Algorithm.

Expectation step: forward ($\alpha_t(i)$ and backward $\beta_t(i)$)

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \ldots\ldots O_T \mid S_t = i, \lambda)$$

with the recursive relation:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} \, b_j(O_{t+1}) \, \beta_{t+1}(j)$$

Re-estimate model parameters, A, B, π using the α and β variables.

---

* Hidden state : S_1 : Sunny S_2 : Rainy

Observation : O_1 : Dry Ground , O_2 : wet ground

O_3 : Clouds O_4 : People carrying umbrella

Parameters:

Initial State Prob. $\pi = [\pi_1, \pi_2] = [0.6, 0.4]$

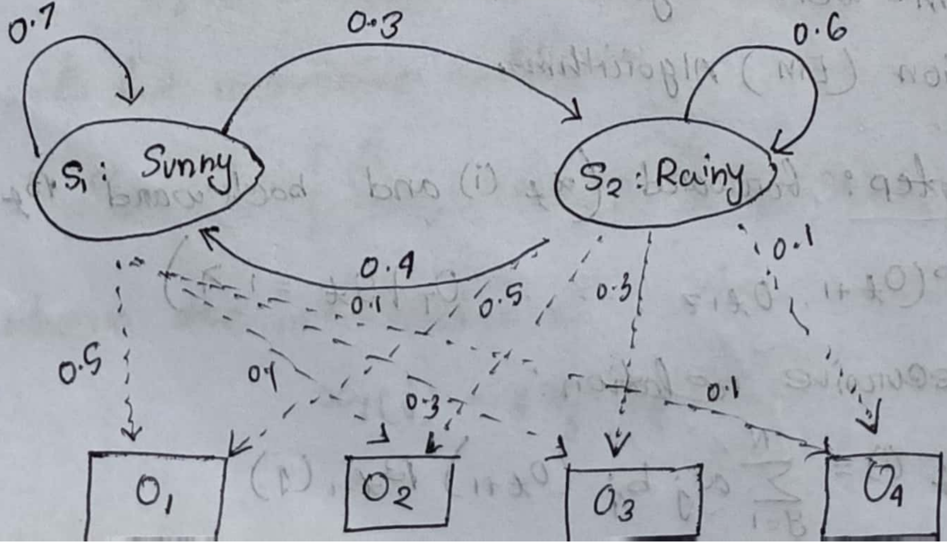
Transition Prob :

$$A_{ij} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix}$$

Emission Prob :

$$B = \begin{bmatrix} b_1(O_1), & b_1(O_2) & b_1(O_3) & b_1(O_4) \\ b_2(O_1) & b_2(O_2) & b_2(O_3) & b_2(O_4) \end{bmatrix}$$

$$= \begin{bmatrix} 0.5 & 0.1 & 0.3 & 0.1 \\ 0.1 & 0.5 & 0.3 & 0.1 \end{bmatrix}$$



objective:

$\delta_t(j) = \max_i [\delta_{t-1}(i) a_{ij}] b_j(t$

$\delta_1(j) = \pi_j b_j(O_1)$

1. Initialization: $\delta_1(i) = \pi_i \, b_i(O_1)$

For $O_1 = O_3$ (clounds), $\delta_1(1) = 0.6 \times 0.3$
$= 0.18$ (Sunny)

$\delta_1(2) = 0.4 \times 0.3 = 0.12$ (Rainy)

2. Recursion:

$$\delta_t(j) = \max_i [\delta_{t-1}(i) a_{ij}] . b_j(O_t)$$

For people carrying umbrella O_4.

$$\delta_2(1) = \max \begin{cases} \delta_1(1) \times a_{11} \times b_1(O_4) \\ \delta_1(2) \times a_{21} \times b_1(O_4) \end{cases}$$

$$= \max \begin{cases} 0.18 \times 0.7 \times 0.1 \\ 0.12 \times 0.4 \times 0.1 \end{cases}$$

$$= \max \begin{cases} 0.0126 \\ 0.0048 \end{cases}$$

$= 0.0126$ (Sunny)

For wet ground O_2.

$$\delta_3(1) = \max \begin{cases} \delta_2(1) \times \cancel{a_{11} \times b_1(O_2)} \\ \delta \end{cases}$$

$$\delta_2(2) = \max \begin{cases} \delta_1(1) \times a_{12} \times b_2(O_4) \\ \delta_2(2) \times a_{22} \times b_2(O_4) \end{cases}$$

$$= \max \begin{cases} 0.18 \times 0.3 \times 0.1 \\ 0.12 \times 0.6 \times 0.1 \end{cases}$$

$= 0.0072$ (Rainy)

Objective

Given the sequence of observation O (O_3, O_4, O_2)

Determine the most likely sequence of hidden states (weather condition) for wet ground (O_2)

$$\delta_3(1) = \max_i \begin{cases} \delta_2(1) \times a_{11} \times b_1(O_2) \\ \delta_2(2) \times a_{21} \times b_1(O_2) \end{cases}$$

$= 0.000882$ (sunny)

$$\delta_3(2) = \max \begin{cases} \delta_2(1) \times a_{12} \times b_2(O_2) \\ \delta_2(2) \times a_{22} \times b_2(O_2) \end{cases}$$

$= 0.00216$ (Rainy)

3. Termination:

$\max(\delta_3(1), \delta_3(2)) = 0.00216$ (Rainy)

4. Path Back tracking:

* From δ_3, we find the last state "Rainy"
* From δ_2, given $\delta_3(2)$ came from $\delta_2(2) \times a_{22}$. the 2nd state is "Rainy"
* From δ_1, given $\delta_2(2)$, came from $\delta_1(2)$, a_{21}; the 1st state is "Rainy"

Most likely seq (Rainy, Rainy, Rainy)

CamScanner