Date : ....
2078/07/28 Page: ....

Chapter-1

Goals in Problem Solving

Goal

- Target / Desired result / Accomplishment (what is to be achieved?)

Planning

- Set / Sequence of tasks to achieve a goal.

Types of planning

1) Linear planning
2) Non-linear planning

☆

Linear	Non-Linear
- Planning Algorithm that works on one goal until completely finished before moving on to next goal.	- Works to achieve goal by ~~alterna~~ interleaving (alternation of goals)
- Doesn't prioritize task.	- Considers priority
- Non-preemtive (doesn't swith)	- Preemtive
- Uses stack (FIFO) for goal achievement.	- Uses set for Goal achievement.
- Requires less storage / reduced search ~~page~~, space.	- Requires more storage / larger search ~~page~~ space

- Simple Algo.	- Complex Algo.
- May not produce optimal solⁿ	- May produce optimal solⁿ
- Incomplete	- Complete

☆ MEA (Means-End Analysis)

- Algorithm that works based on 3 parameters;
  1) Initial state / Current
  2) Final state / Goal
  3) Goal difference (Δ)
- Reducing goal difference between Initial state & final state. Until goal difference (Δ) is 0
- When, Δ = 0, Goal is achieved.

- Algorithm:

1) Define Initial state, goal state & calculate goal difference (Δ), ~~and, state~~
~~2) Define actions~~
2) Choose action / procedure with least goal difference that will ultimately reach to goal.
3) When goal difference (Δ) = 0, goal is reached.

2078/07/29

Date:
Page:

Production Rule System

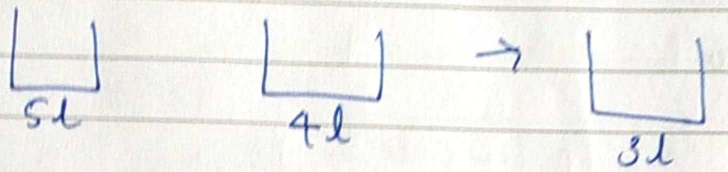
$\boxed{A \rightarrow B}$

If A, then B

↓ eg.

2078/08/01

Water Jug Problem



5l	4l
0	0
0	4
4	0
4	4
5	3

Path cost = 4

5l	4l
0	0
5	0
1	4
1	0
0	1
5	1
2	4
2	0
0	2
5	2
3	4

Path cost = 10

Date: ....
Page: ....

Farmer fox Goose Grain Problem

L = Left side of River
R = Right side of River

Possible actions:
- fa
- fa fo
- fa Go
- fa Gr

L: fa fo Go Gr • → Boat
R:

fa Go

L: fo Gr
R: fa Go •

fa

L: fa fo Gr •
R: Go

fa fo

L: Gr
R: fa fo Go •

fa Go

L: fa Go Gr •
R: fo

fa Gr

L: Go
R: fo fa Gr •

fa

L: fa Go •
R: Gr fo

fa Go

L:
R: fa fo Go Gr

fa Gr

L: fo
R: fa Gr Go •

fa Go

L: fo fa Go •
R: Gr

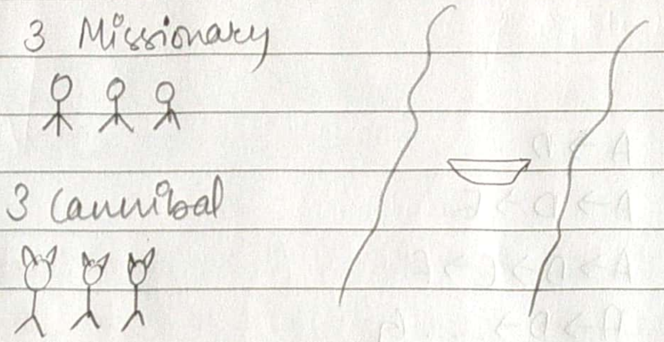
fa fo

L: Go
R: ~~Gr~~ fa fo •

Path Cost = 7
Total no. of States = 8

Date: ....
Page: ....

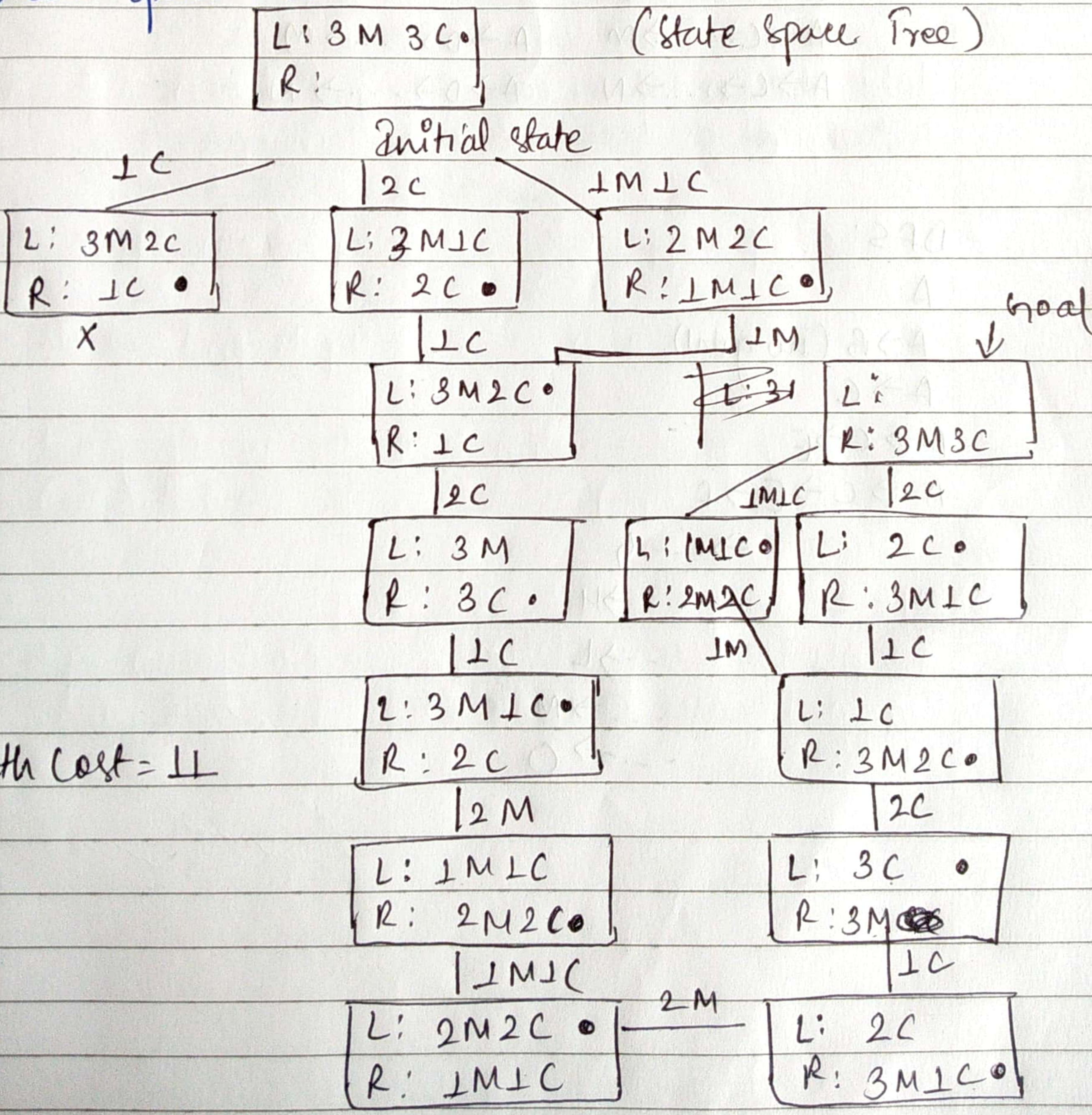
M – C problem (Missionary Cannibal Problem)



Possible Actions

1) 1M
2) 1C
3) 2M
4) 2C
5) 1M1C

a) Draw Space Tree (State Space Tree)



Path Cost = 11

Scanned with CamScanner

2078/08/05

Date: ....

Page: ....

b) Solve using BFS / DFS

BFS:

A

A→B (no goal)

A→C

A→C→E

A→C→E→F

A→C. . . →G

A→C. . . →L

A→C→. . . →M

A→C→. . . →N

A→D

A→D→E

A→D→E→F

A→D→. . . G

A→D→ . . . L

A→D→. . . →M

A→D→. . . →N

DFS:

A

A→B (no goal)

A→C

A→C→E

A→C→E→F

→. . . G

. →H

. . →L

. . . →M

. . . →O

~~2078/08/06~~

Date :
Page:

WI

Learning by Analogy

(Analogical reasoning / learning)

- → Tumor Problem
- → Castle Problem

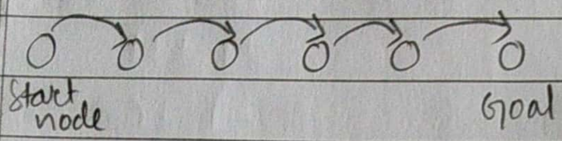
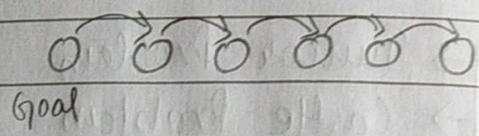
↳ Comparison of problems and applying solution from known problem to the new problem.

