Sabin
209-17

PAGE NO. :
DATE : / /

29-01-18

ARTIFICAL INTELLIGANCE
INT-404

★ **Characteristics of problem :—**

6. **What is the role of knowledge**

knowledge is imp. only to Constrain the Search for Solution.

Ex: Playing chess

7. **Does the task require interaction with the person?**

Solitary:
Ex: Theorem Proving

Conversational:
Ex: Medical diagnosis

★ **Production System Characteristics :—**

It is a good way to describe the operations that Can be performed in a Search for a Solution to a problem.

class of production System

Monotonic production System
a System in which the application of the never prevents the Later application of another rule that Could also have been applied at the time that the first rule was selected. Ex: Theorem Proving.

PAGE NO. :
DATE : / /

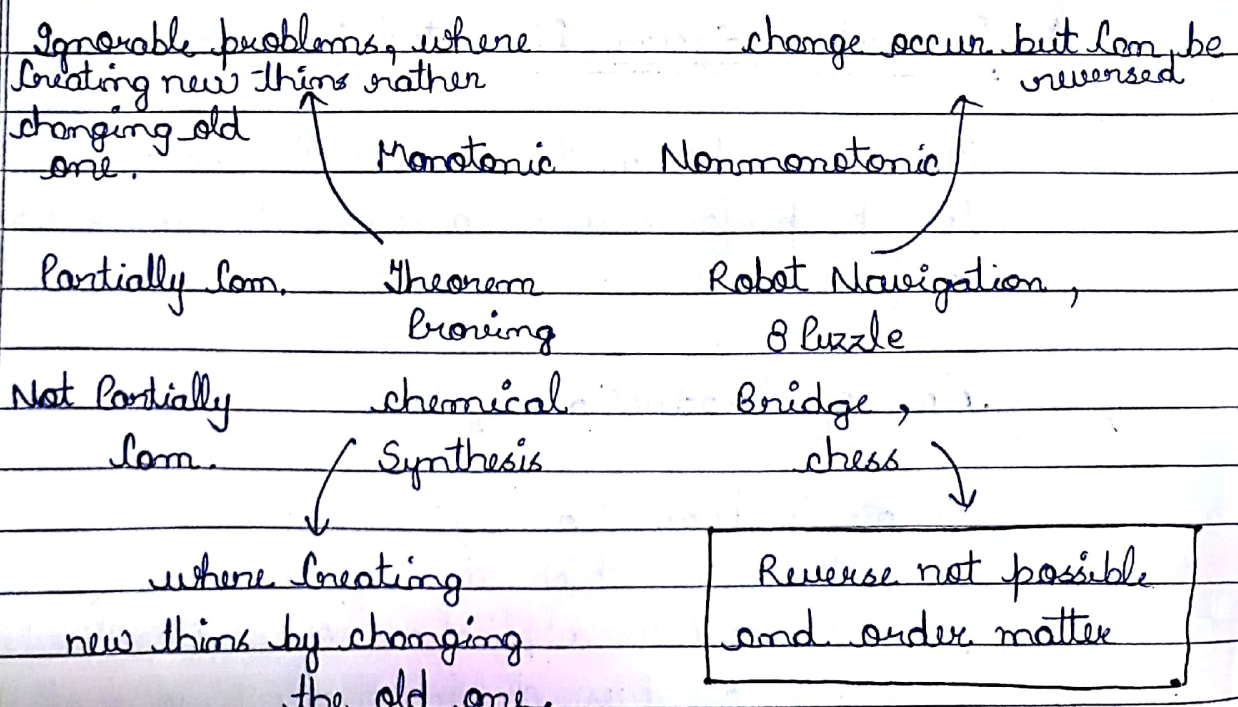
→ **Nonmonotonic Production System**
is one in which this one is not true.

→ **Partially Commutative Production System**
is a system in with the property that if the application of particular sequence of rules transform State x into State y, then any permutation of those rules that is allowable also transform State x in to State y. (Ex: 8 Puzzle Problem.)

→ **Commutative Production System**
is a productive system that is both monotonic and partially commutative.

★ **Relationship b/W problem and Production System:-**

There is no relation ship b/W kind of problem and productive system, in formal Sense.



PAGE NO. :
DATE : / /

★

- State Space Search is the process of Searching through a State Space for a Solution by making explicit a Sufficient portion of an implicit State-Space graph to include a goal node.

  - Hence, initially $V = \{S\}$, where S is the Start node.
  - When S is expanded, its Successor are generated and those nodes are added to V and the associated arcs are added to E.

  - This process Continues until a goal node is generated (include in V) and identified (by goal test)

- During Search, a node Can be in one of the three Categories :—

  - Not generated yet (has not been made explicit yet)
  - OPEN : generated but not expanded.
  - CLOSED : expanded

A General Space State Search Algorithm

PAGE NO. :
DATE : / /

open := {S} ; closed := { };

repeat

n := select (open) ; /* Select one node from open for Expansion*/
if n is a goal
then exit with Success ; /* delayed goal Testing */

expand (n)

/* generate all children of n
put these newly generated nodes in
open (check duplicates)
put n in closed

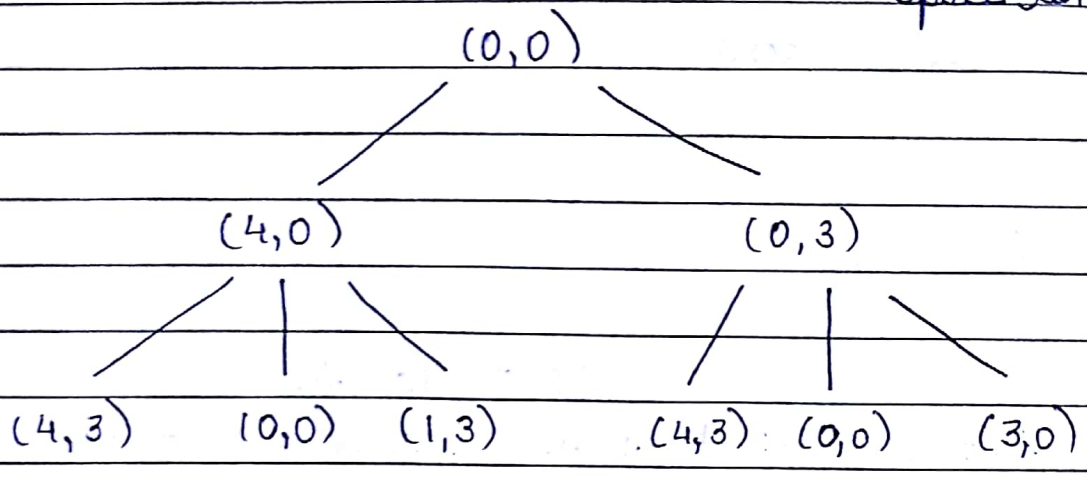
until open = { } ;
exit with failure

DATE : / /

Artifical Intelligance
INT-404

* **Breadth First Search :—**

Water Jug Problem → Large amt. of Space Complexity



Two Levels of a breadth first search Tree.

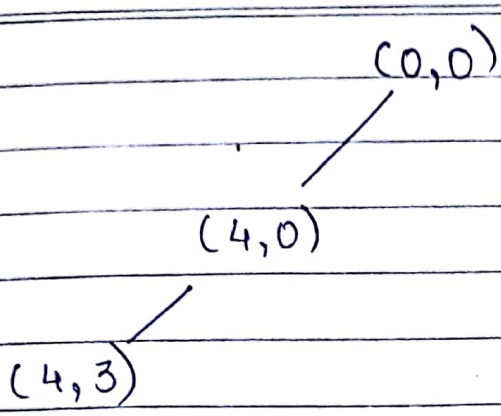
→ Time Complexity = $O(b^d)$

where b is branching factor
d is depth

* **Depth First Search :—**

Disadvantage :— Blind alley { Exploring single branch and there is possibility that we don't get the Answer there).

Example :— Water Jug Problem



Disadvantage :—

→ May not terminate
→ Not Complete
→ does not guarantte us a Solution.

★ Depth-first Iterative Deepening (DFID) :—

• BF and DF both have exponential time complexity $O(b^d)$
BF is complete but have exponential space complexity.
DF is Linear Space Complexity that is incomplete.

• DFID by Korf in 1985
first do DFS to depth 0 (i.e treat start node as having no successors), then, if no solution found, do DFS to depth 1, etc.

until solution found do DFS with depth bound d

$d = d + 1$

Features :—

1. Complete
2. Optimal

★ **Bidirectional Search :—**

→ graph search algorithm that finds a shortest path from initial vertex to a goal vertex in a directional graph.

→ It runs two simultaneous searches : one ~~forward~~ from the initial state, and one backward from the goal, stopping when the two meet in the middle.