↳ 4 Re's

- → Retrieve : Accessing from database
- → Reuse : Mapping same solution
- → Revise : Modification to the soln
- → Retain : Store

Date:
Page:

2078/08/06

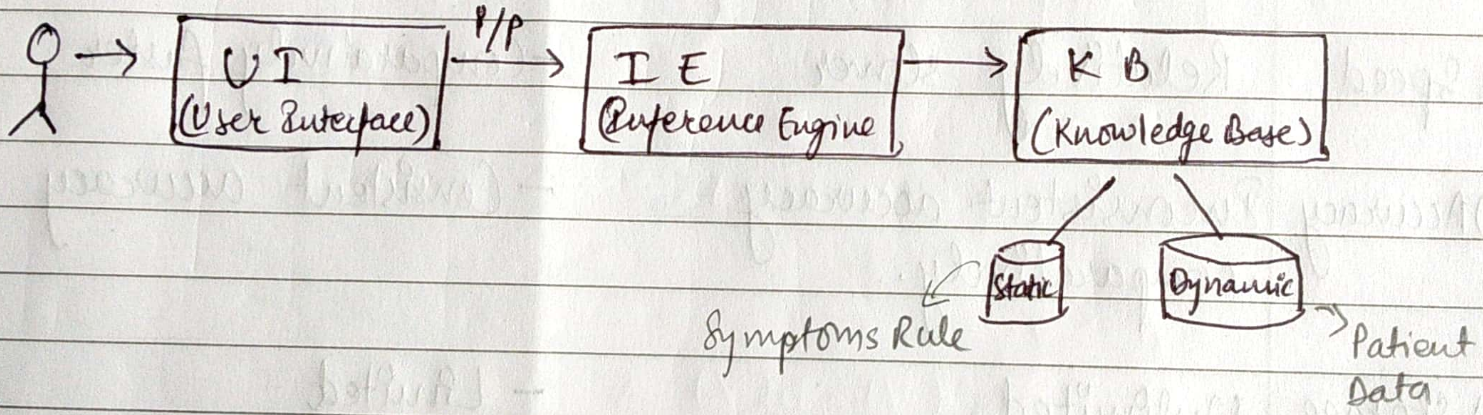
Forward Chaining	Backward Chaining
	
- Starts from initial state and aims for conclusion	- starts from goal and backward search to necessary conclusion.
- data driven inference technique	- goal driven inference technique
- Bottom-up approach	- Top-down approach
- Breadth first search	- Depth first search
- The no. of final solution is infinite. (multiple)	- The no. of final solution is limited. (one)
- eg: Diagnosis Recomendation in a Disease Prediction bots	- eg: Navigation system in Driverless car.

Mycin

↳ Expert system to identify bacterial infections and recommend antibiotics.

↳ 500 production rules
- If symptoms, then this disease
- If disease, then this antibiotics

eg: If ~~stain~~ strain of organism its negative and morphology its rod, then, suggest enterobacteriocease.



Date :
Page:

Chapter- 2

Intelligence

VVE
HI Vs MI

	Human Intelligence (Natural)	Machine Intelligence (Artificial)
1) Operational Ability	Performance declines ~~overtime.~~ over time.	- Multitasking with consistent performance.
2) Speed	Relatively slower	- Comparatively faster
3) Accuracy	Inconsistent accuracy comparatively.	- Consistent accuracy
4) Storage Capacity	Unlimited	- Limited
5) Decision making/ Reasoning	might be Based on ~~multip~~ emotions (Good abstract decisions)	- Unbiased & Rational
6) Perception	Perceives by pattern	- Perceives in terms of Datasets

Scanned with CamScanner

2078/08/08

Date:
Page:

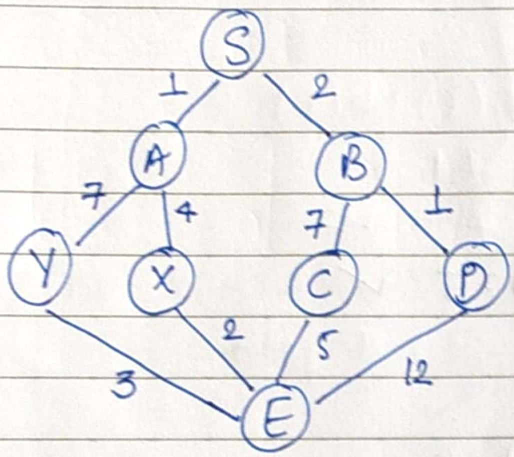
Chapter- 4

Inference & Reasoning

Search Techniques:

1) Blind search or Uninformed search — BFS, DFS
2) Heuristic search or Informed search — Greedy, A*

Q



Values for h (heuristic):

A	5
B	6
C	4
D	15
X	5
Y	8
E	0

(i) **Greedy search:**

$$\boxed{f(n) = h(n)}$$

Where, $f(n)$ = evaluation function
$h(n)$ = heuristic value of node 'n'
= estimated path cost from 'n' to goal

Scanned with CamScanner

Date :
Page:

- Start node : S

S → A

$f(n) = h(n)$
$f(A) = h(A) = 5$ ✓

S → B

$f(B) = h(B) = 6$

- Expanding S → A

S → A → X

$f(X) = h(X) = 5$ ✓

S → A → Y

$f(Y) = h(Y) = 8$

- Expanding S → A → X

S → A → X → E

$f(E) = h(E) = 0$ ✓

∴ S → A → X → E is the best path.
(Path cost = 7)

Date :
Page:

(ii) A* Search :

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

Where,

$g(n)$ = Path cost to reach node 'n'

S → A

$$f(A) = g(A) + h(A)$$
$$= 1 + 5$$
$$= 6 \checkmark$$

S → B

$$f(B) = g(B) + h(B)$$
$$= 2 + 6$$
$$= 8 \checkmark$$

- Expanding S → A

S → A → X

$$f(X) = g(X) + h(X)$$
$$= 5 + 5$$
$$= 10 \checkmark$$

S → A → Y

$$f(Y) = g(Y) + h(Y)$$
$$= 8 + 8$$
$$= 16$$

- Expanding S → B

S → B → C $\therefore f(C) = g(C) + h(C)$

$$= 9 + 4 = 13$$

Scanned with CamScanner

Date :
Page:

$S \to B \to D$

$f(D) = g(D) + h(D)$
$= 3 + 15$
$= 18$

- Expanding $S \to A \to X$

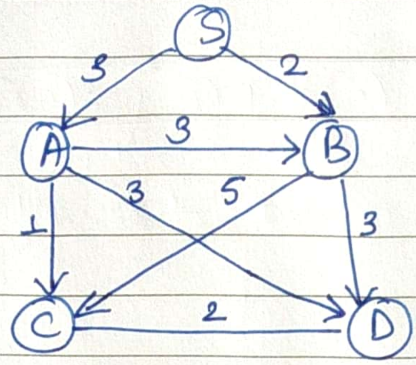
$S \to A \to X \to E$

$f(E) = g(E) + h(E)$
$= 7 + 0$
$= 7$ ✓

$\therefore S \to A \to X \to E$ is the best path
(path cost $= 7$)

Date :
Page:

Q



h(S)	h(A)	h(B)	h(C)	h(D)
1	3	3	0	0

(i) Greedy search:

$$f(n) = h(n)$$

- Start node: S

$S \to A$

$f(A) = h(A) = 3$ ✓

$S \to B$

$f(B) = h(B) = 3$

- Expanding $S \to A$

~~$S \to A \to D$~~

~~$f(D) = h(D) = 0$~~

$S \to A \to B$

$f(B) = h(B) = 3$

$S \to A \to C$

$f(C) = h(C) = 0$

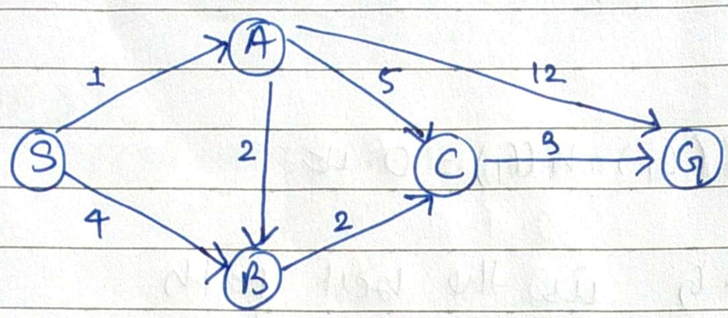
$S \to A \to D$

$f(D) = h(D) = 0$

Scanned with CamScanner

Date :
Page:

Q



State	h
S	7
A	6
B	2
C	1
G	0

(i) Greedy Search:

$$f(n) = h(n)$$

- Start node: S
- Expanding S,

$S \rightarrow A$

$$f(A) = h(A) = 6$$

$S \rightarrow B$

$$f(B) = h(B) = 2$$ ✓

- Expanding $S \rightarrow B$,

$S \rightarrow B \rightarrow C$

$$f(C) = h(C) = 1$$

Date:
Page:

- Expanding $S \to B \to C$,

$S \to B \to C \to G$

$$f(G) = h(G) = 0 \text{ \& }$$

$\therefore S \to B \to C \to G$ is the best path
(path cost = 9)

ii) A* Search:

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

- Start node: S

- Expanding S,

$S \to A$

$$f(A) = g(A) + h(A) = 1 + 6$$
$$= 7$$

$S \to B$

$$f(B) = g(B) + h(B)$$
$$= 4 + 2$$
$$= 6 \checkmark$$

- Expanding $S \to B$,

$S \to B \to C$,

$$f(C) = g(C) + h(C)$$
$$= 6 + 1$$
$$= 7$$

Date : ....
Page: ....

- Expanding $S \to A$, & Expanding $S \to B \to C$,

<u>$S \to A \to B$</u>

$$f(B) = g(B) + h(B)$$
$$= 3 + 2$$
$$= 5 \checkmark$$

<u>$S \to B \to C \to G$,</u>

$$f(G) = g(G) + h(G)$$
$$= 9 + 0$$
$$= 9$$

<u>$S \to A \to C$</u>

$$f(C) = g(C) + h(C)$$
$$= 6 + 1$$
$$= 7$$

<u>$S \to A \to G$</u>

$$f(G) = g(G) + h(G)$$
$$= 13 + 0$$
$$= 13$$

- Expanding $S \to A \to B$,

<u>$S \to A \to B \to C$.</u>

$$f(C) = g(C) + h(C)$$
$$= 5 + 1$$
$$= 6 \checkmark$$

- Expanding $S \to A \to B \to C$,

<u>$S \to A \to B \to C \to G$,</u>

$$f(G) = g(G) + h(G)$$
$$= 8 + 0$$
$$= 8 \checkmark$$

Date:
Page:

- Expanding $S \to A \to C$,

$S \to A \to C \to G$

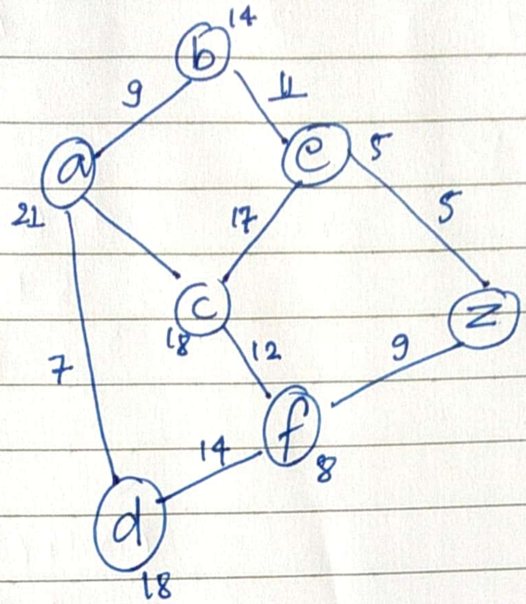
$$f(G) = g(G) + h(G)$$
$$= 9 + 0$$
$$= 9$$

$\therefore S \to A \to B \to C \to G$ is the best path,
(path cost = 8)

Scanned with CamScanner

Date : 2078/08/11
Page:

Q



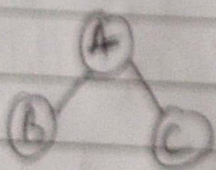
State	h
a	21
b	14
c	18
d	18
e	5
f	8

here,

i) Greedy Method:

Greedy	A*
1) $f(n) = h(n)$	- $f(n) = g(n) + h(n)$
2) Heuristic value	- Also considers path cost
3) Not required	- Back tracking
4) Simple & faster	- Comparatively slower
5) Comparatively lower	- Time complexity & space complexity higher.

eg :



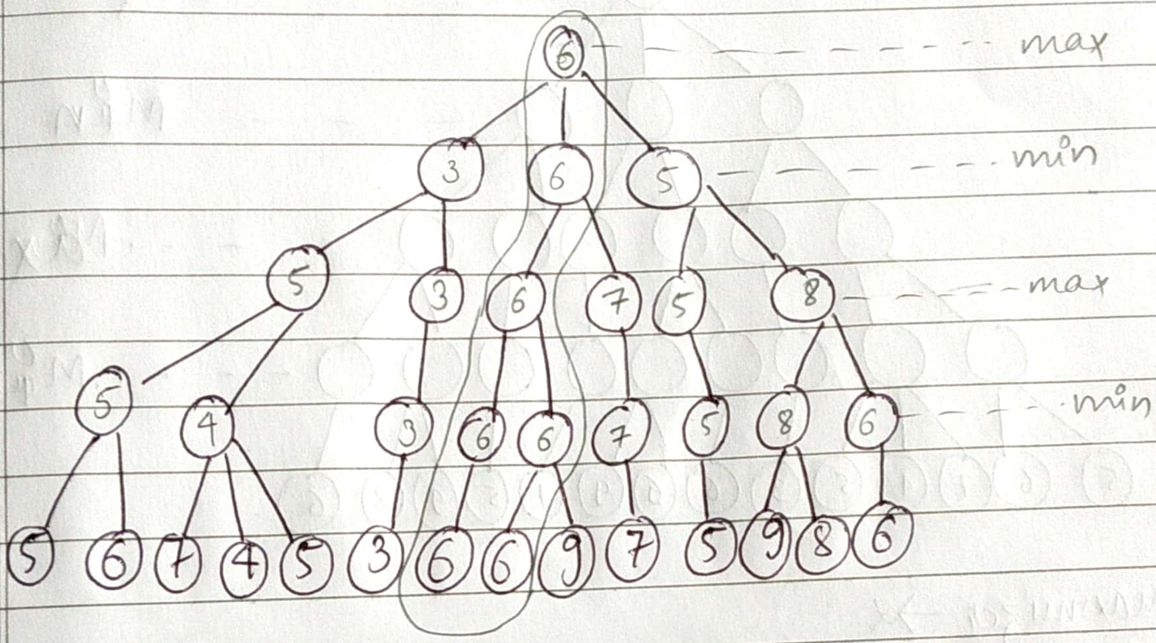
2078/08/12

Date:

Page:

Minmax Algorithm

2-player games (Tic-Tac-Toe, Chess, Checkers)



Minimizer : selects min. value
Maximizer : selects max. value

↳ Game playing algorithm used in 2 player games.
(Tic-Tac-Toe, Chess, checkers)

↳ Two-players : Minimizer
: Maximizer

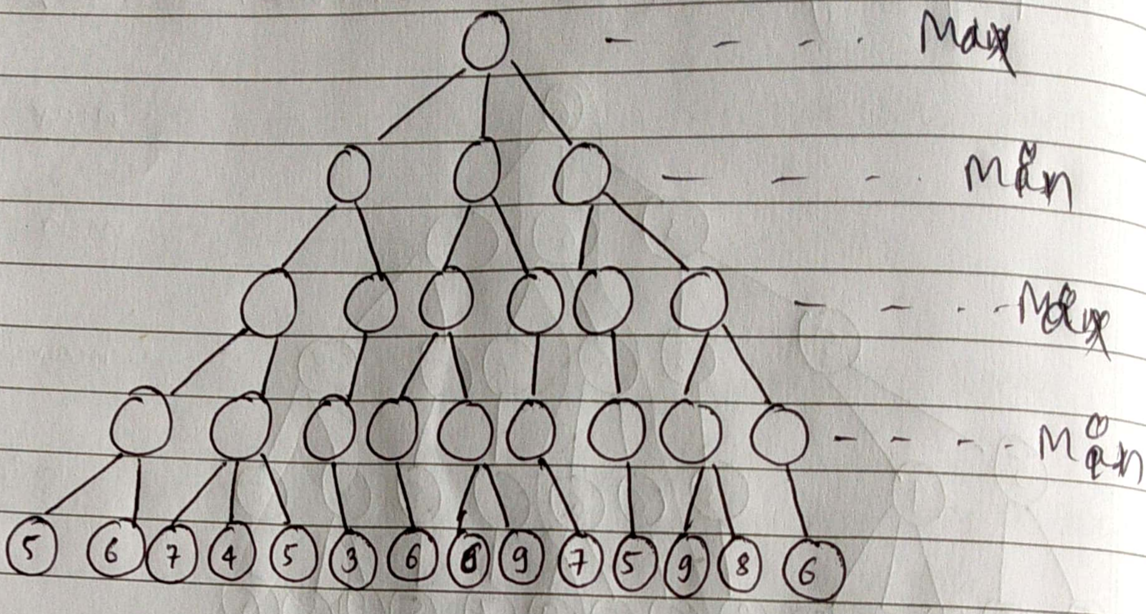
↳ Limitations:
- Time & space complexity higher

↳ Alpha-beta pruning overcomes these limitations.

Scanned with CamScanner

Date: 2078/08/15
Page:

α-β Pruning (Alpha-Beta Pruning)



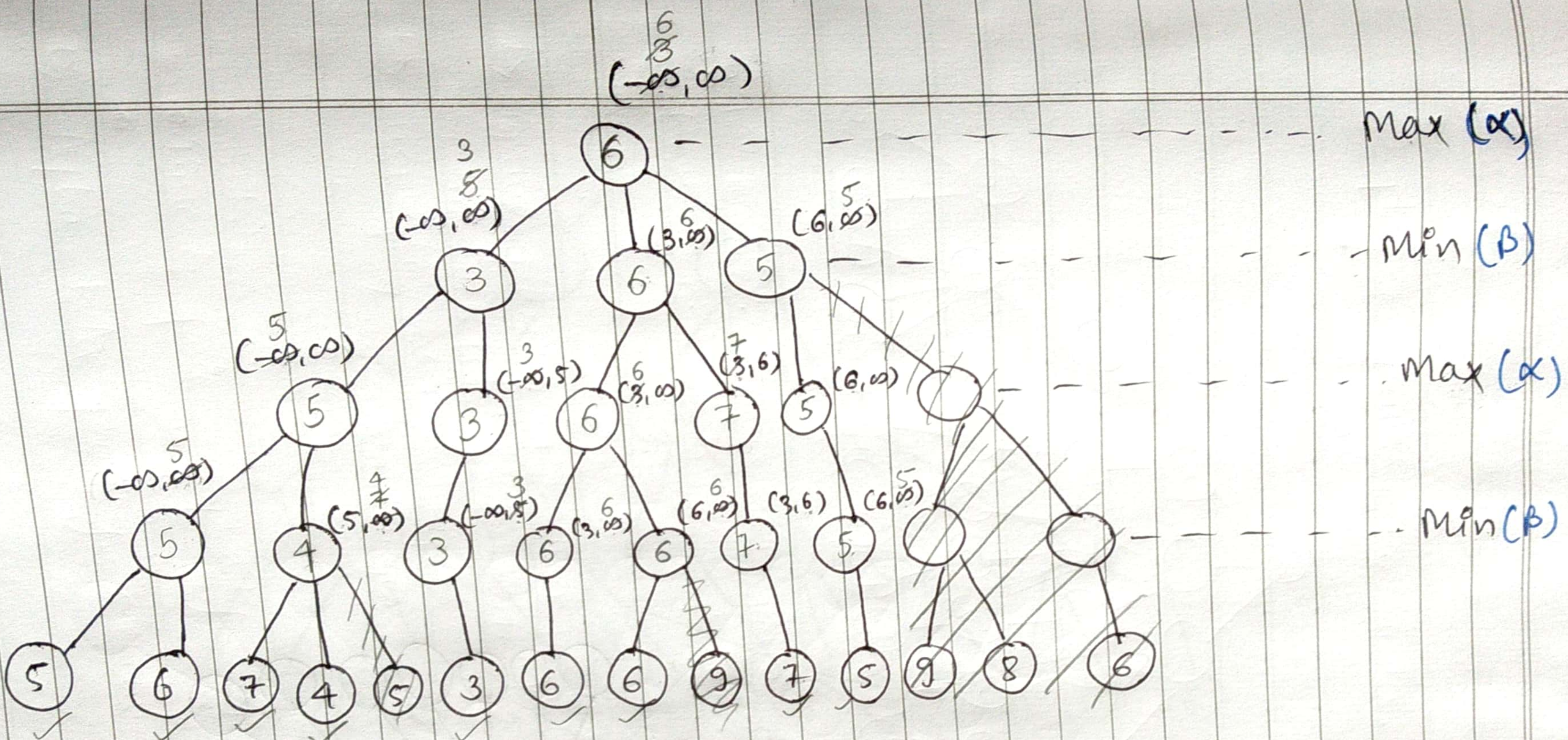
→ Maximizer → α
→ Minimizer → β

→ Initial value $(\overset{\alpha}{-\infty}, \overset{\beta}{\infty})$

1) Check $\overset{(\beta)}{\text{min}}$ or $\overset{(\alpha)}{\text{max}}$ level
2) Update α-β values
3) Push min max values