1-02-18

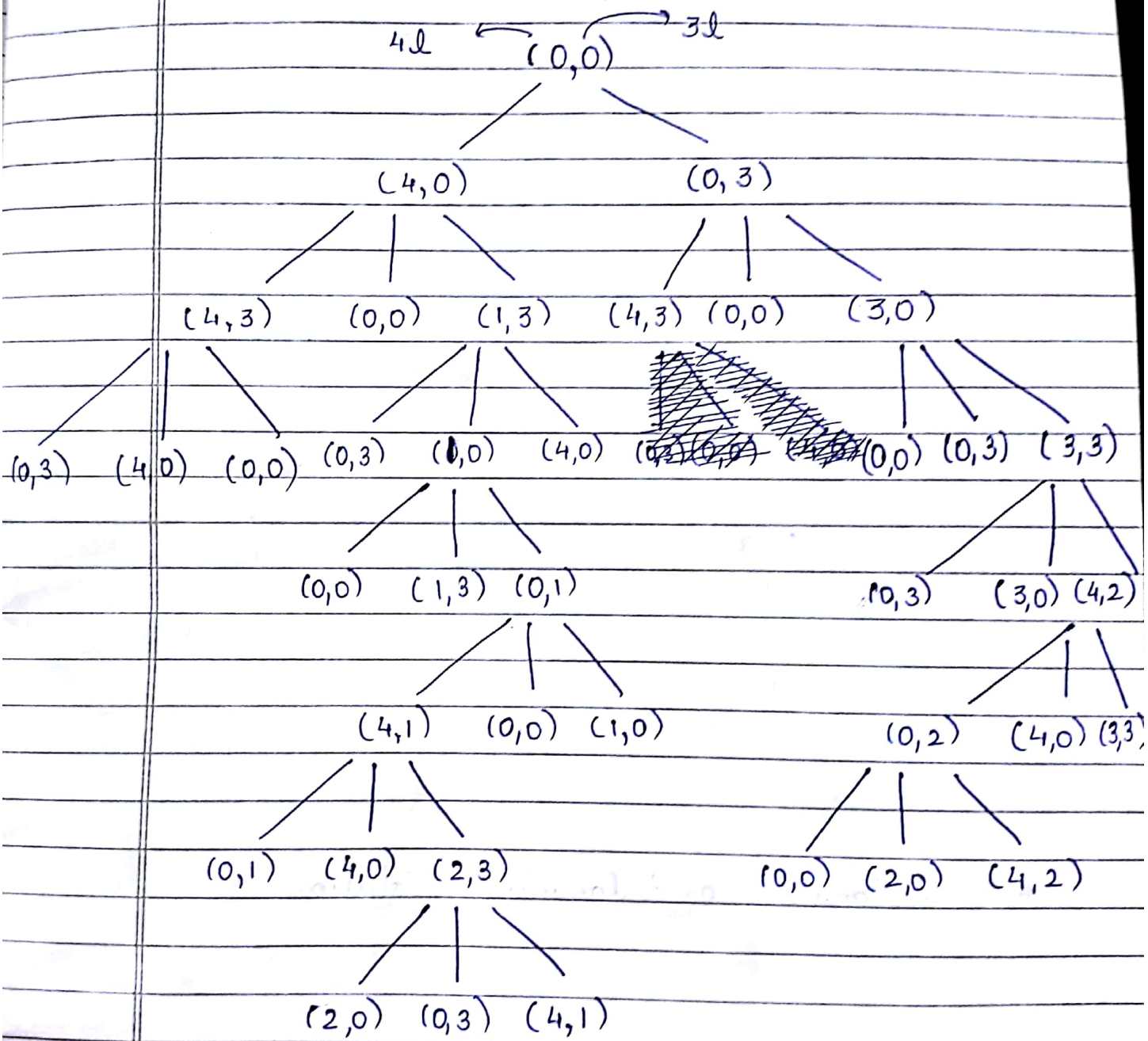
PAGE NO.:
DATE: / /

Tutorial

Artifical Intelligance

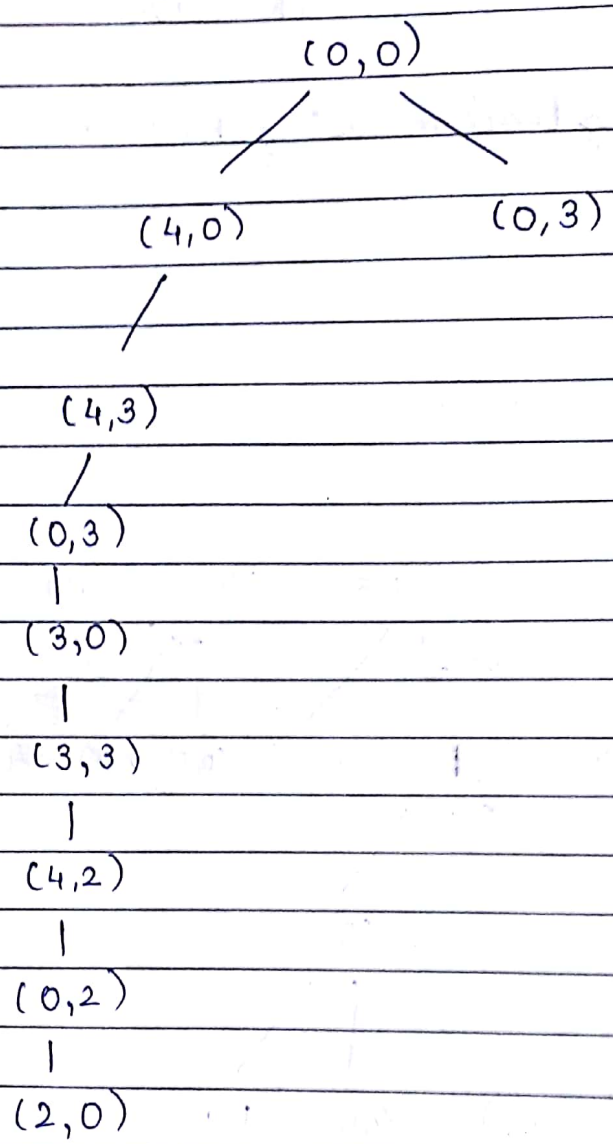
INT-404

★ Water Jug problem using BFS :—



PAGE NO. :
DATE : / /

★ Water Jug Problem using DFS :—



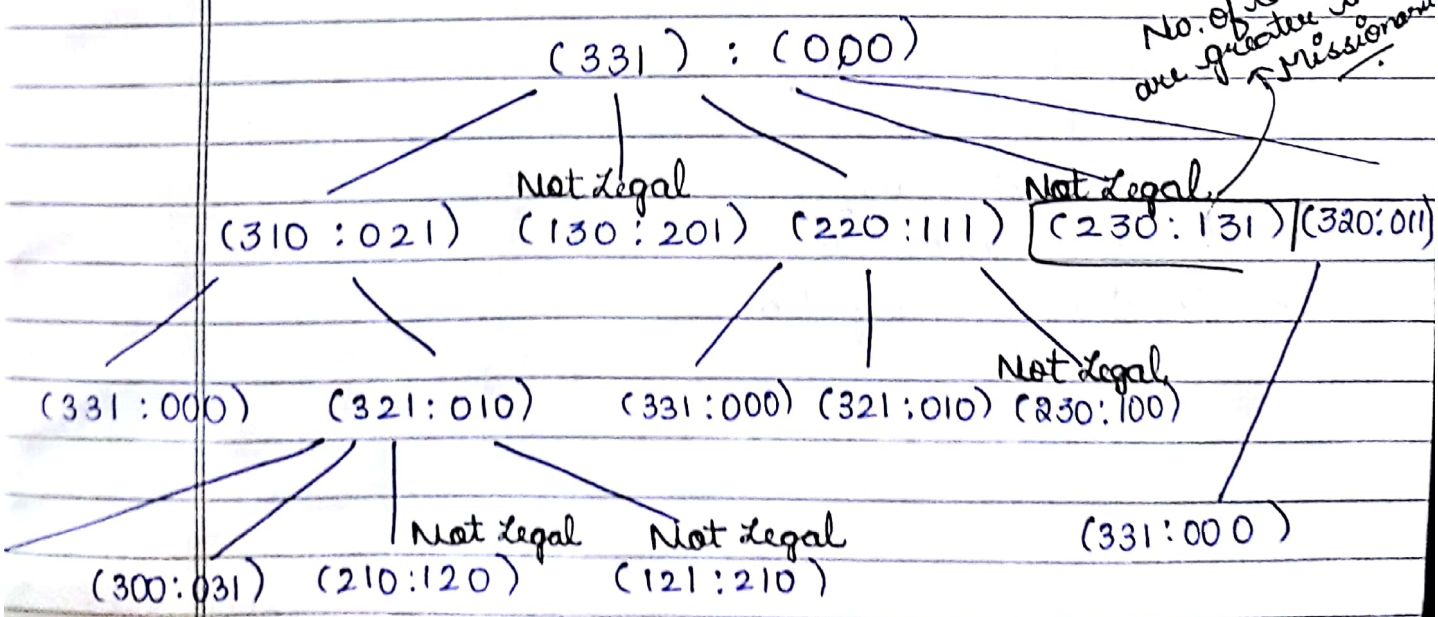
★ Missionaries and Cannibals Solution :—

(220 : 110)

PAGE NO.:
DATE: Near Side

For Side

0.	Initial Setup	MMMCCC	B	
1.	Two Cannibals Cross Over	MMMC	B	CC
2.	One Comes back	MMMCC	B	C
3.	Two Cannibals go over again.	MMM	B	CCC
4.	One Comes back	MMMC	B	CC
5.	Two missionaries Cross	MC	B	MMCC
6.	A missionary & Cannibal return	MMCC	B	MC
7.	Two missionaries Cross again	CC	B	MMMC
8.	A Cannibal returns	CCC	B	MMM
9.	Two Cannibal Cross	C	B	MMMCC
10.	One returns.	CC	B	MMMC
11.	And bring over the third		B	MMMCCC



PAGE NO. :
DATE : / /

Lecture

Artifical Intelligance

INT-404

* Heuristic Functions :—

- is a function that maps from problem state descriptions to measures of desirability, usually represented as numbers.
- Well designed Heuristic function play an important role in guiding a Search process towardas a Solution
- In chess heuristic function should be high whereas in travelling Salesman it is Low.
- The purpose of heuristic func. is to guide the Search process in profitable direction by Suggesting which path to follow first when more than one is available.
- Using good heuristics we can get good possible non optimal Solutions

1. Generate and Test
2. Hill climbing
3. Best First Search
4. Problem reduction
5. Constraint Satisfaction

PAGE NO. :
DATE : / /

★ Generate and Test :—

1. Generate a possible Solution
2. Test to See if this is actually a Solution by Comparing the chose point or the endpoint of the chosen path to the Set of acceptable goal States.
3. If a Solution has been

→ Example :
   1. Finding key of a 3 digit Lock.
   2. 8-Puzzle problem.

→ Inefficient for problems with Large Space.

→ Features of Generate & Test :—

1. Generate Solution randomly.
2. Exhaustive : Consider each Case in depth.
3. Heuristic : not Consider paths that Seem Unlikely to Lead a Solution.
4. Plan

★ Hill Climbing :—

• Hill climbing can be used to Solve problems that have many Solutions but where Some Solutions are better than others.

DATE: / /

- It is often used when a good heuristic function is available for evaluating state but when no other useful knowledge is available.
- Example: In travelling Salesman, Straight Line distance b/w two cities can be a heuristic measure of remaining distance.
- Test function + heuristic function = Hill climbing

* **Simple Hill Climbing :—**

The key difference b/w Simple Hill climbing and Hill climbing is

8-Puzzle Problem (using Simple Hill Climbing)

2	8	3
1	6	4
7		5

Heuristic Value = 4

→

1	2	3
8		4
7	6	5

= 0

2	8	3
1		4
7	6	5

= 3

→

2		3
1	8	4
7	6	5

= 3

PAGE NO. :
DATE : / /

	2	3
1	8	4
7	6	5

= 2

⟶

1	2	3
	8	4
7	6	5

= 1

1	2	3
8		4
7	6	5

= 0

* Steepest - Ascent Hill Climbing :-

→ Consider all the moves from Current State.
→ Selects the best one as the next State.
→ Also known as Gradient Search.

$$h(n) = d(n) + w(n)$$

$d(n)$ ↓ 0

$w(n)$ ↳ No. of tiles misplaced

5-02-18

→ It uses BFS.
→ [illegible]

* Disadvantages of Hill Climbing :-

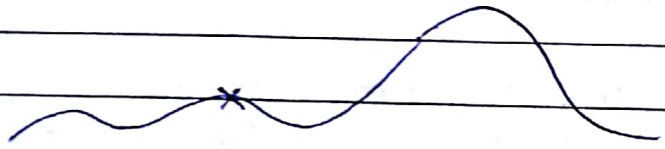
1. Local maximum : A State that is better than all of its neighbours, but not better than some other states far away.

PAGE NO. :
DATE : / /

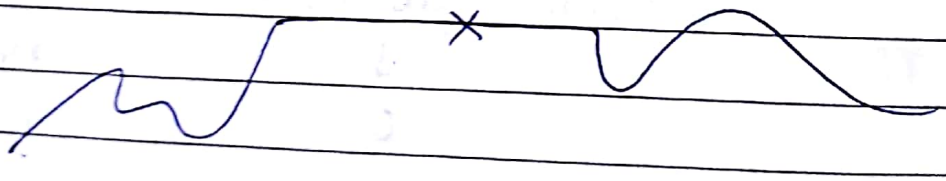
2. Plateau : A flat area of Search Space in which all neighbouring State have the Same Value.

3. Ridge : Special kind of Local maximum. It is an area of Search Space that is higher than the surrounding area and itself has a Slope on but cannot be traversed by Single move in any one direction. Means move in Several direction in one.

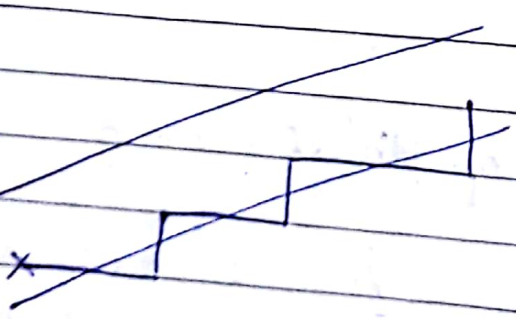
- Local maximum



- Plateau



- Ridge



PAGE NO. :
DATE : / /

To Solve these 3 problems :

1. Backtrack to some earlier node and try going in a different direction. (good way in dealing with Local maxima)

2. Make a big jump to try to get in a new Section. (good way in dealing with plateaus).

3. Moving in several directions at once. (good Strategy for dealing with ridges).

* A Hill climbing Problem

A	H
H	G
G	F
F	E
E	D
D	C
C	B
B	A
Initial State	Goal State

Local : Add one pt. for every block that is resting on thing it is supposed to be resting on. Subtract one point from every block that is sitting on wrong thing.

PAGE NO. :
DATE : / /

-1 ~~-1~~	A	
~~+1~~ +1	H	
+1	G	
+4 = +1	f	
+1	E	
+1	D	
+1	C	
As B was supposed to on A. ~~+1~~ -1	B	

Initial

H	+1
G	+1
f	+1
E	+1 = +8
D	+1
C	+1
B	+1
A	+1

Goal

	H	+1
	G	+1
	f	+1
	E	+1
	D	+1
	C	~~+1~~ +6
A +1	B	-1

A
H
G
f
E
D
C
B

(a)

	G
	f
	E
	D
H	C
A	B

(b)

		G
		f
		E
		D
		C
A	H	B

(c)

point

Global : Add one point for every block in Correct Support Structure, Subtract one point for every block in existing Support Structure.

	-7	A
	-6	H
= -28	-5	G
	-4	f
	-3	E
	-2	D
	-1	C
	0	B

Initial State

H	+7
G	+6
f	+5
E	+4
D	+3
C	+2
B	+1
A	0

= 28

Goal State

	H	-6
	G	-5
	f	-4
	E	-3
	D	-2
	C	-1
A	B	0

= -21

A							
H							
G				G	-5	G	-5
= -28 f				f	-4	f	-4
E				E	-3	E	-3
D				D	-2	D	-2
C		-1	H	C	-1	C	-1
B		0	A	B	0	A H B	0

= -16

= -15

PAGE NO. :
DATE : / /

06-02-18

[Lecture 1]

INT-404

Artifical Intelligence

* Simulated Annealing :—

→ A variation of hill climbing in which at the beginning

→ Probability of transition to higher energy state is given by function :

$$P = e^{-\Delta E/KT}$$ → No. of steps downhill

where

ΔE is the positive change in the Energy Level
T is Temperature
K is Boltzmann Constant

Suppose K = 1

$$P = e^{\Delta E/T}$$

$$\Delta E = h(\text{Current State}) - h(\text{N.S})$$ (N.S → Next State)

Annealing Schedule : If the Temp. is Lowered Sufficiently Slowly, then the goal will be attained.

PAGE NO. :
DATE : / /

★ BEST FIRST SEARCH :—

- OR Graph
- A* Search
- Agenda Search

→ Combines the advantages of both DFS and BFS into a single method.

→ Depth first Search : not all competing branches having to be expanded.

→ Breadth First Search : not getting trapped on dead-end paths.

⇒ Combining the two is to follow a single path at a time, but switch paths whenever some competing path look more promising than the current one.

- General approach of At each step of BFS search process, we select the most promising of the nodes we have generated so far.

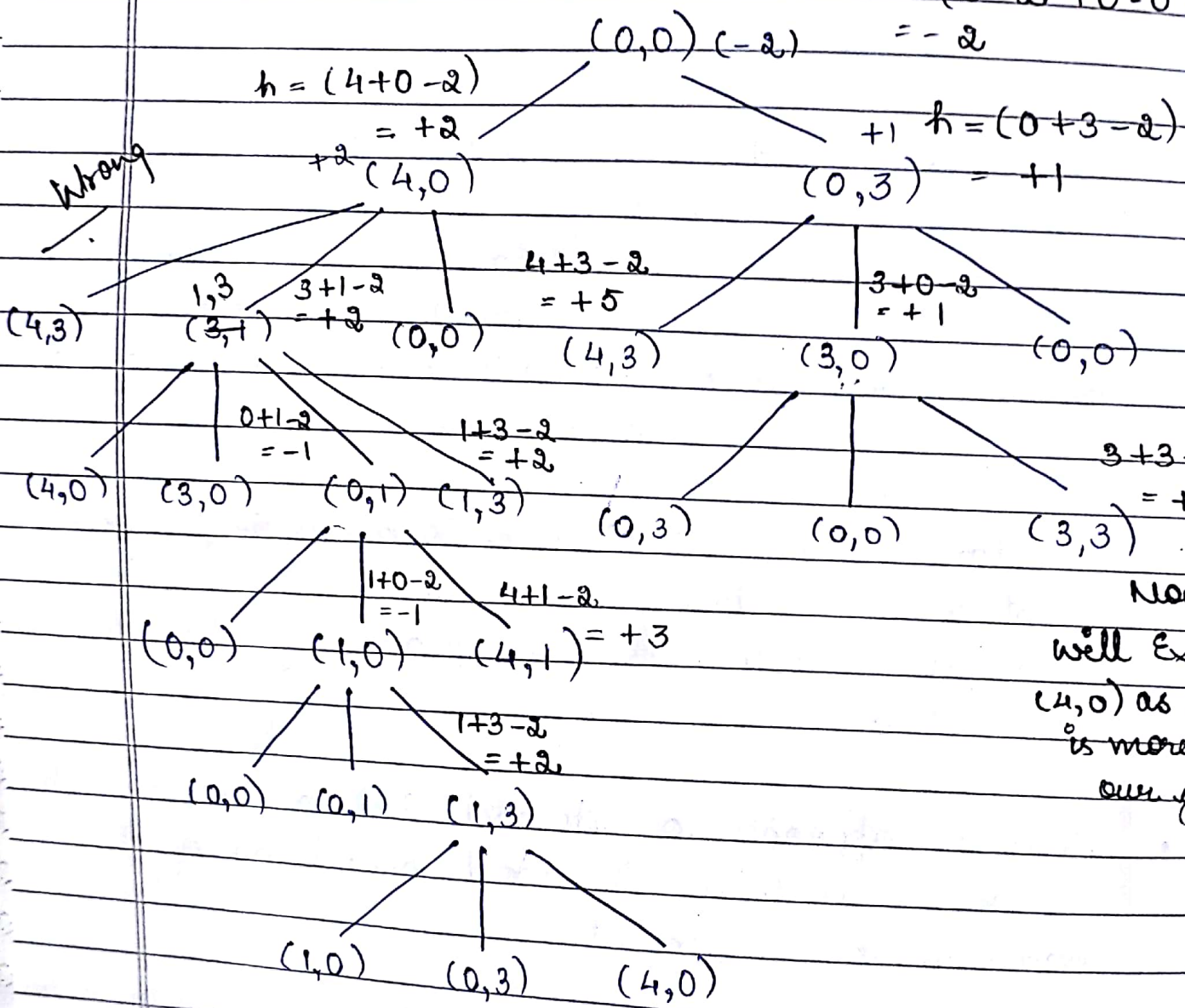
- This is called OR-graph, since each of its branches represents an alternative problem solving path.

Water Jug Problem using BFS

PAGE NO. :
DATE : / /

$h = (x_c - x_g + (y_c - y_g))$

$h = (x_c + y_c - 2)$



Now, we will Expand (4,0) as its heuristic is more close our goal.

PAGE NO. :
DATE : / /

Best First Search Vs Hill Climbing

→ In hill climbing, one move is selected and all the others are rejected. In BFS, one move is selected but

→ OPEN : nodes that have been generated but have not examined. This is organized as Priority queue.

→ CLOSED :

★ **A* SEARCH :—**

Evaluation function : $f(n) = g(n) + h(n)$

- $g(n)$ the cost to reach the node.
- $h(n)$ estimated cost to get from node to the goal.
- $f(n)$ estimated total cost of path through n to goal.

PAGE NO. :
DATE : / /

08-02-18

Tutorial

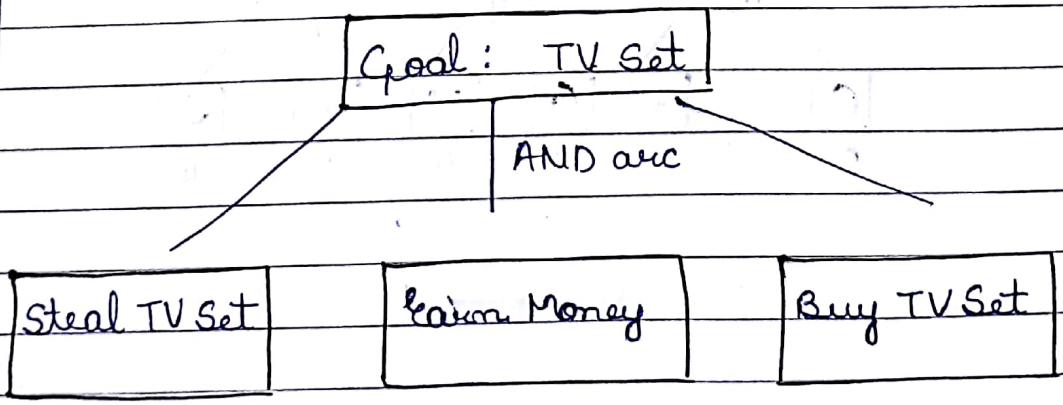
Artifical Intelligance
INT-404

★ **AND OR Graph :-**

When a problem can be divided into a set of sub problems, where each sub problem can be solved separately and a combination of these will be a solution, AND-OR graphs or AND-OR trees are used for representing the solution.

The decomposition of the problem or problem reduction generates AND arcs. One AND arc may point to any number of successor node.

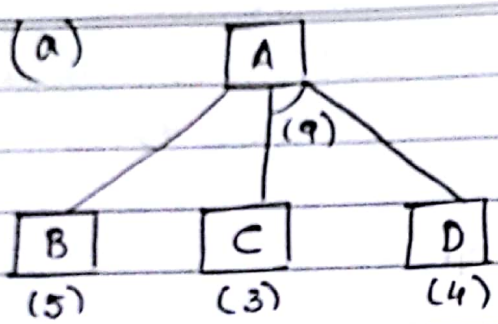
All these must be solved so that the arc will rise to many arcs, indicating several possible solutions. Hence the graph is known as AND-OR instead of AND.

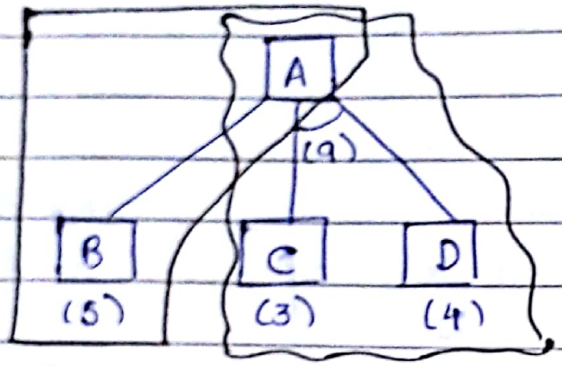


PAGE NO. :
DATE : / /

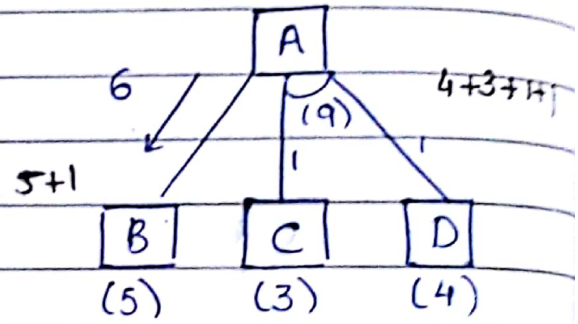
Example :

(a)





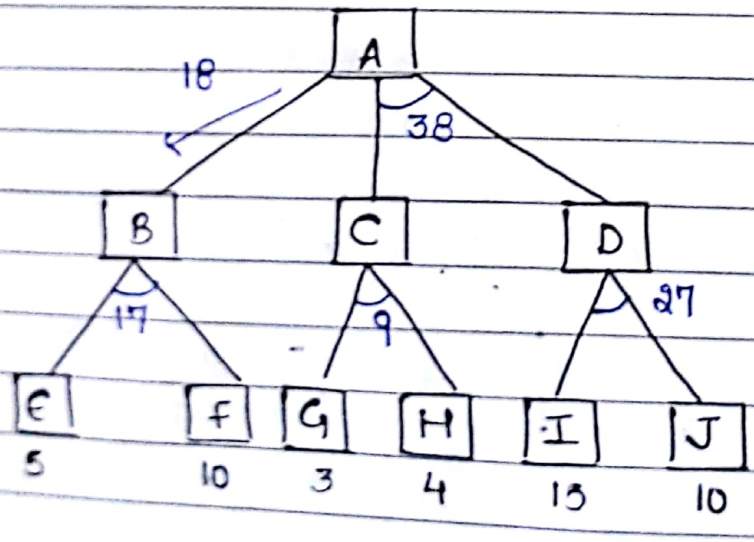
OR



AND

We will expand B as it having 6.

(b)



PAGE NO. :
DATE : / /

* Constraint Satisfaction :—

→ The goal is to discover some problem state that satisfies a given set of constraints. Ex: Puzzles, real world perceptual labeling problems, design tasks etc.

→ Helps in reducing the amt. of search required as compared with a method that attempts to form partial solutions directly by choosing specific value for components of the eventual solutions.

(a)

$$\begin{array}{r} {}^{1}\,{}^{9}L\ O\ {}^{4}G\ {}^{2}I\ C \\ +\ {}^{1}\,{}^{9}L\ O\ {}^{4}G\ {}^{2}I\ C \\ \hline P\ {}^{8}R\ O\ {}^{9}I\ O\ {}^{4}G \end{array}$$

Savvy

So P = 1

O = 1
I = 5
C = 2
G = 4
L = 9
R = 8

(i) No two letters can have same digits.
(ii) Should satisfy Arithmetic operations.

(b)

$$\begin{array}{r} C_4\ C_3\ C_2\ C_1 \\ S\ E\ N\ D \\ +\ M\ O\ R\ E \\ \hline M\ O\ N\ E\ Y \end{array}$$

Savvy

Assuming the savvy C_1, C_2, C_3, C_4

$C_2 + E + 0 = N$

$C_2 + E = N$

$C_2 \neq 0$ as E can't be equal to N

$C_2 = 1$

$C_1 + N + R = E$

$C_1 + 3 + R = 12$

$C_1 = 0$, $R = 9$

$C_1 = 1$, $R = 8$

$R = 8, 9$

$M = 1$

$S = 8, 9$

$O = 0$

$E = 2$ (assumption)

13-02-18

Lecture

PAGE NO. :
DATE : / /

Artifical Intelligance
INT-404

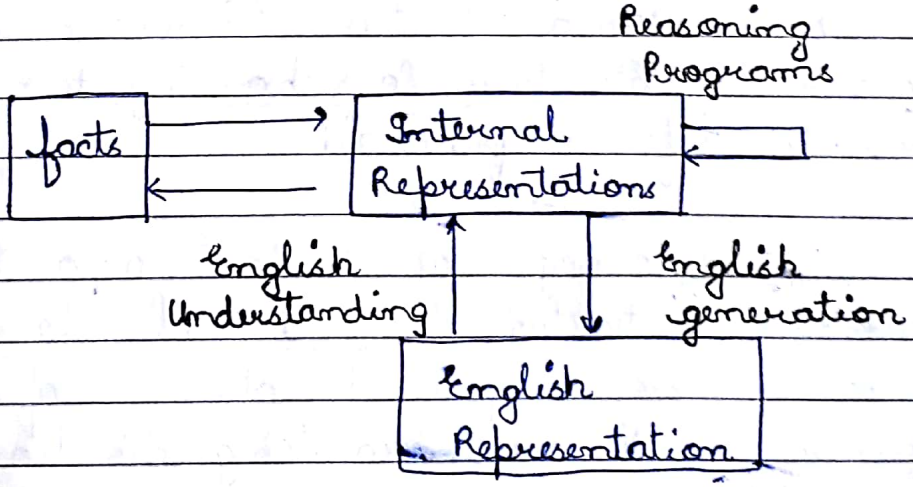
* **Knowledge Representation :—**

Knowledge can be defined as body of facts and principles accumulated by human-kind or the act, fact or state of knowing.

Knowledge can be of following types

1. Declarative (Statements)
2. Procedural (facts, rules, function)
3. Heuristic (rule of thumb/ experiance)

→ **Mapping b/W facts and Representation**



Facts : Truths in some relevant world. These are the things we want to represent.

PAGE NO. :
DATE : / /

1. Spot is a dog → dog (Spot)
2. All dogs have tails → $\forall x : dog(x) \rightarrow hastail(x)$
3. Then using the deduction method of Logic, we may generate the new representation object : has tail (Spot)

4. So we can, using backward mapping function : Spot has a tail.

2 marks :→ A good System for the representation of knowledge should have following properties :—

1. Representational Adequacy : the ability to represent all of the kinds of knowledge that are needed in that domain.

2. Inferential Adequacy : The ability to manipulate the representational structures in such a way as to derive new structures corresponding to new knowledge inferred from old.

3. Inferential Efficiency : ability to incorporate into knowledge structure the additional information that can be used to focus the attention of inference mechanism in the most promising direction.

4. Acquisitional Efficiency : Acquiring new information easily using automatic methods wherever possible rather than reliance on human intervention.

PAGE NO. :
DATE : / /

4 types of knowledge representation :-

1. Simple Relational knowledge

Player	Height	Weight	Runs
Dhoni	6-0	150	3000
Sachin	5-4	140	15000
Zaheer	6-2	160	1000

This even does not tell us who is heavist player

2. Inheritable knowledge

→ It is possible to enhance the basic representation with inference mechanism that operate on the structure of representation.

→ The most useful form of inference is Property Inheritance, in which elements of specific classes inherit attributes and values from ~~some specific~~ one general classes in which they are included.