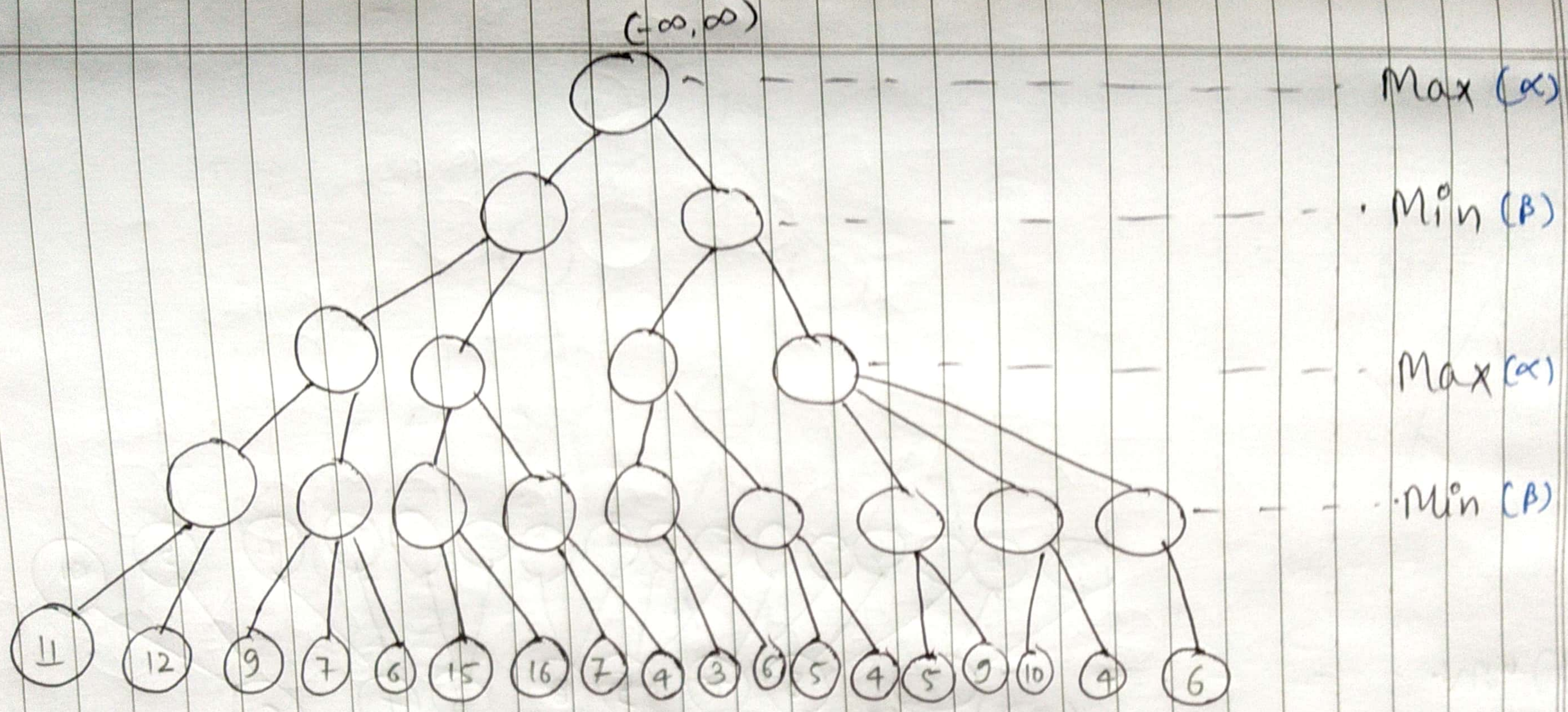
Working Condition

$\boxed{\alpha \geq \beta}$,
prune the branch
(cut off)



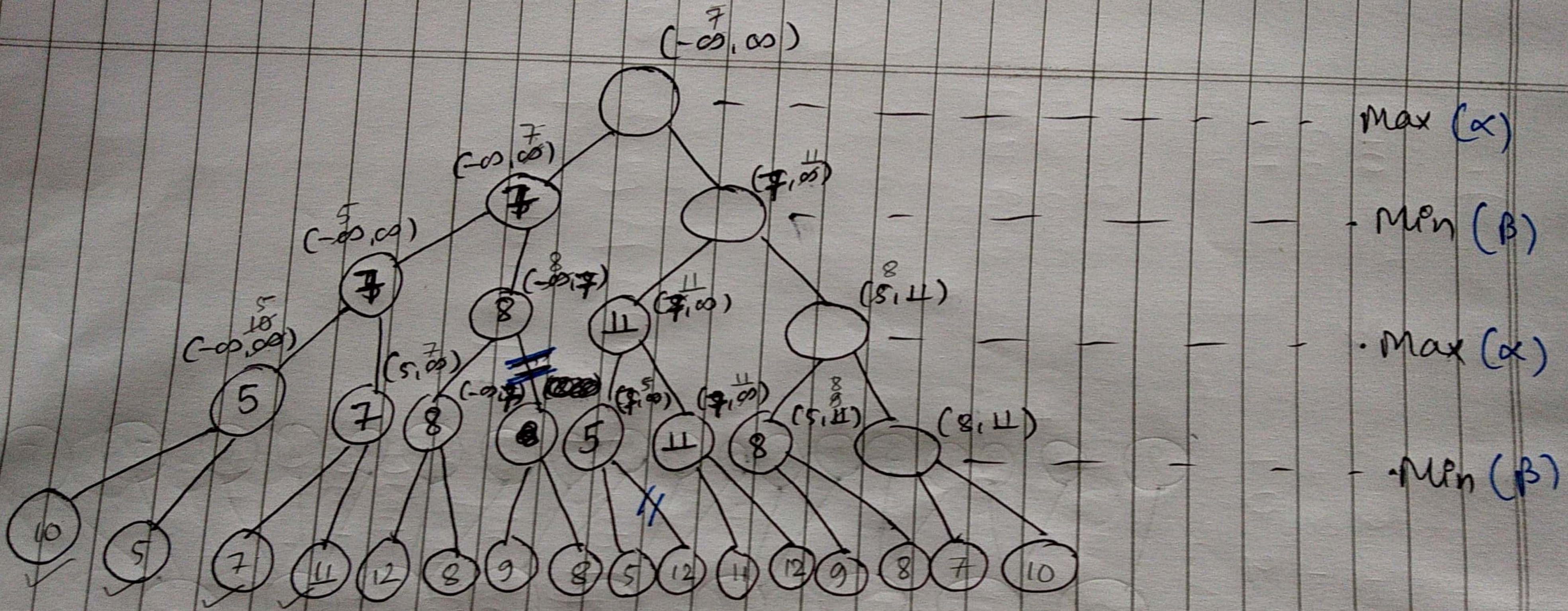
α - cutoff = 0

β - cutoff = 3

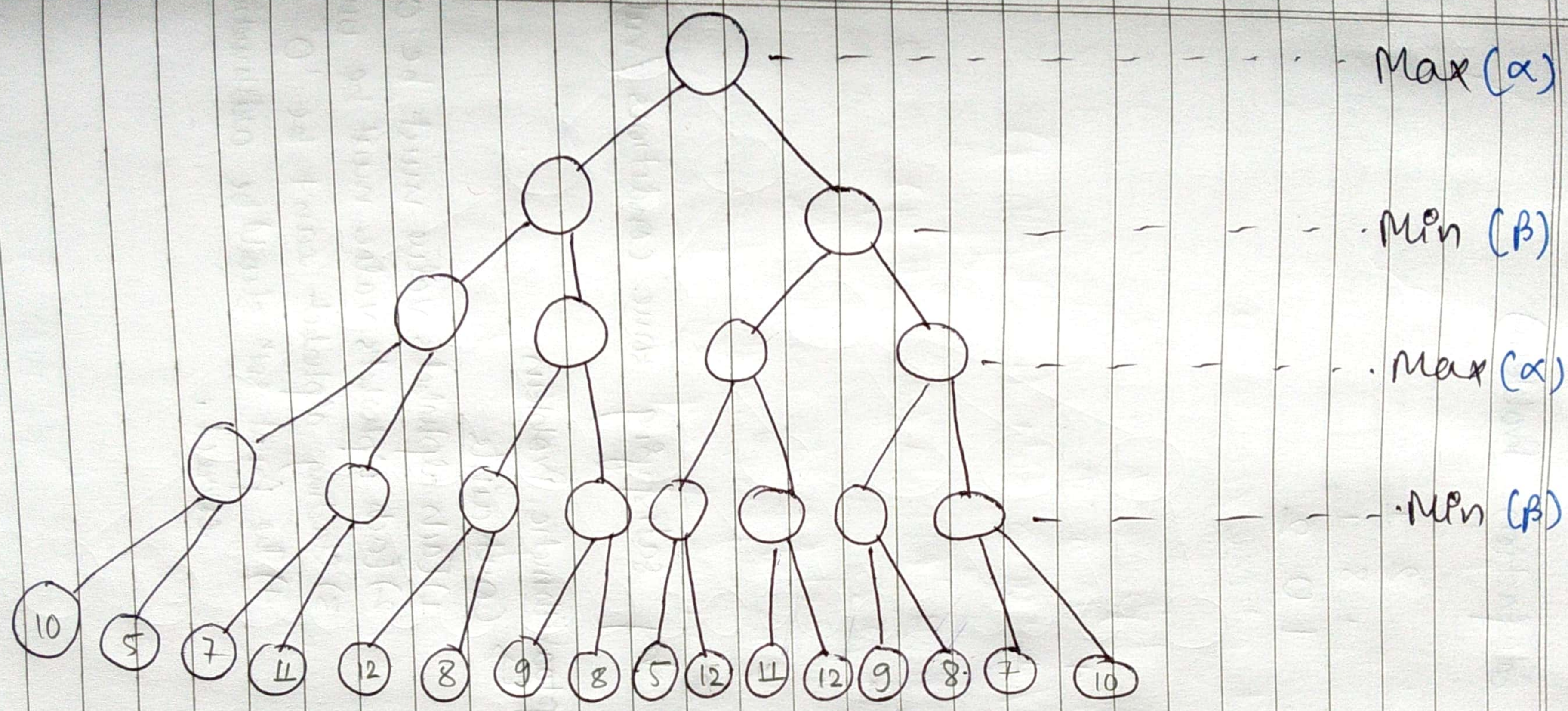


Scanned with CamScanner

Date : ....
Page: ....