15-02-18

Tutorial

Artificial Intelligence
INT-404

$C_3\ C_2\ C_1$

1. $\begin{array}{r} TWO \\ +\ TWO \\ \hline FOUR \end{array}$

1. $F = 1$
2. $O \neq 0$ as then R will also be Zero.
   $O \neq 1$ as F is 1.
   $O = 2$ So R is 4.
3. $C_2 + 2T = 12$
   $\downarrow$
   $0 \qquad T = 6$

4. Assume $O = 4$ 3. $R = 8$
   $C_2 + 2T = 14$
   $\downarrow$
   $0 \qquad T = 7$

5. $C_1 + W + W = U$
   $\downarrow$
   $0 \qquad W + W = U$

   $W = 3 \quad , \quad U = 6$

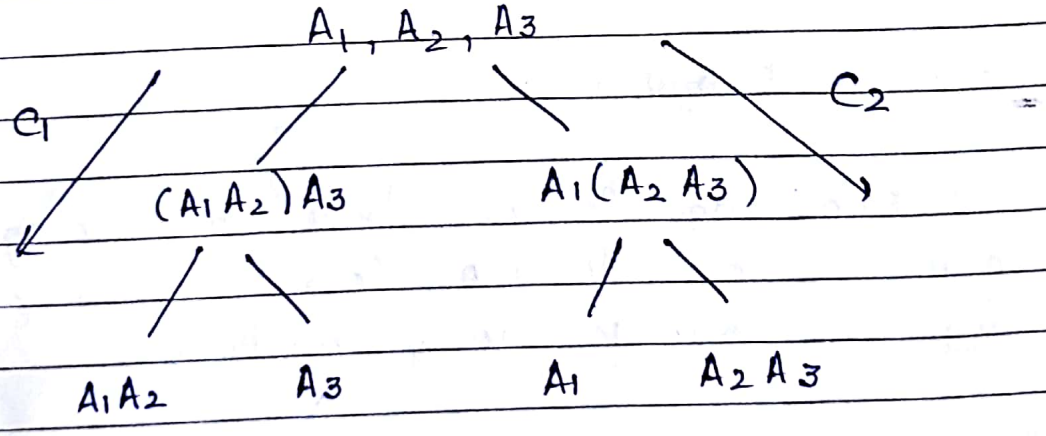
PAGE NO. :
DATE : / /

2. ~~$C_4 C_3 C_2 C_1$~~

$$\begin{array}{r} E\ A\ T \\ +\ T\ H\ A\ T \\ \hline A\ P\ P\ L\ E \end{array}$$

1. $A = 1$ as it is a carry.
2. $C_1 + A + A = L$
   $C_1 + 1 + 1 = L$
   $L = 2 + C_1$
   $C_1 + 2 = L$
   $\boxed{3 = L}$
3. $T = 9$
4. $T + T = E$
   $9 + 9 = E$
   $18 = E \quad (C_1 = 1)$
   $E = 8$
5. $P = 0$
6. $H = 2$

★ AO* :-



PAGE NO. :
DATE : / /

Lecture

19-02-18

Artifical Intelligence
INT - 404

Knowledge Representation

→ 4 Types of knowledge Representation :—

3. Inferential knowledge

Sometimes all the power of traditional Logic is necessary to describe the inferences that are needed. We can represent inferential knowledge about a domain using first order Predicate Logic.

1. Marcus was a man.
man (marcus)

2. Marcus was a Pompeian.
Pompeian (marcus)

4. Procedural knowledge

- This knowledge can be represented in programs in many ways. The most common way is simply as code for doing something.
- The machin uses the knowledge when it executes the code to perform a task.

Imp. 10 Marks

PAGE NO. :
DATE : / /

* Issues in Knowledge Representation :-

1. Important Attributes

→ There are two attributes that are of very general significance, and we have already seen their use; instance and isa.

→ Attributes are imp. bcoz they support the property inheritance.

2. Relationship among Attributes

properties as :

- Inverse
- Existance in an isa hierachy
- Single-Valued Attributes

3. Choosing the Granularity of the Representation.

It is necessary to answer the question "At what level of detail should the world be Represented?".

Another way this question is phrased is "what should be our primitives?"

PAGE NO.:
DATE: / /

20-02-18

Lecture 1

Artifical Intelligence
INT-404

★ Issues in knowledge Representation :—

4. Representing Set of objects

It is imp. to be able to represent sets of objects for several reasons :

- There are some properties that are true of sets that are not true of the

Two ways to represent the object :

1. Extensional : List the members
2. Intensional : provide a rule that, when a particular object is evaluated return true or false depending the object is in the set or not.

Ex :
Extensional description of the set of our Sun's planets on which people live is {earth} and intensional description is :

$$\{x : \text{Sun-planet}(x) \wedge \text{human-inhabited}(x)\}$$

chapter - 5

PREDICATE LOGIC

* Propositional Logic & Predicate Logic :—

1. Socrate is a man

   Man Socrate (Propositional Logic)

2. Plato is a man

   Man Plato (Propositional)

3. All men are mortal

   Mortal Man (Propositional)

1. Socrate is a Man → Man (Socrate)
2. Plato is a Man → Man (Plato)
3. All men are mortal → Mortal (Socrate)

} Predicate Logic

Propositional	Predicate
1. Represent using Variable.	1. Represent using classes and instance.
2. New knowledge cannot be easily generated.	2. New knowledge is easily generated
3. No Quantifiers	3. Quantifiers $\forall$ → For all $\exists$ → There exist

→ Some Simple facts in Propositional Logic :—

1. It is raining
   RAINING
2. It is Sunny
   SUNNY
3. It is Windy
   WINDY
4. If it is raining, then it is not Sunny.
   RAINING → ¬ SUNNY

PAGE NO. :
DATE : / /

$\wedge$ → and
$\vee$ → or
$\neg$ → negation
$\rightarrow$ → implies

→ Predicate Logic :—

also called First order predicate Logic or Simply First order Logic or Predicate Calculus or FOPL.

It is more versatile than propositional counterpart for its added features.

→ Limitations of Propositional Logic :—

Socrates is a man
SOCRATE MAN

Plato is a man
PLATOMAN

} Can't able to draw similarities b/W Socrates and Plato.

Better representation :

MAN(SOCRATE)
MAN (PLATO)

All men are mortal
MORTAL MAN

} fail to capture relationship b/W any individual being a man and that individual being mortal.

$\forall x$: man(x) → mortal (x)

Predicate Logic Example

1. Marcus is a man.
2. Marcus was a pompeian
3. All pompeian were romans.
4. Caesar was a ruler
5. All romans were either Loyal to Caesar or hated him.
6. Everyone is Loyal to Someone.
7. People only try to assassinate ruler they are not Loyal to.
8. Marcus tried to assassinate Caesar.

1. Man (Marcus)
2. Pompeian (Marcus)
3. $\forall x$: pompeian $(x) \rightarrow$ Romans (x)
4. Ruler (Caesar)
5. $\forall x$: Romans $(x) \rightarrow$ Loyal to (Caesar, x) $\vee$ hated (Caesar, x)
6. $\forall x, \exists y$: $\rightarrow$ Loyal to (x, y)
7. $\forall x : \forall y$: people $(x) \wedge$ ruler $(y) \rightarrow$ try to assassinate $(x, y) \wedge \neg$ Loyal to (x, y)
8.

PAGE NO. :
DATE : / /

Tutorial

02-18

Artifical Intelligence
INT-404

1. All hounds howl at night.
2. Anyone who has any cats will not have any mice.
3. Light sleepers do not have anything which howls at nights.
4. John has either a cat or a hound.
5. If John is a Light Sleeper then John does not have any mice.

1. $\forall x : hound(x) \rightarrow howl(x)$.

2. $\forall x \forall y : have(x, y) \wedge cat(y) \rightarrow \forall z \neg (have(x, z) \wedge mouse(z))$

3. $\forall x : LightSleeper(x) \rightarrow \neg \exists y (have(x, y) \wedge howl(y))$

4. $\exists x : have(John, x) \wedge cat(x) \vee hound(x)$

5. $LightSleeper(John) \rightarrow \neg \exists x (have(John, x) \wedge mouse(x))$

PAGE NO.: _____
DATE: / /

1. Anyone whom mary loves is a football star.
2. Any Student who does not pass does not play.
3. John is a Student.
4. Any Student who does not Study does not pass.
5. Anyone who does not play is not a football star.
6. If John does not Study then mary does not Love John.

1. $\forall x$: Loves (Mary, x) $\rightarrow$ Football Star (x).
2. $\forall x$: Student (x) $\wedge$ $\neg$ pass (x) $\rightarrow$ $\neg$ play (x).
3. Student (John).
4. $\forall x$: Student (x) $\wedge$ $\neg$ Study (x) $\rightarrow$ $\neg$ pass (x).
5. $\forall x$: $\neg$ play (x) $\rightarrow$ $\neg$ football Star (x).
6. $\neg$ Study (John) $\rightarrow$ $\neg$ Love (Mary, John).

1. Every child Loves Santa.
2. Everyone who Loves Santa Loves Reindeer.
3. Raudolf is an reindeer and Raudolf has a red nose.
4. Anything which has a Red nose is ~~Bored~~ weird or clown.
5. No Reindeer is a clown.
6. Scrooge does not Love anything which is weird.
7. Scrooge is not a child.

PAGE NO. :
DATE : / /

1. $\forall x$: child $(x) \rightarrow$ Loves $(x,$ Santa$)$

2. $\forall x$: Loves $(x,$ Santa$) \rightarrow \forall y$: Reindeer $(y) \wedge$ Loves (x, y)

3. Reindeer (Rudolf) $\wedge$

4. $\forall x$: has (red nose) $\rightarrow$ Weird $\vee$ clown

5. [illegible]

6. $\forall x$: Loves (Scrooge, x) $\rightarrow$ Weird

7. $\neg$ child (Scrooge)

1. No mortal Lives Longer than 150 years.
2. All Romans died when volcano erupted in 1979 AD.

1. $\forall x$: Mortal $(x) \wedge$ born $(x, t_1) \wedge$ gt $((t_2 - t_1), 150)$ $\rightarrow$ dead (x)

(gt: greater than)

2. $\forall x$: Roman $(x) \rightarrow$ dead $(x, 1979)$

3. Alive means not dead.

$\forall x : \forall t$: alive $(x, t) \rightarrow \neg$ dead (x, t)

Lecture.

PAGE NO. :
DATE : / /

* Clause Normal form (CNF) :-

1. Eliminate $\rightarrow$, using : $a \rightarrow b = \neg a \vee b$
2. Reduce the Scope of each $\neg$ to a Single term using :

   - $\neg (\neg P) = P$
   - de Morgan's Laws : $\neg (a \wedge b) = \neg a \vee \neg b$
     $\neg (a \vee b) = \neg a \wedge \neg b$
   - $\neg \forall x : P(x) = \exists x : \neg P(x)$
   - $\neg \exists x : P(x) = \forall x : \neg P(x)$

3. Standardize variables.
4. Move all quantifiers to the Left of the formula without changing their relative order.
5. Eliminate existential quantifiers by inserting Skolem Fun.
6. Drop the Prefix.
7. Convert the matrix into a Conjuction of disjuncts, using associativity and distributivity.
8. Create Separate clauses for each Conjunct.
9. Standardize apart the variable in the Set of clauses generated in Step 8, using the fact that

$$(\forall x : P(x) \wedge Q(x)) = \forall x : P(x) \wedge \forall x : Q(x)$$

$\forall x : Student(x) \wedge pass(x) \rightarrow play(x)$

Step 1: $\forall x : \neg Student(x) \wedge \neg pass(x)) \vee \neg play(x)$

Step 2: $\forall x : \neg Student(x) \vee \neg(\neg(pass(x) \vee \neg play(x))$

Step 3: $\forall x_1 : \neg Student(x_1) \vee pass(x_1) \vee \neg play(x_1)$

Step 5: Example :

$\exists y : President(y)$

will be converted to

$President(S1)$

While

$\forall x : \exists y : father\text{-}of(y, x)$

will be converted to

$\forall x : father\text{-}of(S2(x), x))$

Lecture

26-02-18 Artifical Intelligence

PAGE NO. :
DATE : / /

1. Marcus was a man.
2. Marcus was a pompeian.
3. All pompeians were Romans.
4. All Romans were either Loyal to Caesar or hated him.
5. Caesar was a ruler.
6. Everyone is Loyal to Someone.
7. People Only try to assassinate ruler they are not Loyal to.
8. Marcus tried to assassinate Caesar.

1. man (Marcus)
2. Pompeian (Marcus)
3. $\forall x$: Pompeian (x) $\longrightarrow$ Roman (x)

$\boxed{\neg \text{Pompeian}(x_1) \vee \text{Roman}(x_1)}$ } In CNF.

4. ruler (Caesar)
5. $\forall x$: Roman (x) $\longrightarrow$ Loyal to (Caesar, x) $\vee$ hate (Caesar, x)

$\forall x \neg$ Roman (x) $\vee$ Loyal to (Caesar, x) $\vee$ hate (Caesar, x)
$\forall x_2 \neg$ Roman (x_2) $\vee$ Loyal to (Caesar, x_2) $\vee$ hate (Caesar, x_2)
$\boxed{\neg \text{Roman}(x_2) \vee \text{Loyal to}(\text{Caesar}, x_2) \vee \text{hate}(\text{Caesar}, x_2)}$

6. $\forall x \exists y$: Loyal to (x, y)

$\forall x_3 \exists y_1$ Loyal to (x_3, y_1)
$\boxed{\forall x_3 \text{ Loyal to } [x_3, f(x_3)]}$

7. $\forall x \forall y$: People $(x) \wedge$ ruler (x) $\wedge$ Trytoassassinate $(x,y) \wedge \rightarrow \neg$ Loyal to (x,y).