Scanned with CamScanner



Date :

Page:

Scanned with CamScanner

Date :
Page:

2078/08/20

CSP (Constraint Satisfaction Problem)

$$\begin{array}{r} TO \\ +\,GO \\ \hline OUT \end{array} \rightarrow \begin{array}{r} 21 \\ +\,81 \\ \hline 102 \end{array}$$

$$\begin{array}{r} TWO \\ +\,TWO \\ \hline FOUR \end{array} \Rightarrow \begin{array}{r} 734 \\ +\,734 \\ \hline 1468 \end{array} \quad \begin{array}{r} 836 \\ +\,836 \\ \hline 1672 \end{array} \quad \begin{array}{r} 928 \\ 928 \\ \hline 1856 \end{array}$$

$$\begin{array}{r} 938 \\ +\,938 \\ \hline 1876 \end{array}$$

↳ Solving problem by satisfying some conditions / rules.

eg:

Sudoku, Cryptoarithmetic Problem

↳ Constraints

1) Each alphabet's value must be 0-9
2) Each alphabet's value must be unique
3) Starting alphabet can't be '0'
4) The final soln should be arithmetically correct.

Scanned with CamScanner

	L O G I C				9	0	4	5	2
+	L O G I C		+		9	0	4	5	2
	P R O L O G			1	8	0	9	0	4

		7	9	6	4	8
+		7	9	6	4	8
	1	5	9	2	9	6

			$c_2=$	$c_1=1$	$c_0=1$	
	S E N D		9	5	6	7
+	M O R E	+	1	0	8	5
	M O N E Y	1	0	6	5	2

E	N
0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9

1) $E+1=N$
2) $N+R+C_0=10+E$

$\cancel{E}+1+R+C_0=10+\cancel{E}$

$R+C_0=9$

Date :
Page:

Uncertainity with Baye's Rule

Q) What is uncertainity? Sources? How to overcome? Baye's Rule? Eg.?

↳ Unable to predict outcome / conclusion

↳ Sources of Uncertainty

1) Incomplete data / Unknown data (Under fitting) (Insufficient)
2) Vague data / Redundant Data (Over fitting) / Weak Implication
3) Imprecise Language:
   Use of ambiguous words (hardly ever, rarely, often, sometimes)
4) Combining / Multiple views of experts

★ To overcome uncertainity, use Baye's Rule.

Date : ....
Page: ....

Bayes' Rule / Bayesian Network / BBN (Bayes' Belief Network)

↳ Handles uncertainty / uncertain data

↳ deals with conditional probability, probability of an event 'A' ($P(A/B)$) given that 'B' has occured prior.

$$P(A/B) = \frac{P(A \cap B)}{P(B)} \quad \text{———} \quad ① \qquad P(A \cap B) = P(A/B) \cdot P(B)$$

$$P(B/A) = \frac{P(B \cap A)}{P(A)} \quad \text{———} \quad ② \qquad P(A \cap B) = P(B/A) \cdot P(A)$$

$$P(A/B) = \frac{P(B/A) \cdot P(A)}{P(B)} \quad \text{———} \quad ③$$

Date :
Page:

Numerical

1) While watching a game of champion League of football in a cafe, you observe someone who is clearly supporting Manchester United in the game. What is the prob. that they were born within 25 miles of Manchester.

Assume that:

→ Prob. that a randomly selected person in that bar is born within 25 miles of Manchester is $1/20$.

→ The chance that a person born within 25 miles of manchester actually ~~st~~ supports Manchester United is $7/10$.

→ Prob. that ~~that~~ a person not born within 25 miles of Manchester supports Manchester United is $1/10$.

Soln,

S = Supports Man U
B = Born within 25 miles of Manchester

Given,

$$P(B) = \frac{1}{20}$$

$$P(S/B) = \frac{7}{10} \qquad P(B/S) = ?$$

$$P(S/\bar{B}) = \frac{1}{10}$$

Date :

Page:

now,

$$P(B/S) = \frac{P(S/B) \cdot P(B)}{P(S)} \quad \text{——} \; 1$$

here,

$$P(S) = P(B) \cdot P(S/B) + P(\bar{B}) \cdot P(S/\bar{B})$$

$$= \frac{1}{20} \cdot \frac{7}{10} + \left(1 - \frac{1}{20}\right) \cdot \frac{1}{10}$$

$$= 8$$

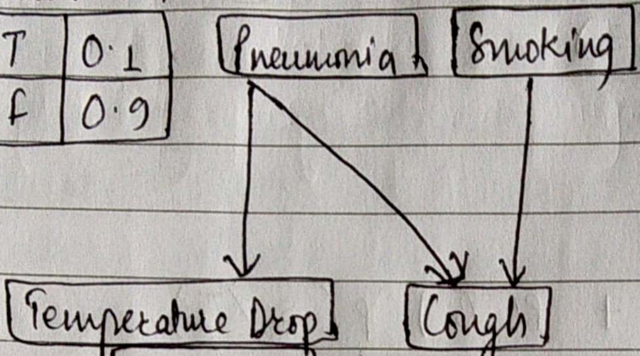
Date : ....
Page: ....

2078/08/26

Bayesian Network

Pneumonia (Pn) (S) Smoking

T	0.1
f	0.9



T	0.2
f	0.8

Temperature

Pneumonia	Y	N
Y	0.9	0.1
N	0.2	0.8

Cough

Pneumonia	Smoking	T	F
T	Y	0.95	0.05
T	N	0.8	0.2
f	Y	0.6	0.4
f	N	0.08	0.95

Conditional Probability Table (CPT)

1) $P(C/S \cap P_n) = 0.95$ (Prob. of cough due to Pneumonia & Smoking)
2) $P(C)$ = Prob. of cough
3) $P(C/S)$ = Prob. of cough due to smoking.

Date :
Page:

2) $P(C)$ = prob. of Cough

$$= P(P_n) \cdot P(S) \cdot P(C/P_n \cap S) + P(P_n) \cdot P(\bar{S}) \cdot P(C/P_n \cap S)$$
$$+ P(\bar{P_n}) \cdot P(S) \cdot P(C/\bar{P_n} \cap S) + P(\bar{P_n}) \cdot P(\bar{S}) \cdot P(C/\bar{P_n} \cap \bar{S})$$

3) $P(C/S)$ = Prob. of cough due to smoking

$$= P(P_n) \cdot P(S) \cdot P(C/P_n \cap S) + P(\bar{P_n}) \cdot P(S) \cdot P(C/\bar{P_n} \cap S)$$

Date : ....
Page: ....

2) $P(C)$ = prob. of Cough

$$= P(P_n) \cdot P(S) \cdot P(C/P_n \cap S) + P(P_n) \cdot P(\bar{S}) \cdot P(C/P_n \cap S) + P(\bar{P_n}) \cdot P(S) \cdot P(C/\bar{P_n} \cap S) + P(\bar{P_n}) \cdot P(\bar{S}) \cdot P(C/\bar{P_n} \cap \bar{S})$$

3) $P(C/S)$ = Prob. of cough due to smoking

$$= P(P_n) \cdot P(S) \cdot P(C/P_n \cap S) + P(\bar{P_n}) \cdot P(S) \cdot P(C/\bar{P_n} \cap S)$$

Date:
Page:

Chapter-3

Knowledge Representation

↳ The branch of AI that deals with storing or representing knowledge / datasets in Database / KB (Knowledge Base) for inference or reasoning.

KR schemes:

1) Production Rule
2) Logic
3) Semantic Network
4) Frames

1) Production Rule:

If A, then, B
Condition → Action

fill 4L jug.
$x < 4 \rightarrow (4, y)$

2) Logic:
→ Propositional Logic
→ Predicate Logic

a) Proportional Logic:
- Declarative sentence that can be either True or false

Logical Operators:

1) Unary Operator ($\neg$): Single operand
2) Binary Operator ($\wedge, \vee, \rightarrow, \leftrightarrow$): Two operands

Truth Table:

(Or/Disjunction) (And/Conjunction) (Conditional/Implication) (Bi-conditional/Double Implication)

P	Q	$\neg P$	$\neg Q$	$P \vee Q$	$P \wedge Q$	$P \rightarrow Q$	$P \leftrightarrow Q$	$\neg P \vee Q$	$Q \rightarrow P$	$(P \rightarrow Q) \wedge (Q \rightarrow P)$
T	T	F	F	T	T	T	T	T	T	T
T	F	F	T	T	F	F	F	F	F	F
F	T	T	F	T	F	T	F	T	T	F
F	F	T	T	F	F	T	T	T	T	T

$\equiv$ $\equiv$

P: "It is raining"
Q: "I will take an umberalla"

Prove:

a) $P \rightarrow Q \equiv \neg P \vee Q$
b) $P \leftrightarrow Q \equiv (P \rightarrow Q) \wedge (Q \rightarrow P)$

2078/08/27

Date: ....
Page: ....

Propositional Logic:

Normal forms:

a) CNF (Conjunctive Normal form) Conjunction of Disjunctions
b) DNF (Disjunctive Normal form) Disjunction of Conjunction.