$\forall x \forall y : \neg$ [People $(x) \wedge$ ruler (x) $\wedge$ Trytoassassinate $(x,y)] \vee \neg$ Loyal to (x,y)

$\forall x_4 \forall_2 : \neg$ [People $(x_4) \wedge$ ruler $(x_4) \wedge$ Trytoassassinate $(x_4, y_2)] \vee \neg$ Loyal to (x_4, y_2)

$\rightarrow \neg$ People $(x) \vee \neg$ ruler $(x) \vee \neg$ Trytoassassinate $(x,y) \vee \neg$ Loyal to (x,y)

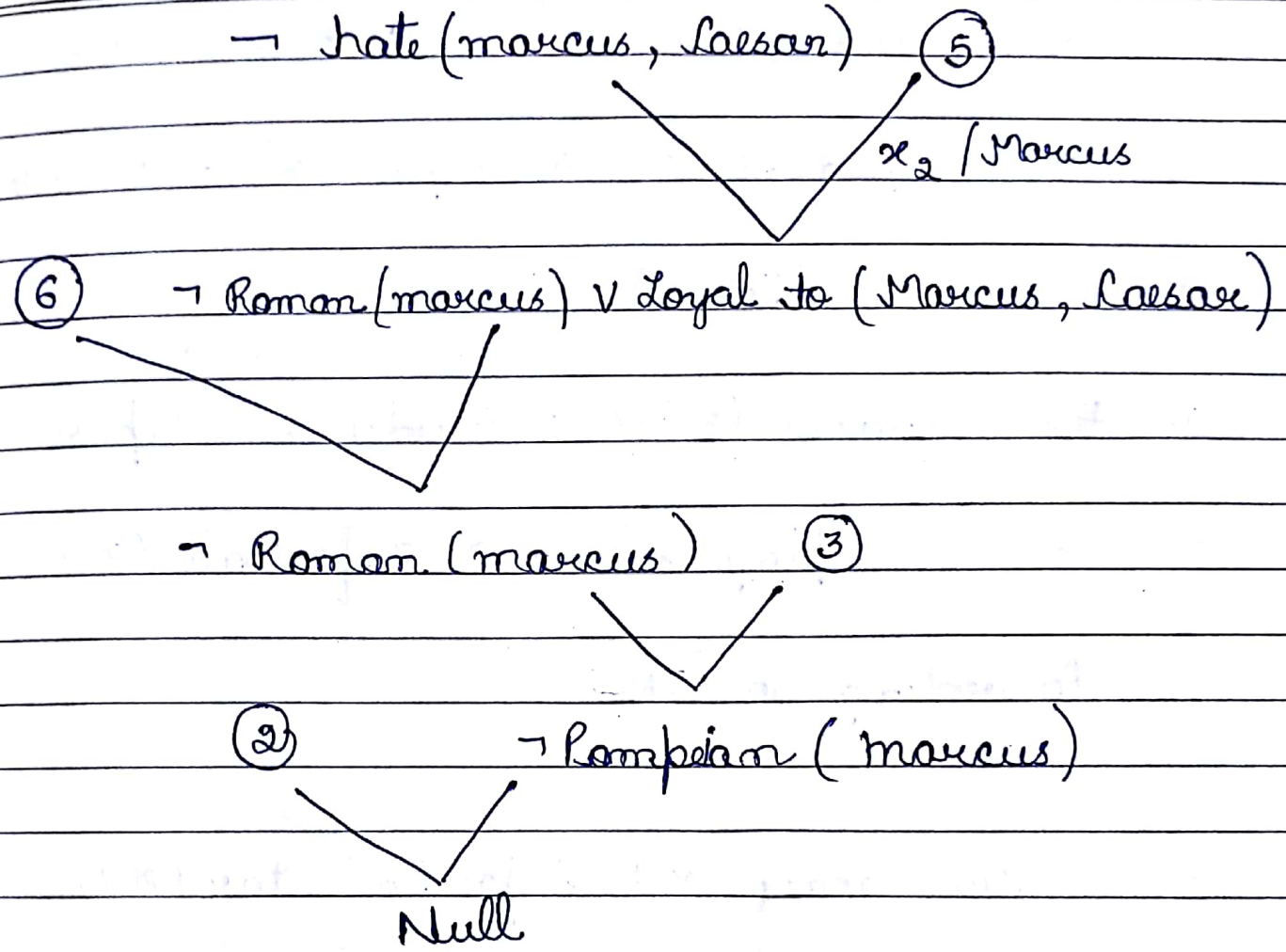
$\neg$ People $(x_4) \vee \neg$ ruler $(x_4) \vee \neg$ Trytoassassinate (x_4, y_1) $\vee \neg$ Loyal to (x_4, y_1).

8. Trytoassassinate (Marcus, Caesar)

Prove: Marcus hate Caesar
hate (Marcus, Caesar)

We will start by assuming the contradiction part.

Resolution Proof



(2.) 1. Anyone whom mary loves is a football star.
2. Any Student who does not pass do not play.
3. John is a Student.
4. Any Student who does not Study does not pass.
5. Anyone who does not play is not a football Sta
6. If John does not Study then mary does not Love John.

- Prove : If John does not Study, then Mary does not Love John.

PAGE NO.:
DATE: / /

Predicate Logic

1. $\forall x : Loves(Mary, x) \rightarrow Football\ Star(x)$.
2. $\forall x : Student(x) \wedge \neg pass(x) \rightarrow \neg Play(x)$
3. $Student(John)$
4. $\forall x : Student(x) \wedge \neg Study(x) \rightarrow \neg pass(x)$
5. $\forall x : \neg play(x) \rightarrow \neg football(x)$

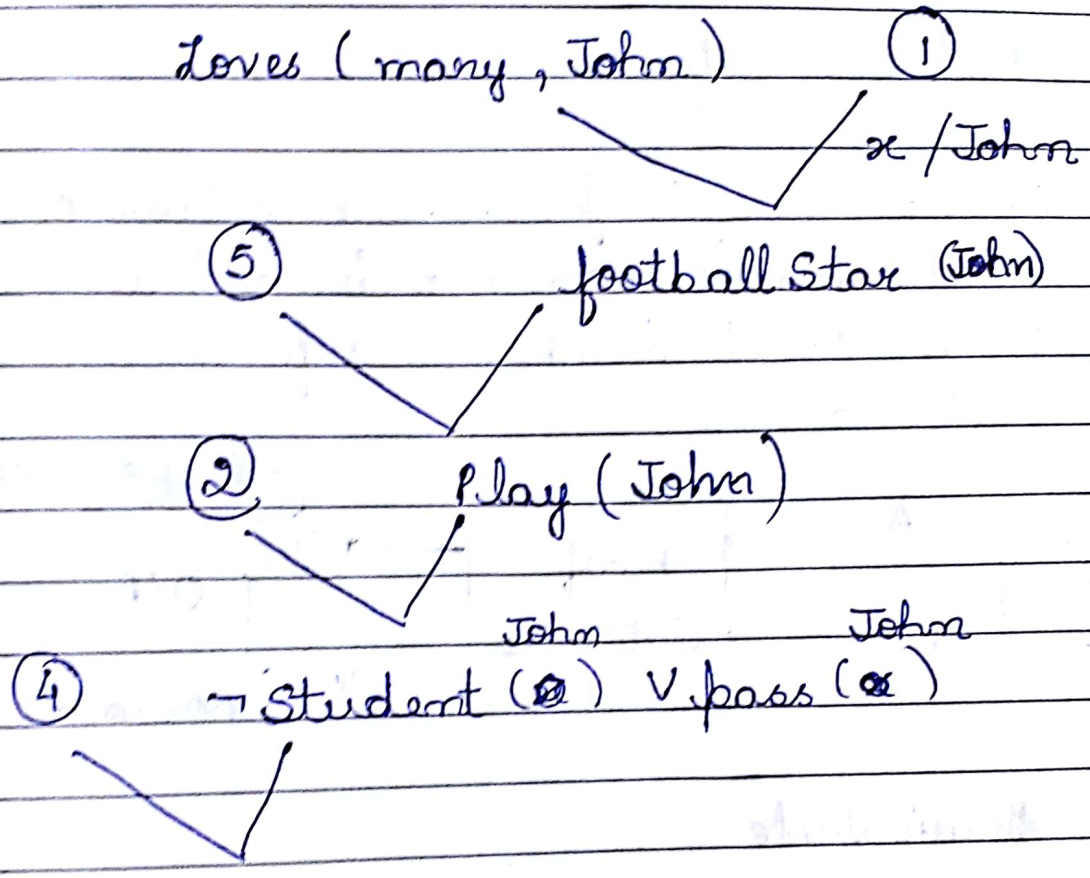
Converting to CNF

1. $\neg Love(mary, x) \vee football\ Star(x)$.
2. $\neg (Student(x) \wedge \neg pass(x)) \vee \neg Play(x)$
   $\neg Student(x) \vee pass(x) \vee \neg Play(x)$
3. $Student(John)$
4. $\neg (Student(x) \wedge \neg Study(x)) \vee \neg pass(x)$
   $\neg Student(x) \vee Study(x) \vee \neg pass(x)$
5. $\neg (\neg play(x)) \vee \neg football\ Star(x)$
   $play(x) \vee \neg football\ Star(x)$

PAGE NO. :
DATE : / /

Resolution Proof

Prove : ¬ Loves (mary, John)



PAGE NO. :
DATE : / /

Lecture

Artifical Intelligence
INT-404

1. Anyone who rides a Harley is a rough character.
2. Every biker rides either Harley or bmw.
3. Anyone who rides a BMW is a yuppie.
4. Every yuppie is a Lawyer.
6. Mary is a nice girl and John is a biker.
7. Prove : If John is not a Lawyer then mary doesnot date John.

Predicate Logic

1. $\forall x$: rides (x, Harley) $\rightarrow$ rough character (x)
2. $\forall x$: biker (x) $\rightarrow$ rides (x, Harley) $\vee$ rides (x, BMW)
3. $\forall x$: rides (x, BMW) $\rightarrow$ yuppie (x).
4. $\forall x$: yuppie (x) $\rightarrow$ Lawyer (x)
5. Any nice girl doesnot date anyone who is a rough character.
   $\forall x \forall y$: Nice girl (x) $\wedge$ $\neg$ date (x, y) $\rightarrow$ RC (y)

   OR

6. $\forall x \forall y$: Nice girl (x) $\wedge$ RC (y) $\rightarrow$ $\neg$ date (x, y)
6. Nice girl (Mary) $\wedge$ biker (John)
7. $\neg$ Lawyer (John) $\rightarrow$ $\neg$ date (mary, John)

PAGE NO. :
DATE : / /

- **CNF Conversion**

1. $\neg$ rides (x, harley) V rough character (x).
2. $\neg$ biker (x_2) V rides (x_2, harley) V rides (x_2, BMW)
3. $\neg$ rides (x_3, BMW) V yuppie (x_3)
4. $\neg$ yuppie (x_4) V Lawyer (x_4).
5. $\neg$ (NC(x_5) $\wedge$ RC(y_1)) V $\neg$ date (x_5, y_1)

   $\neg$ NC(x_5) V $\neg$ RC(y_1) V $\neg$ date (x_5, y_1)

6. Nice girl (Mary) $\wedge$ biker (John)
7. $\neg$ ($\neg$ Lawyer (John)) V $\neg$ date (Mary, John)
   Lawyer (John) V $\neg$ date (Mary, John)

- **Resolution Proof by forward Reasoning**

$\neg$ NC(Mary) V $\neg$ RC(John) V $\neg$ date (mary, John) (6)

(2) $\neg$ RC(John) V $\neg$ date (mary, John) $\wedge$ biker (John)

$\neg$ RC(John) V $\neg$ date (mary, John) V rides (John, harley)
V rides (John, BMW)

$\neg$ RC (John) $\neg$ date (M, J) V rides

PAGE NO :
DATE : / /

chapter - 6

* Representing Knowledge For Rules :—

→ Procedural V/S Declarative knowledge

Declarative knowledge is one in which knowledge is specified but the use to which that knowledge is to be put in, is not given.

Procedural knowledge is one in which the control information that is necessary to use the knowledge is considered to be embedded in the knowledge itself.

Declarative knowledge is defined as the factual information stored in memory and known to be static in nature.

Procedural is the knowledge of how to perform or how to operate.

→ Logic Programming

is a programming language para

1-03-18

Tutorial

PAGE NO. :
DATE : / /

Artifical Intelligence
INT-404

1. John Likes all kinds of food.
2. Apples are food.
3. chicken is food.
4. Anything anyone eats and is not killed by food.
5. Bill eats peanuts and is still alive.
6. Sue eats everything bill eats.

Prove : John Likes peanuts. Likes (John, Peanuts)

Predicate Logic

1. $\forall x$: food $(x) \rightarrow$ Likes (John, x)
2. food (Apple)
3. food (chicken)
4. $\forall x \forall y$: eats $(x, y) \wedge \neg$ killed by$^{(x)} \rightarrow$ food (y)
5. eats (Bill, peanuts) $\wedge$ alive (Bill)
6. $\forall x$: eats (Bill, x) $\rightarrow$ eats (Sue, x)

CNF Conversion

1. $\neg$ food (x) V Likes (John, x)
2. food (Apple)
3. food (chicken)
4. $\neg$ [eats (x, y) $\wedge$ $\neg$ killed (x) V food (y)]

$\neg$ eats (x, y) V killed (x) V food (y)
$\neg$ not alive (x)

5. (a) eats (Bill, Peanuts)
   (b) alive (Bill)

6. $\neg$ eats (Bill, x) V eats (Sue, x)

Resolution Proof

(1) $\neg$ Likes (John, Peanuts)

x / Peanuts

$\neg$ food (Peanuts) (4)

y / Peanuts

$\neg$ eats (x, Peanuts) V killed (x) Will backtrack

(5a) $\neg$ eats (x, Peanuts) V $\neg$ not alive (x)

x / Bill

$\neg$ not alive (x) 5(b)

x / Bill

Null

PAGE NO.:
DATE: / /

1. Every child Loves Santa.
2. Every who Loves Santa Loves Reindeer.
3. Raudolf is an reindeer and Raudolf has a red nose.
4. Anything which has a Red nose is weird or clown
5. No Reindeer is a clown.
6. Scrooge does not Love anything which is weird.

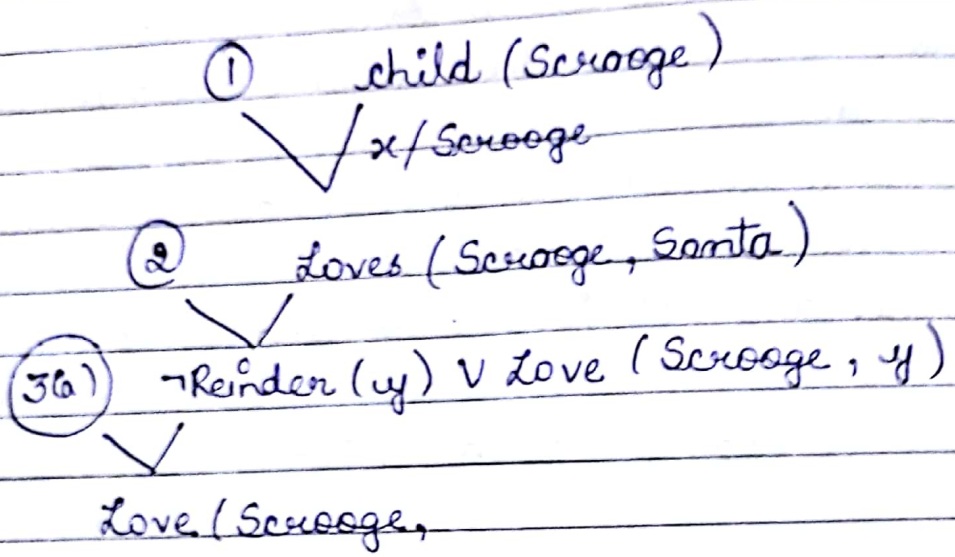
Prove: Scrooge is not a child.

1. $\forall x$: child $(x) \longrightarrow$ Loves $(x,$ Santa$)$
2. $\forall x$: Loves $(x,$ Santa$) \longrightarrow \forall y$: Reindeer $(y) \wedge$ Loves (x,y)
3. ~~Reindeer (Raudolf) has (Raudolf, red nose)~~
4. $\forall x$: ~~Reindeer (x)~~ $\longrightarrow$ weird $(x) \vee$ clown (x)
5. $\neg \exists x$: Reindeer $(x) \wedge$ clown (x)
6. $\forall x$: $\neg$ Loves $($Scrooge$, x) \longrightarrow$ weird (x)

~~3. Reinder (Raudolf) ∧ Reindeer (Red, Nose)~~
3(a) 3(b)

1. $\neg$ child $(x) \vee$ Loves $(x,$ Santa$)$
2. $\neg$ Loves $(x,$ Santa$) \vee$ Reindeer $(y) \wedge$ Loves (x,y)
3. (a) Reindeer (Raudolf)
   (b) Reindeer (Red Nose)
4. $\neg$ Reindeer $(x) \longrightarrow$ weird $(x) \vee$ clown (x)
5. $\forall x$: $\neg$ [Reindeer $(x) \wedge$ clown (x)]
   $\neg$ Reindeer $(x) \vee \neg$ clown (x)
6. $\forall x$: weird $(x) \longrightarrow \neg$ Love $($Scrooge$, x)$
   $\neg$ weird $(x) \vee \neg$ Love $($Scrooge$, x)$

Resolution Proof



Lecture

★ Factors to decide whether to use Backward or forward Reasoning :—

1. Are there more possible start states or goal state? We would like to go from Smaller Set of states to Larger Set of states.

2. In which direction is the branching factor (the average number of nodes that can be reached directly from a Single node) greater? We would like to proceed in the direction with the Lower

Contd...

PAGE NO. :
DATE : / /

* MATCHING :-

1. Indexing :

Do a simple search through all the rules, comparing each one's precondition to the current state and extracting all the ones that match.
But this has two problems
→ In order to solve any

2. Matching with Variables :

The problem of selecting applicable rules is made more difficult when preconditions are not stated as exact descriptions of particular situations but

3. Complex and Approximate Matching :

An even more complex matching process is required if rules should be

PAGE NO. :
DATE : / /

4. Conflict Resolution :

1. Preferences based on rules that match.
2. Preferences based on objects that match.
3. Preferences based on actions that match.

5-03-16

→ UNIT-1

chapter - 1
- AI
- Applications of AI.
- 4 Algos of Tic - Tac - Toe.

chapter - 2.
- Problem characteristics
- Production System
- State Space
- State Space Search
- Additional Problems
- Control Strategies

Imp.

→ UNIT- 2

chapter - 3
- Heuristics
  → Generate and Test
  → Hill climbing Algos
    (i) Simple hill climbing
    (ii) Steepest Ascent HC

Control Strategies
- BFS
- DFS
- Bidirectional Search

(iii) Simulated Annealing

- Best First Search
  (i) OR - Algo
  (ii) A* Algo

- Problem Reduction
  (i) AND - OR Algo
  (ii) AO*

- Constraint Satisfaction
  - Cryparithmetic

→ UNIT - 3

chapter - 4
- Approches to knowledge Representation
- Issues
- Properties of knowledge Representation

chapter - 5
- Diff. b/w Propositional Logic & Predicate Logic
- Resolution (Imp)
  (i) Predicate Conversion
  (ii) CNF
  (iii) Resolution

chapter - 6
- Procedural vs declarative knowledge
- Forward & Backward Reasoning
- Matching
- Conflict Resolution

PAGE NO. :
DATE : / /

5-03-18

Artificial Intelligence
INT-404

* Symbolic Reasoning under uncertainity :—

explore techniques for solving problems with incomplete and uncertain models.

→ The ABC Murder Mystery

→ Non-monotonic Reasoning :—
in which the axioms and/or the rules of inference are extended to make it possible to reason with incomplete information.

→ Statistical Reasoning :—
in which representation is extended to allow some kind of numeric measure of certainity (rather than true or false) to be associated with each statement.

→ Default Reasoning :—
- We use non monotonic reasoning to perform, what is commonly called Default Reasoning.
- Two approches are :
1. Non-monotonic Logic
2. Default Logic

- Two common kinds of Non Monotonic reasoning

PAGE NO. :
DATE : / /

that can be defined by these Logics :-

1. Abduction
2. Inheritance

* Non-Monotonic Logic :-

→ It is one in which the Language of FOPL is augmented with a modal operator M, which can be read as "is Consistent".

→ ~~∀~~ x, y : Related (x, y) ^ M Getalong (x, y) → Will Defend (x, y)

→ for all x and y, if x and y are related and if the fact that x gets along with y is Consistent with everything else that is believed, then Conclude that x will defend y.

* Default Logic :-

$$\frac{A : B}{C}$$

PAGE NO. :
DATE : / /

Tutorial

5-18

Artificial Intelligence
INT - 404

$C_4\ C_3\ C_2\ C_1$

1.
$$\begin{array}{r} B\ A\ S\ E \\ +\ B\ A\ L\ L \\ \hline G\ A\ M\ E\ S \end{array}$$

Assume C_1, C_2, C_3 and C_4 as carry

$\boxed{G = 1}$

$A = [illegible]\ 4$ $\qquad \boxed{A = 4}$

$C_3 + B + B \neq A$ $\qquad \boxed{B = 7}$

$C_3 + 7 + 7 = 8$

$\swarrow \quad \searrow$

$1 \quad 0$

$C_2 + A + A = M$

$C_2 + 4 + 4 = M$

$C_2 + 8 = M$

$\swarrow \quad \searrow$

1 or 0

$\boxed{M = 8}$ or $\boxed{M = 9}$

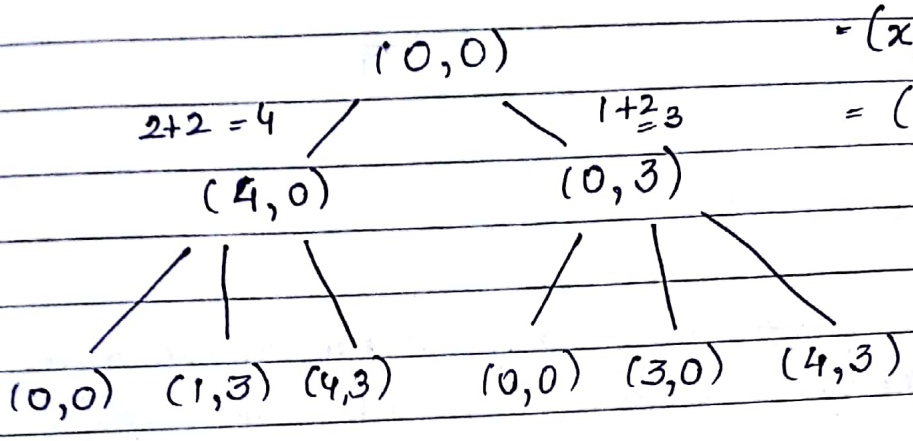
$\boxed{S = 8}$ $\qquad \boxed{L = 5}$ $\qquad \boxed{E = 3}$

PAGE NO. :
DATE : / /

2. Water Jug Problem using A* Algorithm. choose any heuristic, any initial State, Goal State but Consider the cost as 2 at every node.

$h(n) = (x_c - x_g) + (y_c - y_g)$

$= (x_c + y_c - 2)$



3. Convert to PROLOG

$\forall x : pet(x) \wedge Small(x) \longrightarrow apartment(x)$

apartment(X) :- pet(X), Small(X)

$\forall x : Cat(x) \vee dog(x) \longrightarrow pet(x)$

pet(X) :- Cat(X)
pet(X) :- dog(X)

Rules

1. Variables are written with Capital Letter.
2. '$\wedge$' in if is Converted to ','
3. '$\vee$' in if is Converted to 2 parts.
4. '$\vee$', '$\wedge$' in then part → Separate rules.

PAGE NO.:
DATE: / /

$\forall x$: poodle $(x) \rightarrow$ dog $(x) \wedge$ Small (x)

dog (x) :- poodle (x)
Small (x) :- poodle (x)

Lecture

→ Default Logic :—

for performing default based reasoning is Reiter's Default Logic (DL) in which a new class of inference rules is introduced. In this approch,

$$\frac{A : B}{C}$$

→ Abduction :—

Standard logic performs deduction. Give 2 axioms

→ $\forall x : A(x) \rightarrow B(x)$
→ $A(C)$
→ We conclude $B(C)$ using deduction.

→ $\forall x$: Measels $(x) \rightarrow$ Spots (x)

Lecture

9-03-18

Artifical Intelligence

* Inheritance in Default Logic :-

Inheritance says that "An object inherits attributes values from all the classes which it is member unless doing so leads to contradiction, in which case a value from a more restricted class has precedance over a value from a broader class.

* Minimalist Reasoning :-

Two Kinds of minimalist reasoning :

1. closed World Assumption (CWA)
2. Circumscription.

1. closed World Assumption (CWA):
This type of assumption is useful in application where most of the facts are true and it is, therefore, reasonable to assume that if a proposition cannot be proven, it is false.

2. Circumscription :

Two advantages over CWA :

→ operates on whole formulas, not individual predicates.

PAGE NO.:
DATE: / /

→ Allows some predicates to be marked as closed and others as open.

• It is not fesiable to write all the exceptions.

→ AB Predicated

Adultmale(x) : ¬ Baseball P(x) ^ ¬ Midget(x) ^ ¬ Jockey(x) ^ height (x, 5-10)

height (x, 5-10)

∀x : Adultmale (x) ∧ ¬ AB (x, aspect) → height (x, 5-10)

∀x : Baseball player (x) → AB (x, aspect 1)
∀x : Midget (x) → AB (x, aspect 1)
∀x : Jockey (x) → AB (x, aspect 1)

★ Augmented Problem Solver :—
(Backward Rules Using UNLESS)

→ Implementation Issues :—

1. How to find out only those rules from non monotonic reasoning which are actually useful.
2. How to update our knowledge base while solving our problem.

3. There can be possibility where we can draw difference inference from same rule.

4. This is not computationally effective / or is difficult.

We will be solving the issues in 2 steps :-

1. Actual Problem Solver.
2. TMS → Truth Maintainence System.

→ require to do away with (2) issue
→ will help us to eliminate the 2nd issue.