CNF Conversion Steps:

Step 1:

Eliminate Conditional and Bi-conditional / Double Implication (if Present)

$$P \to Q \equiv \neg P \vee Q$$

P = "It is cold"
Q = "I will wear a sweater"

$$P \leftrightarrow Q \equiv (P \to Q) \wedge (Q \to P)$$
$$= (\neg P \vee Q) \wedge (\neg Q \vee P)$$

Step 2:

De-Morgan's Law:

$$\neg(P \vee Q) \equiv \neg P \wedge \neg Q$$
$$\neg(P \wedge Q) \equiv \neg P \vee \neg Q$$

Scanned with CamScanner

Date :
Page:

Step 3 :

Apply Distributive law :

$$P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$$
$$P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$$

* Convert $P \rightarrow Q \rightarrow R$ into CNF

Step 1 :

$$(P \rightarrow Q) \rightarrow R$$
$$\equiv (\neg P \vee Q) \rightarrow R$$
$$\equiv \neg (\neg P \vee Q) \vee R$$

Step 2 :

$$\neg (\neg P \vee Q) \vee R$$
$$\equiv (P \wedge \neg Q) \vee R$$

Step 3 :

$$P (\wedge \neg Q) \vee R$$
$$\equiv (P \vee R) \wedge (\neg Q \vee R)$$

reqⁿ CNF

Truth Table :

PTO

P	Q	R	$P\to Q$	$(P\to Q)\to R$	$\neg Q$	$\neg Q\vee R$	$P\vee R$	$(P\vee R)\wedge(\neg Q\vee R)$
T	T	T	T	T	F	T	T	T
T	T	F	T	F	F	F	T	F
T	F	T	F	T	T	T	T	T
T	F	F	F	T	T	T	T	T
F	T	T	T	T	F	T	T	T
F	T	F	T	F	F	F	F	F
F	F	T	T	T	T	T	T	T
F	F	F	T	F	T	T	F	F

~~Proof by Resoluti~~

Proof by Resolution / Resolution Refutation

If it is hot, then it is humid. If it is humid, then it will rain. It is hot. Prove that it will rain.

Steps

Convert all statements into CNF.
Negate the desired conclusion
Apply resolution rule.

$$\begin{array}{l} P \lor Q \\ \neg Q \lor R \\ \hline \therefore P \lor R \end{array}$$

When we derive contradiction, our assumption was false and given statement is proved.

Given:

= "It is hot"
= "It is humid"
= "It will rain"

$\rightarrow Q$
$\rightarrow R$

Prove: R

Date :
Page:

Step 1 :

1) $P \rightarrow Q$
$\equiv \neg P \vee Q$

2) $Q \rightarrow R$
$\equiv \neg Q \vee R$

3) P

4) $\neg R$ (Assumption)

~~Step 2~~

Steps	Formulas	Derivations
1)	$\neg P \vee Q$	Given
2)	$\neg Q \vee R$	"
3)	P	"
4)	$\neg R$	"
5)	$\neg P \vee R$	Resolution in 1,2
6)	R	Resolution in 3,5

$\therefore$ We reach to a contradiction, hence "It will rain"

Scanned with CamScanner

Date :
Page:

Q. Either Heather attended the meeting or Heather was not invided. If the boss wanted Heather at the meeting, then, she was invited. Heather didn't attend the meeting. If the boss didn't want Heather there and the boss didn't invite her there, then, she is going to be fired.

Prove: Heather is going to be fired.

~~Given~~ Let:

P = "Heather attended the meeting"
$\neg Q$ = "Heather was not invited"
R = " Boss wanted Heather at meeting
S = "She is going to be fired"

Given:

1) $P \vee \neg Q$
2) $R \rightarrow Q$
3) $\neg P$
4) $(\neg R \wedge \neg Q) \rightarrow S$
5) $\neg S$ (Assumption)

Prove: S

Step 1:

1) $P \vee \neg Q$

2) $R \rightarrow Q \equiv \neg R \vee Q$

Date :
Page:

3) $\neg P$

4) $(\neg R \land \neg Q) \rightarrow S$
$\equiv \neg(\neg R \land \neg Q) \lor S$
$\equiv (R \lor Q) \lor S$

5) $\neg S$ (Assumption)

Steps	formulas	Derivations
1)	$P \lor \neg Q$	Given
2)	$\neg R \lor Q$	"
3)	$\neg P$	"
4)	$(R \lor Q) \lor S$	"
5)	$\neg S$	"
6)	$P \lor \neg R$	Resolution in 1, 2
7)	$\neg R$	Resolution in 3, 6
8)	$\neg Q$	Resolution in 1, 3
9)	$Q \lor S$	Resolution in 4, 7
10)	S	Resolution in 8, 9

Scanned with CamScanner

2078/09/05

Date: ....
Page: ....

Predicate Logic : / FOL / FOPL : statement consisting of variables, constant, predicates, quantifiers & logical operators.

Limitation of Proportional logic / Strength of Predicate Logic:

1) Proportional logic can't express equations.
"n is greater than 1"

2) It can't express logical equivalence between sentences in cases like: "Not all birds can fly" ≡ Some birds can't fly"

3) Proportional logic can't generalize statements.

2020 spring

Q Convert into FOL / FOPL

FOL → First order Logic
FOPL → First order predicate logic

1) Ramesh doesn't have a laptop
2) All purple mushroom are venomous.
3) Anything anyone eats and is not killed by is food.
4) Brijan eats mushroom and is still alive.
5) Ram likes all kinds of food.
6) Krishna eats everything Radha eats.
7) All marvel fans watch Spiderman.
8) Ram likes samosa.
9) If Ram is rich, he has a nice house
" Shyan " " " " " "
" Hari " " " " " " "
10) Not all birds can fly.
Some birds can't fly.
11) n is greater than 1.

Date: ....
Page: ....

2078/09/05

Predicate Logic : / FOL / FOPL : statement consisting of variables, constant, predicates, quantifiers & logical operators.

Limitation of Proportional logic / Strength of Predicate Logic:

1) Proportional logic can't express equations.
"x is greater than 1"

2) It can't express logical equivalence between sentences in cases like : "Not all birds can fly" ≡ Some birds can't fly"

3) Proportional logic can't generalize statements.

2020 spring

First order Logic → First order predicate Logic

Q Convert into FOL / FOPL

1) Ramesh doesn't have a laptop
2) All purple mushroom are venomous.
3) Anything anyone eats and is not killed by is food.
4) Brijan eats mushroom and is still alive.
5) Ram likes all kinds of food.
6) Krishna eats everything Radha eats.
7) All marvel fans watch Spiderman.
8) Ram likes samosa.
9) If Ram is rich, he has a nice house
" Shyan " " " " " " "
" Hari " " " " " " "
10) Not all birds can fly.
Some birds can't fly.
11) x is greater than 1.

Scanned with CamScanner

Date :
Page:

1) $\neg$Have (Ram, Laptop)
2) $\forall x$ Purple Mushroom $(x) \rightarrow$ Venomous (x)
3) $\forall x \forall y$ Eats $(x, y) \wedge \neg$Killed by $(x, y) \rightarrow$ food(y)
4) Eats (Bryan, Mushroom) $\rightarrow$ Still Alive (Bryan)
5) $\forall x$ food$(x) \rightarrow$ Likes (Ram, x)
6) $\forall x$ food$(x) \wedge$ Eats (Radha, x)
$\forall x$ Eats (Radha, x) $\rightarrow$ Eats (Krishna, x)
7) $\forall x$ Marvel fan $(x) \rightarrow$ watch (x, Spiderman)
8) Likes (Ram, Samosa
9) $\forall x$ IsRich $(x) \rightarrow$ Has(x, Nice House)
10) $\neg \forall x$ Birds$(x) \rightarrow$ Can fly (x)
$\exists x$ Birds $(x) \rightarrow \neg$CanFly(x)
11) $\exists x$ GreaterThan $(x, 1)$
$\exists x$ Number $(x) \rightarrow$ GreaterThan $(x, 1)$

2078/09/08

Date: ....
Page: ....

1) Ram likes all kinds of food
→ $\forall x \neg food(x) \rightarrow Likes(Ram, x)$

2) Orange is a food.
→ $food(Orange)$

3) Rice is a food
→ $food(Rice)$

4) Anything anyone eats and is not killed is food.
→ $\forall x$

5) Krishna eats Popcorn and is still alive.
→ $Eats(Krishna, Popcorn) \land Alive(Krishna)$

6) Krishna eats everything Radha eats.
→ $\forall x\ Eats(Radha) \rightarrow Eats(Krishna, x)$

7) Prove that "Ram likes Popcorn" using RRS
→ $Likes(Ram, Popcorn)$

Resolution Table:

Steps	~~Steps~~ Facts	Derivation
1)	$\neg food(x) \lor likes(Ram, x)$	Given
2)	$food(Orange)$	"
3)	$food(Rice)$	"
4)	$\neg Eats(a, f(a)) \lor Killed(a) \lor Food(y)$	"
5)	$Eats(Krishna,$ ~~Alive~~ $Popcorn)$	"
6)	$Alive(Krishna) \equiv \neg Killed(Krishna)$	"
7)	$\neg Eats(Radha, b) \lor Eats(Krishna, b)$	"
8)	$\neg Likes(Ram, Popcorn)$	Assumption

Date: ....
Page: ....

9) $\neg$ Eats(Krishna, f(Krishna)) $\vee$ food(f(Krishna))

a = Krishna
Resolution in 4, 6
f(Krishna) = Popcorn
Resolution in 5, 9
x = Popcorn

10) food(Popcorn)

11) Likes(Ram, Popcorn)

$\therefore$ Here, we reach a contradiction,
Hence proved.

2018 spring

i) Anyone whom Mary loves is a football star.
ii) Any student who doesn't pass doesn't play.
iii) John is a student.
iv) Any student who doesn't study doesn't pass.
v) Anyone who doesn't play is not a football star.

Prove: "If John doesn't study, then Mary doesn't love John".