→ The computational ~~System~~ Solutions to these problems are separated into two parts :

1. Problem Solver that uses whatever mechanism it happens to draw conclusion as necessary.

2. TMS whose job is to do the book keeping required to provide a solution to our 2nd problem.

Two mechanism of Problem Solving

1. Backward Reasoning
2. Forward Reasoning

PAGE NO. :
DATE : / /

1. Problem Solver

Suspect (x) ← Beneficiary (x)
UNLESS Alibi (x)

Alibi (x) ← Somewhere Else (x)

Somewhere Else (x) ← Registered Hotel (x, y) and
Far Away (y)
UNLESS Forged Register (y)

Alibi (x) ← Defends (x, y)
UNLESS Lies (y)

Somewhere Else (x) ← Picture Of (x, y) and
Far away (y)

Contradiction ( ) ← TRUE
UNLESS ∃x : Suspect (x)

DATE : / /

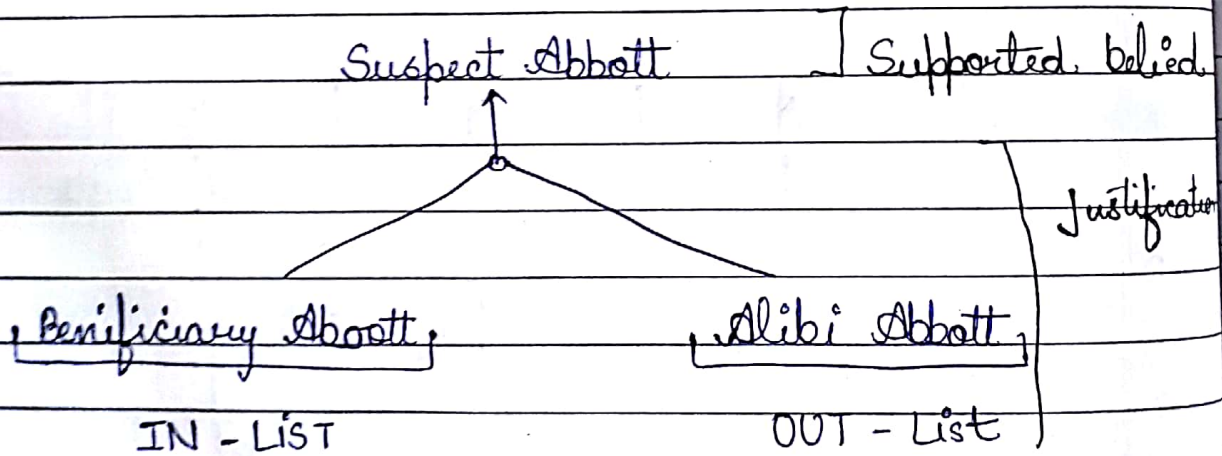
Artifical Intelligence
INT - 404

* **ABC Murder Mystery :—**

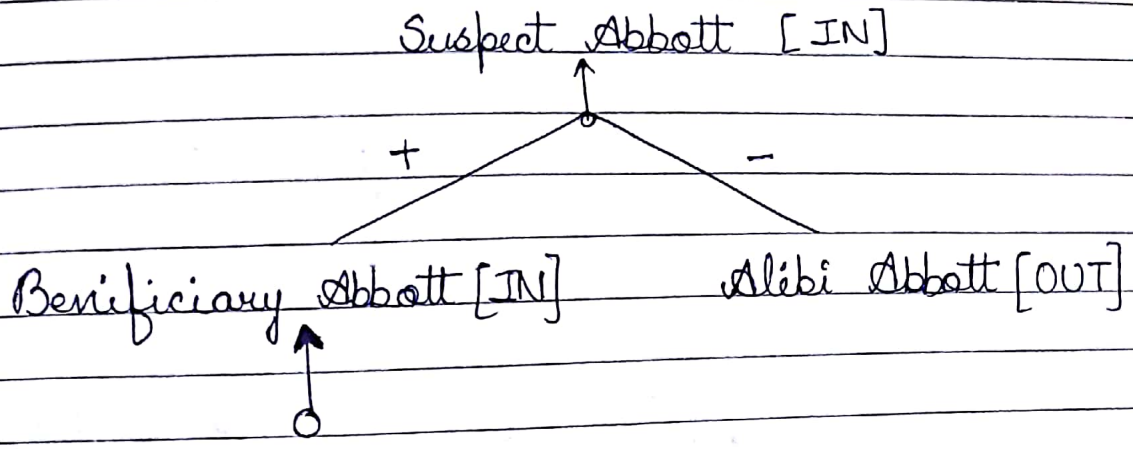
Various techniques for TMS

1. JTMS (Justification Based TMS)
2. ATMS (Assumption Based TMS)
3.

1. JTMS (Justification based TMS)

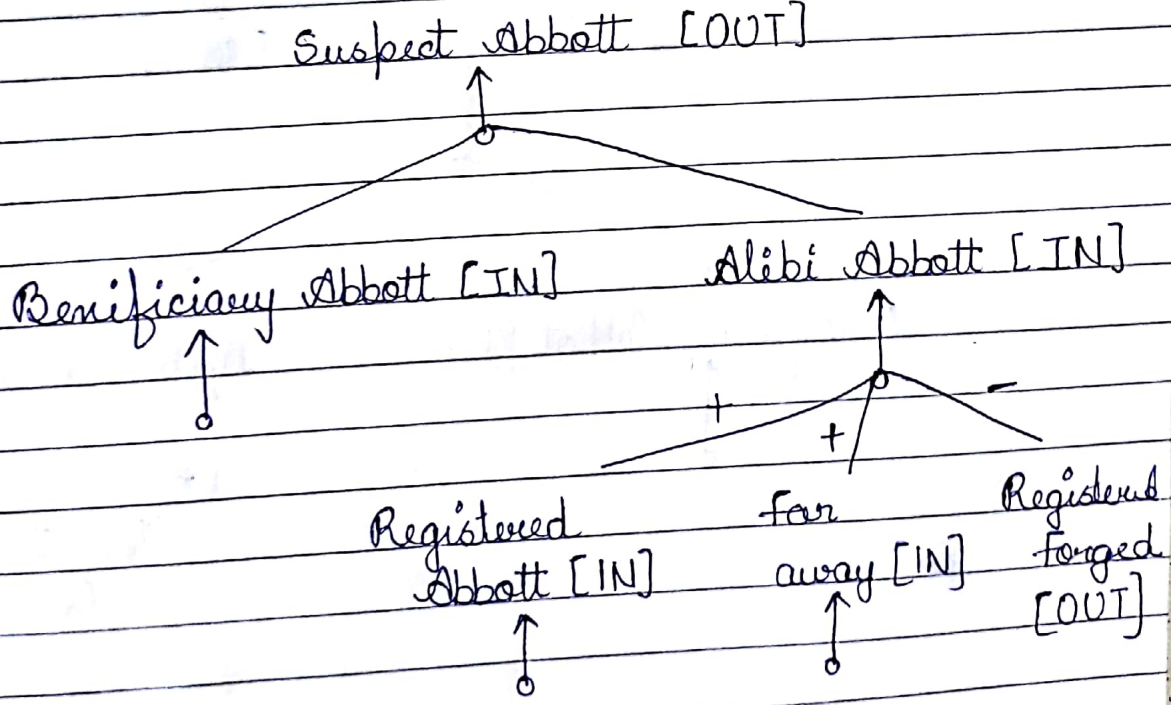


- A justification is Valid if every assertion in the IN-list is believed and none of those in the OUT-list is.
- A justification is Non-monotonic if its OUT-list is not empty, or, recursive, if any assertion in its IN-list has a Non-monotonic justification.



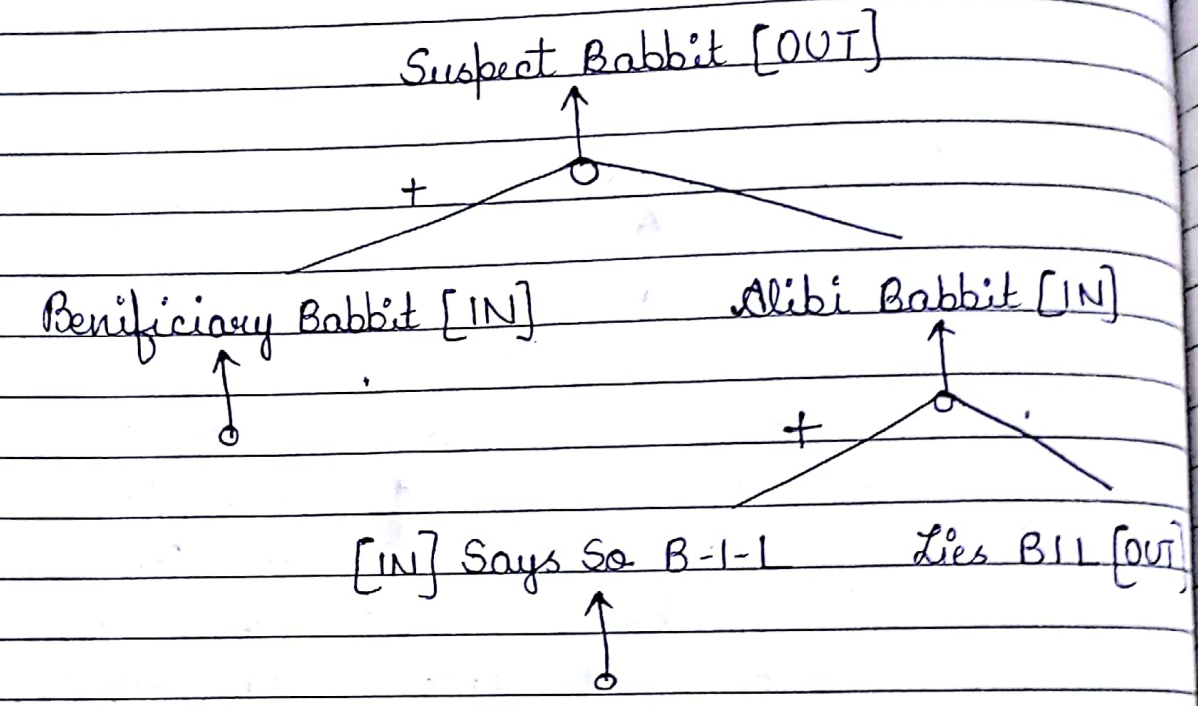
We are told that Abbott is a Benificiary, We have no further justification for this fact So we must simply except it. For such facts we give a premise justification i.e a justification with empty IN and OUT List.

changed Labeling

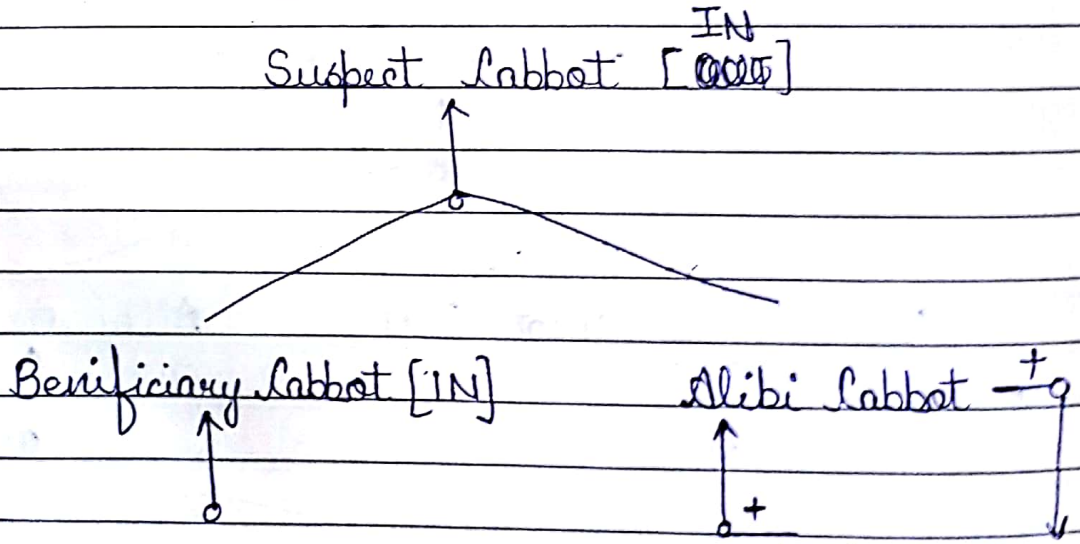


PAGE NO.:
DATE: / /

Babbit's Justification

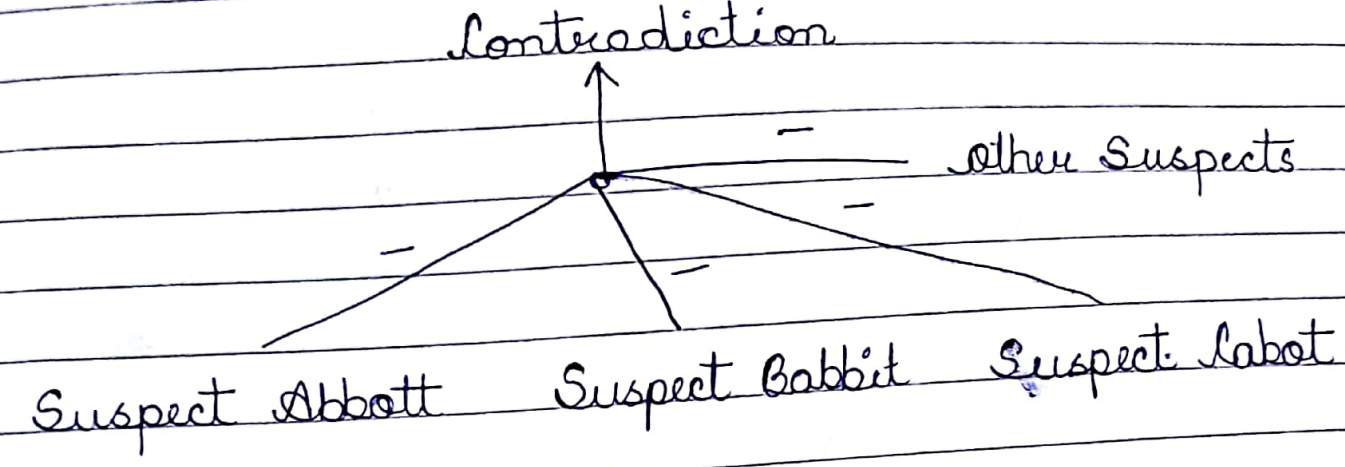


Labbot's Justification

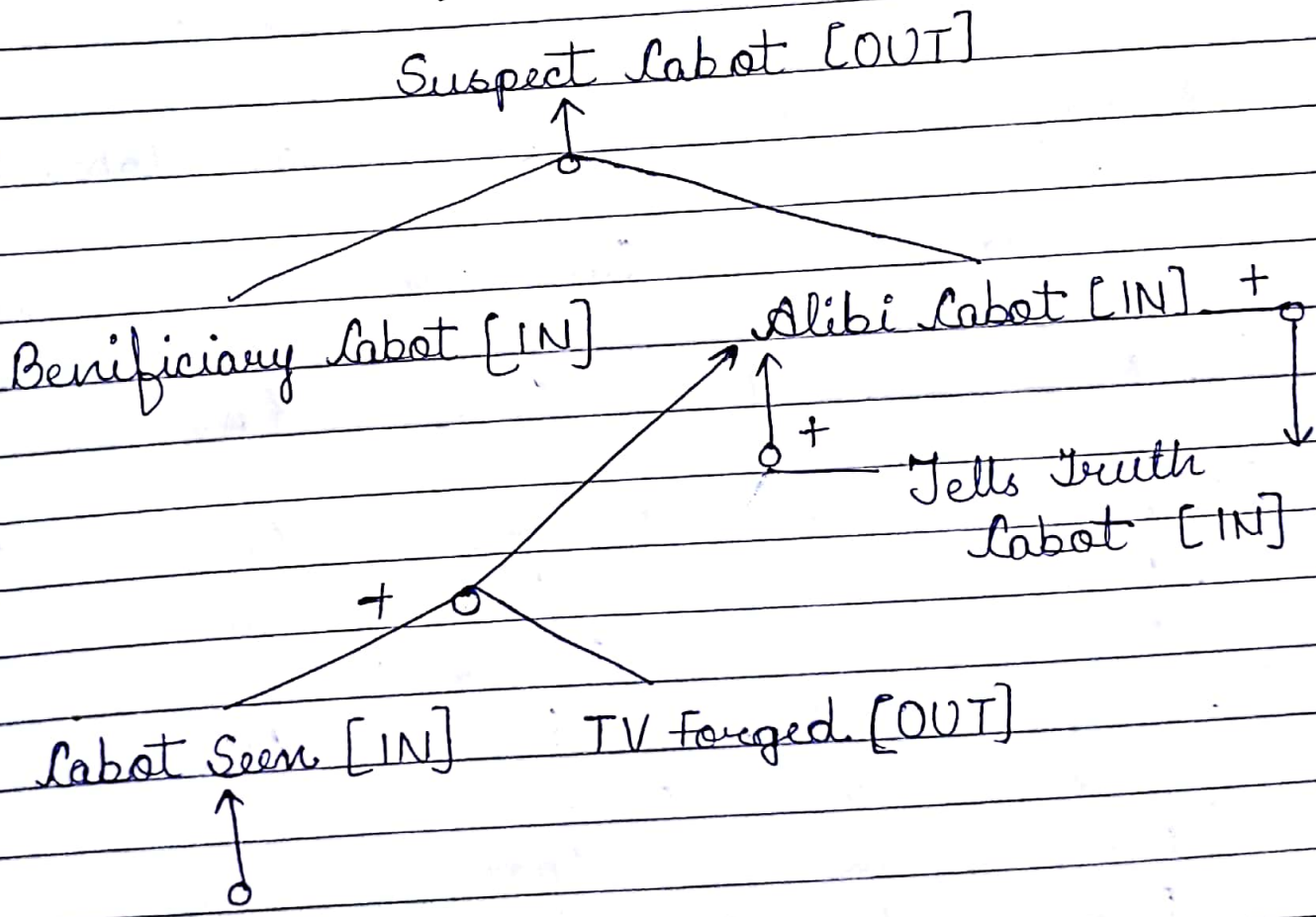


PAGE NO. :
DATE : / /

A Contradiction



A Second Justification



Lecture

PAGE NO. :
DATE : / /

* ATMS (Assumption based TMS) :-

A1 : Hotel register was forged
A2 Hotel register was not forged
A3 Babbitt's brother-in-Law Lied.
A4 Babbitt's brother-in-Law did not Lied.
A5 Labot Lied.
A6 Labot did not Lie.
A7 Abbott, Babbitt, Labot are only Suspects.
A8 Abbott, Babbitt, Labot are not only Suspects.

* Nodes and their Justifications and Labels :-

	Nodes	Justifications		Labels
[1]	Register was not forged.	{A2}	{A2}	{A2}
[2]	Abbott at Hotel	[1] → [2]	{A2}	{A2}
[3]	BIL didn't Lie	{4}	{A4}	{A4}
[4]	Babbid at BIL	[3] → [4]	{A4}	{A4}
[5]	Labbot didn't Lie	{6}	{A6}	{A6}
[6]	Labbot at Show	[5] → [6]	{A6}	{A6}
[7]	A, B, C only Susp.	{A7}	{A7}	{A7}
[8]	Prime Susp A	[7] ∧ [13] ∧ [14] → [8]	{A7, A4, A6}	{A7, A4, A6}
[9]	Prime Sp. B	[7] ∧ [12] ∧ [14] → [9]	{A7, A2, A6}	{A7, A2, A6}
[10]	Prime Susp. C	[7] ∧ [12] ∧ [13] → [10]	{A7, A2, A4}	{A7, A2, A4}
[11]	A, B, C not Sup.	{A8}	{A8}	{A8}

PAGE NO. :
DATE : / /

[12] Not prime Sup. A

[2] → [12] {A2} {A2}{A8}
[11] → [12] {A8}
[9] → [12] {A7, A2, A6}
[10] → [12] {A7, A2, A4}

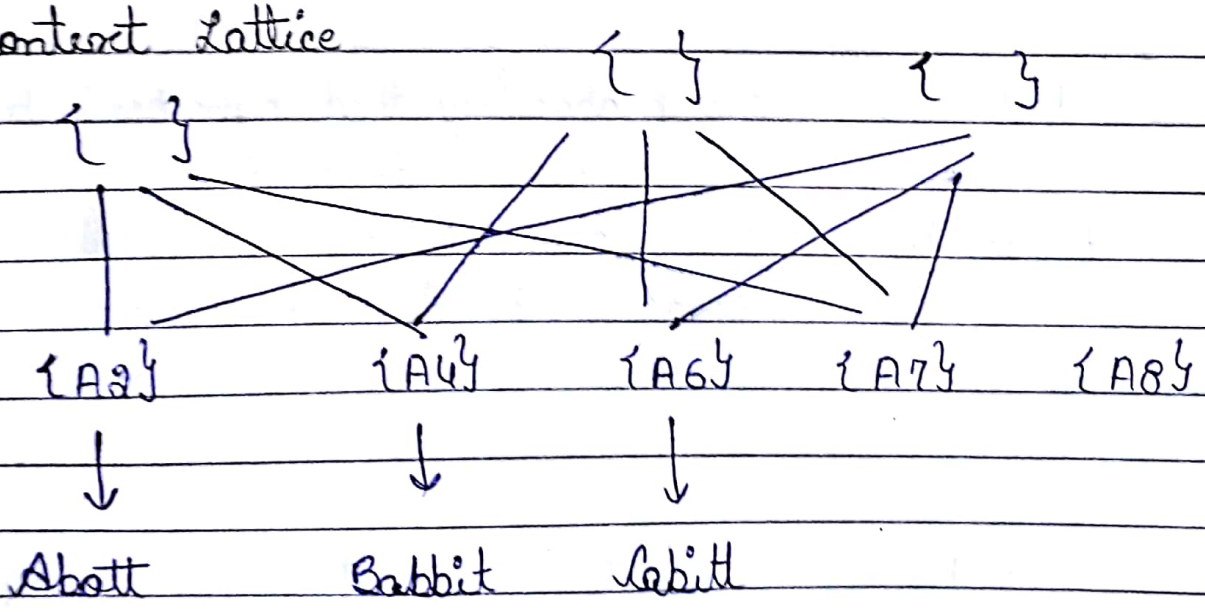
[13] Not prime Sup. B

[4] → [13] {A4} {A4}{A8}
[11] → [13] {A8}
[8] → [13] {A7, A4, A6}
[10] → [13] {A7, A4, A2}

[14] Not prime Sup. C

[6] → [14] {A6} {A6}{A8}
[11] → [14] {A8}
[8] → [14] {A7, A4, A6}
[9] → [14] {A7, A2, A6}.

Context Lattice



PAGE NO.:
DATE: / /

26-03-18

chapter -8

Artifical Intelligence
INT-404

* **Statistical Reasoning :-**

→ **Bayes Theorem :-**

Probability =

$$P(x) = \frac{\text{No. of favourable outcome}}{\text{Total outcome}}$$

The notion of conditional probability : $P(H/E)$

Let

$P(H_i/E)$ = the probability that hypothesis H_i is true given evidence E.

$P(E/H_i)$ = the probability that we will observe evidence E that the hypothesis H_i is true.

$$P(H_i/E) = \frac{P(E/H_i) \cdot P(H_i)}{\sum P(H_i) \cdot P(E/H_i)}$$

Car, Bus, Train

PAGE NO :
DATE : / /

Ex: $P(C) = \frac{1}{3}$ $\quad P(L/C) = 0.5$

$P(B) = \frac{1}{3}$ $\quad P(L/B) = 0.2$

$P(T) = \frac{1}{3}$ $\quad P(L/T) = 0.01$

$$P(C/L) = \frac{P(C) \cdot P(L/C)}{P(C) \cdot P(L/C) + P(B) P(L/B) + P(T) P(L/T)}$$

$$= \frac{(1/3).(0.5)}{\frac{1}{3}.(0.5) + \frac{1}{3}.(0.2) + \frac{1}{3} \times (0.01)}$$

$$= \frac{1/3 \times 5/10}{\frac{1}{3} \times \frac{5}{10} + \frac{1}{3} \times \frac{2}{10} + \frac{1}{3} \times \frac{1}{100}}$$

(b) $P(B) = 0$ $\quad P(L/C) = 0.5$

$P(T) = 0.9$ $\quad P(L/B) = 0.2$

$P(C) = \frac{10}{100} = 0.1$ $\quad P(L/T) = 0.01$

→ Adding Certainty factors to Rules :-

Example of Mycin Rule :
→ Expert System

Ex :

$A_1 \rightarrow$ it will rain
$A_2 \rightarrow$ it will not rain

$$P(A_1) = \frac{5}{365} \qquad P(A_2) = \frac{360}{365}$$

$B \rightarrow$ Weatherman predicted the rain
$P(B/A_1) = 90\%$.
$P(B/A_2) = 10\%$.

$$P(A_1/B) = \frac{P(A_1) \cdot P(B/A_1)}{P(A_1) \cdot P(B/A_1) + P(A_2) \cdot P(B/A_2)}$$

$$= \frac{\left(\frac{5}{365} \times \frac{90}{100}\right)}{\left(\frac{5}{365} \times \frac{90}{100}\right) + \left(\frac{360}{365} \times \frac{10}{100}\right)}$$

$$= 0.111 \quad \text{Ans.}$$

PAGE NO. :
DATE : / /

→ Measures of Belief :—

Certainity factor [-1 to +1]

1. $CF(h,e) = MB(h,e) - MD(h,e)$

2. $MB(h, e_1 \wedge e_2) = MB[h,e_1] + MB[h,e_2] \cdot [1 - MB[h,e_1]]$

$MD(h, e_1 \wedge e_2) = MD[h,e_1] + MD[h,e_2] \cdot [1 - MD[h,e_1]]$

$CF(h, e_1 \wedge e_2) = MB[h, e_1 \wedge e_2] - MD[h, e_1 \wedge e_2]$

3. $MB[h_1 \wedge h_2, e] = \min [MB(h_1,e), MB[h_2,e]]$

$MD[h_1 \wedge h_2, e] = \min [MD(h_1,e), MD(h_2,e)]$

4. $MB[h_1 \vee h_2, e] = \max [MB(h_1,e), MB(h_2,e)]$

$MD[h_1 \vee h_2, e] = \max [MD(h_1,e), MD(h_2,e)]$

h → hypothesis

e → evidence

MB → Measure of Belief
MD → Measure of Disbelief

27-03-18

Artifical Intelligence
INT-404

Ques.1 $MB(h, e_1) = 0.3$
$MD(h, e_1) = 0$
$CF(h, e_1) = 0.3$
$MB(h, e_2) = 0.2$

$$MB(h, e_1 \wedge e_2) = MB(h, e_1) + MB(h, e_2)[1 - MB(h, e_1)]$$

$$= 0.3 + 0.2(1 - 0.3)$$
$$= 0.3 + 0.2 \times 0.7$$
$$= 0.3 + 0.14$$
$$= 0.44$$

$$MD(h, e_1 \wedge e_2) = MD(h, e_1) + MD(h, e_2)[1 - MD(h, e_1)]$$

$$= 0 + 0[1 - 0]$$
$$= 0$$

$$CF(h, e_1 \wedge e_2) = MB(h, e_1 \wedge e_2) - MD(h, e_1 \wedge e_2)$$

$$= 0.44 - 0$$
$$= 0.44 \quad \text{Ans.}$$

Ques.2. $MB(h, e_1) = 0.5$
$MB(h, e_2) = 0.2$
$MB(h, e_2) = 0.6$

Find $CF(h, e_1 \wedge e_2 \wedge e_3)$

PAGE NO. :
DATE : / /

$$MB(h, e_1 \wedge e_2) = MB(h, e_1) + MB(h, e_2)[1 - MB(h, e_1)]$$

$$= 0.5 + 0.2[1 - 0.5]$$

$$= 0.6$$

$$MB(h, (e_1 \wedge e_2) \wedge e_3) = MB(h, e_1 \wedge e_2) + MB(h, e_3)[1 - MB(h, e_1 \wedge e_2)]$$

$$= 0.6 + 0.6(1 - 0.6)$$