FOL:

i) $\forall x$ Loves(Mary, x) $\rightarrow$ Football Star(x)
ii) $\forall x$ Student(x) $\wedge \neg$ Pass(x) $\rightarrow \neg$ Play(x)
iii) Student(John)
iv) $\forall x$ Student(x) $\wedge \neg$ Study(x) $\rightarrow \neg$ Pass(x)
v) $\forall x$ $\neg$ Play(x) $\rightarrow$ ~~Footbal~~ $\neg$ Football Star(x)

Date: 2025/08/22

Page:

~~Object Diagrams~~

~~[illegible]~~

~~Resolution Table~~

~~Steps Feel~~

CNF:

i) $\equiv \forall x \, \neg Loves(Mary, x) \lor FootballStar(x)$

$\equiv \neg Loves(Mary, x) \lor FootballStar(x)$

ii) $\forall x \, \neg(Student(x) \land \neg Pass(x)) \lor \neg Play(x)$

$\equiv \forall x \, \neg Student(x1) \lor Pass(x1) \lor \neg Play(x1)$

$\equiv \neg Student(x1) \lor Pass(x1) \lor \neg Play(x1)$

iii) $Student(John)$

iv) $\equiv \forall x \, \neg(Student(x) \land \neg Study(x)) \lor \neg Pass(x)$

$\equiv \forall x2 \, \neg Student(x2) \lor Study(x2) \lor \neg Pass(x2)$

v) $\equiv \forall x \, \neg(\neg Play(x)) \lor FootballStar(x)$

$\equiv \forall x3 \, Play(x3) \lor FootballStar(x3)$

$\equiv Play(x3) \lor \neg FootballStar(x3)$

Proof:

FOL:

$\neg Study(John) \rightarrow \neg Loves(Mary, John)$

CNF:

$\equiv \neg(\neg Study(John)) \lor \neg Loves(Mary, John)$

$\equiv Study(John) \lor \neg Loves(Mary, John)$

Date : ....
Page: ....

Resolution Table:

Steps	Facts	Derivations
1)	$\neg$ Loves(Mary, x) $\vee$ FootballStar(x)	Given
2)	$\neg$ Student (x1) $\vee$ Pass (x1) $\vee$ $\neg$ Play(x1)	"
3)	Student (John)	"
4)	$\neg$ Student (x2) $\vee$ Study (x2) $\vee$ $\neg$ Pass (x2)	"
5)	Play (x3) $\vee$ Football Star (x3)	"
6)	$\neg$ Student (John) $\vee$ Pass (John) $\vee$ $\neg$ Play (John)	x1 = John (Instatiation) Resolution in 2, 3
7)	Pass (John) $\vee$ $\neg$ Play (John)	Resolution in 3, 6
8)	$\neg$ Student (John) $\vee$ Study (John) $\vee$ $\neg$ Pass (John)	x2 = John
9)	Study (John) $\vee$ $\neg$ Pass (John)	~~Resolution~~ in Resolution in 3, 8
10)	Pass (John) $\vee$ $\neg$ FootballStar (John)	Resolution in 5, 7 x3 = John
11)	Study (John) $\vee$ $\neg$ FootballStar (John)	Resolution in 9, 10
12)	Study (John) $\vee$ $\neg$ Loves (Mary, John)	Resolution in 1, 11

2018 fall

2078/09/12

Date: ....

Page: ....

Q. Birendra likes easy courses.
SMP is a hard course.
~~AII~~ All courses in Software Department are easy.
AINN is Software Development course.
Use resolution to answer
What course Birendra likes?

~~Soln~~:

FOL:

i) $\forall x$ EasyCourse $(x) \rightarrow$ Likes (Birendra)

ii) HardCourse(SMP) $\equiv$ $\neg$ EasyCourse (SMP)

iii) $\forall x$ SoftwareDeptCourse $(x) \rightarrow$ EasyCourse (x)

iv) SoftwareDeptCourse (AINN)

CNF:

i) $\forall x$ EasyCourse $(x) \rightarrow$ Likes (Birendra, x)
$\equiv$ $\forall x$ $\neg$ EasyCourse (x) $\vee$ Likes (Birendra, x) [∵ Elimination]
$\equiv$ $\neg$ EasyCourse (x) $\vee$ Likes (Birendra, x) [∵ Drop $\forall x$]

iii) $\forall x$ SoftwareDeptCourse $(x) \rightarrow$ EasyCourse (x)
$\equiv$ $\forall x$ $\neg$ softwareDeptCourse(x) $\vee$ EasyCourse(x)
$\equiv$ $\forall a$ $\neg$ SoftwareDeptCourse(a) $\vee$ EasyCourse(a)
$\equiv$ $\neg$ SoftwareDeptCourse (a) $\vee$ EasyCourse (a)

Date :
Page:

Resolution Table:

Steps	facts	Derivations
1)	$\neg$EasyCourse(x) $\vee$ Likes(Birendra, x)	Given
2)	$\neg$EasyCourse (SMP)	"
3)	$\neg$SoftwareDeptCourse(a) $\vee$ EasyCourse(a)	"
4)	SoftwareDeptCourse (AINN)	"
5)	EasyCourse (AINN)	a=AINN (instantiation) Resolution in 3, 4
6)	Likes (Birendra, AINN)	x = AINN R in 1, 5

$\therefore$ Birendra likes AINN

Date :
Page:

Q/ Marcus was a man.
Marcus was a Roman
All men are people.
Caesar was a Ruler.
All Romans were either loyal to Caesar or hated him.
Everyone is loyal to someone.
People only try to assassinate ruler they are not loyal to.
Marcus tried to assassinate Caesar.
Prove:
Marcus hated Caesar

FOL:

i) Man (Marcus)
ii) Roman (Marcus)
iii) $\forall x$ Man $(x) \rightarrow$ People (x)
iv) Ruler (Caesar)
v) $\forall x$ Roman $(x) \rightarrow$ LoyalTo $(x, Caesar)$ $\vee$ Hated $(x, Caesar)$
vi) $\exists x \forall y$ LoyalTo (x, y)
vii) $\forall x \forall y$ People$(x) \wedge$ Ruler$(y) \wedge \neg$ LoyalTo$(x, y) \rightarrow$ TriedtoAssassinate (x, y)
viii) Tried To Assassinate (Marcus, Caesar)
Prove:
Marcus Hated Caesar [Hated (Marcus, Caesar)]

CNF:

show all steps

iii) $\neg$Man (x) $\vee$ People (x)
v) $\neg$Roman(a) $\vee$ LoyalTo $(a, Caesar)$ $\vee$ Hated $(a, Caesar)$
vi) LoyalTo $(c, f(c))$
vii) $\neg$People (b) $\vee$ $\neg$Ruler $(f(b))$ $\vee$ LoyalTo $(b, f(b))$ $\vee$ TriedToAssassinate $(b, f(b))$

2078/09/13

VVI

Semantic Networks & Frames

KR scheme that represents knowledge in graphical networks. Also called "Associative network".

* Associative Network:
  - ↳ shows association between nodes
  - ↳ consists of;
    1) Nodes
    2) Links / Edges
    3) Link labels

1) Nodes:

Represents entity (nouns) or objects represented by circle, ellipse or rectangle.

2) Links / Edges:

Represents direction of relationship between nodes.

3) Link Labels:

Represents the relationship between nodes.

* Knowledge Representation
  - ↳ Production Rule
  - ↳ Logic
    - Proposition
    - Predicate
  - ↳ Semantic Nets
  - ↳ Frames

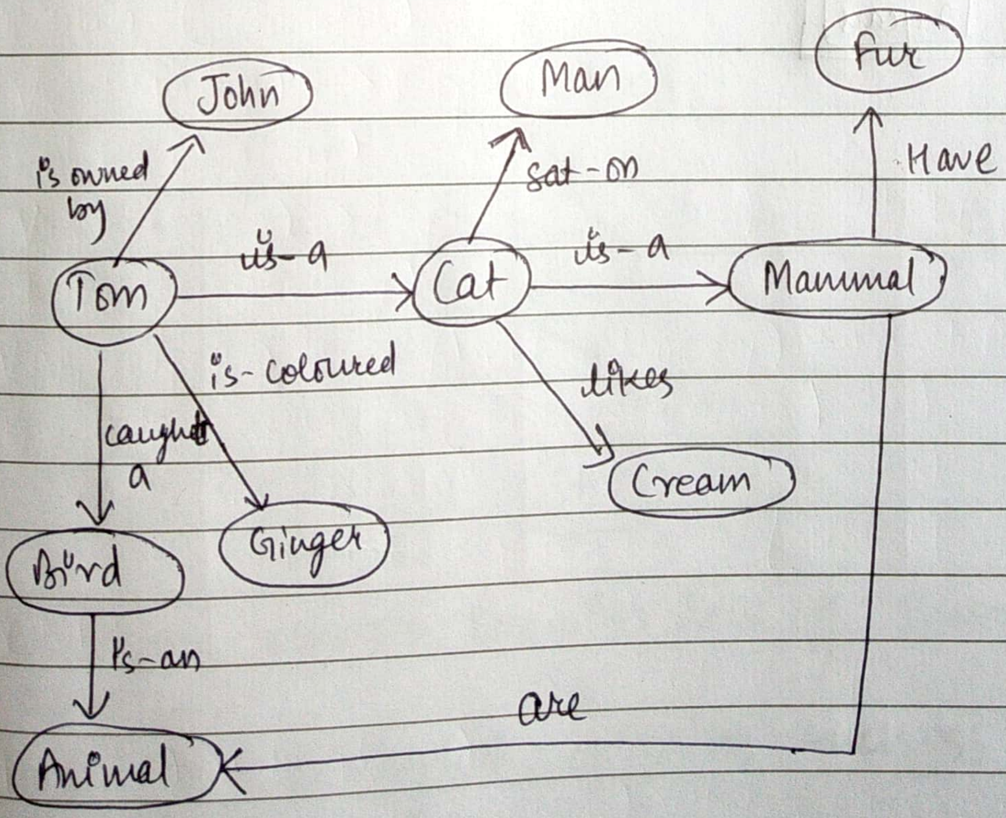
Date :
Page:

VVI

Represent following in Semantic Network

1) Tom is a Cat.
2) Tom caught a bird.
3) Tom is ginger in color.
4) Cats like cream.
5) Tom is owned by John
6) The cat sat on the mat.
7) A cat is a mammal.
8) All bird is an animal.
9) All mammals are animals.
10) Mammals have fur.

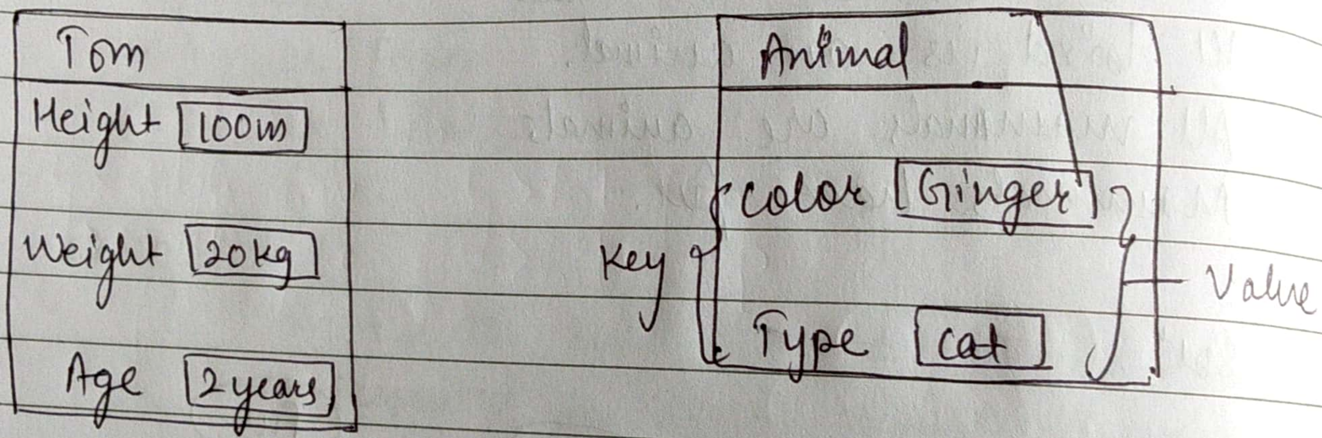
Sol^n



Frames:

- ↳ represent additional proper or characteristics of entity / nodes in semantic networks.
- ↳ extention of semantic networks
- ↳ Represent characteristics in key-value pair

eg:



2078/09/18

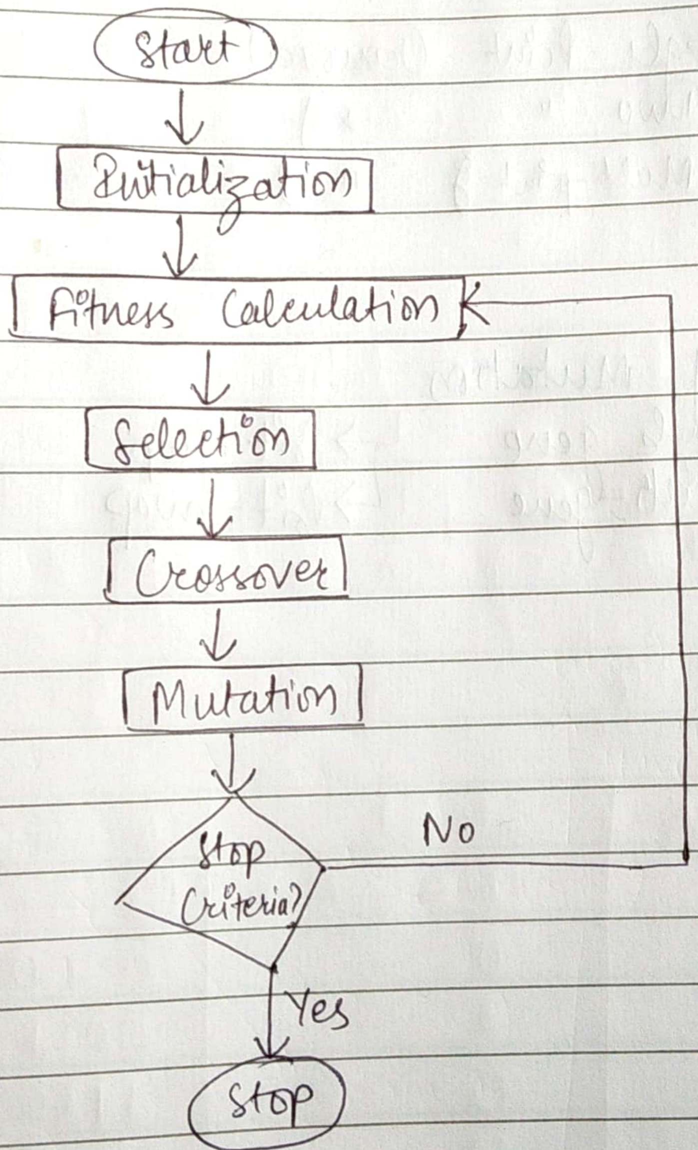
Date: ....

Page: ....

Chapter-5

Genetic Algorithm (GA)

↳ Maximization / Optimization Algorithm



↳ Based on Charle's Darwin's law of Theory of Natural selection.

↳ Based on Biological operators such as;

→ Selection

→ Crossover

→ Mutation

Date :

Page:

~~2078/09/19~~

Q Maximize $f(x) = x^2$; $0 \leq x \leq 31$, $n = 4$

~~$f(x) = 15x - x^2$, $0 \leq x \leq 16$, $n = 4$~~

↳ Types of Crossover:

1) SPX (Single Point Crossover)
2) TPX (Two " ")
3) MPX (Multi-point " ")

↳ Types of Mutation

↳ Single gene
↳ Multi-gene

↳ Bit-flop
↳ Bit-swap

Scanned with CamScanner

2078/09/19

2019 fall

Use GA to find maximum value of function $f(x) = 15x - x^2$, $0 \le x \le 15$ and take population size of 4.

Step 1: Initialization

Select chromosomes randomly where no. of chromosomes equals to population size.

$x = 0, 3, 12, 14$

Step 2: Fitness Calculation

Calculate fitness of selected chromosomes. The fitness equation is defined by $f(x) = 15x - x^2$

1st Gen:

Fitness Table

Chromosome Label	Selected Chromosome (x)	Chromosome (in Binary)	Fitness Value ($f(x)$)
A	0	0000	0
B	3	0011	36
C	12	1100	36
D	14	1110	14
			$\Sigma f(x) = 86$

Step 3: Selection

B & C , B & D

Select parent chromosomes for crossover.

Date :
Page:

Step 4: Crossover / Mating / Reproduction

B = 0|011   B = 0|011
C = 1|100   D = 1|110
SPX   SPX

Exchange of genes between parent chromosome

B' = 0100   B" = 0110
C' = 1011   D = 1011

A = 10|10|11   A = 10|00|100
B = 01|10|10   B = 01|10|0010
TPX   MPX

A' = 101011   A' = 1010100
B' = 011010   B' = 0100010

Step 5: Mutation

Random change in gene of chromosome

C' = $\boxed{10}$11 = 0111
Bitswap

2nd Gen

PTO

Date : ....
Page: ....

Fitness Table

Chromosome Label	Selected Chromosome (x)	Chromosome (in binary)	Fitness value ($f(x)$)
B'	4	0100	44
C''	7	0111	56
B''	6	0110	54
D'	11	1011	44
			$\Sigma f(x) = 198$

Conclusion:

1) fitness value increased from 86 to 198
2) When, $x = 7$, $f(x)$ is maximum

Scanned with CamScanner

Date :
Page:

Q $f(x) = 15x - x^2$, $0 \le x \le 31$, $n = 6$

Step 1: Initialization

$x = 0, 3, 12, 5, 6, 20$

Step 2: Fitness Calculation

1st Gen

Fitness Table

Chromosome Label	Selected Chromosomes(x)	Chromosome (in binary)	Fitness Value [f(x)]
A	0	00000	0
B	3	00011	36
C	12	01100	36
D	5	00101	50
E	6	00110	54
F	20	10100	-100
			$\Sigma f(x) = 76$

Step 3: Selection

E & B , E & C , B & D

Scanned with CamScanner

Date :
Page:

Step 4: Crossover

E = 00|110
D = 00|101

E = 00|110
C = 01|100

B = 00|011
D = 00|101

E' = 00101
D' = 00110

E'' = 00100
C' = 01110

B' = 00101
D'' = 00011

Here, Mutation in E' & B'

B' = $\boxed{0}$0101
B'' = 10101

(single gene bit-flip)

2nd Gen

Fitness Table

Chromosome Label	Selected Chromosome(n)	Chromosome (in binary)	Fitness Value [f(n)]
E'	5	00101	50
D'	6	00110	54
E''	4	00100	44
C'	14	01110	14
B''	21	10101	-126
D''	3	00011	36
			$\Sigma f(n) = 72$

1) fitness value decreased from 76 to 72
2) When, $n = 6$, $f(n)$ is maximum

Scanned with CamScanner

Date :
Page:

2078/09/21

Q. No. of bits in chromosome = 8

$f(x) = (a+b) - (c-d) + (e+f) - (g \cdot h)$

Take pop^n size $(n) = 6$

Sol^n,

→ Initialization :

1st gen :

	a	b	c	d	e	f	g	h
A =	5	4	3	2	1	1	2	3
B =	1	2	3	4	5	6	7	8
C =	9	9	9	9	9	9	9	9
D =	2	4	6	8	1	3	5	7
E =	0	0	0	0	0	0	0	0
F =	9	8	4	1	0	4	2	6

→ Fitness Calculation :

A = 11 — 5

B = 16 — 3

C = 36 — 1

D = 14 — 4

E = 0 — 6

F = 22 — 2

+99

→ Selection :

C&F , C&B , C&D

GA Operators
- Fitness calc
- selection
- crossover
- Mutation

→ explain these

Date :
Page:

→ **Crossover** :

C = 9999 | 9999
F = 9841 | 0426
SPX

C = 9999 | 9999
B = 1234 | 5678
SPX

C = 9999 | 9999
D = 2468 | 1357
SPX

C' = 9999 0426
F' = 9841 9999

C'' = 99995678
B = 12349999

C''' = 9999 1357
D' = 24689999

→ **Mutation** : No Mutation

2nd gen :

	f(x)
C' =	26
F' =	32
C'' =	30
B' =	22
C''' =	24
D' =	26

Date :
Page:

Machine Learning (ML)

↳ Application of AI that deals with training machines for learning and improving from experience without ~~beign~~ being explicitly programmed.

↳ Types of ML

1) Supervised ML : i/p + processing given / training labels given
2) Unsupervised ML: i/p, self-processing / training labels not given

1) Supervised ML: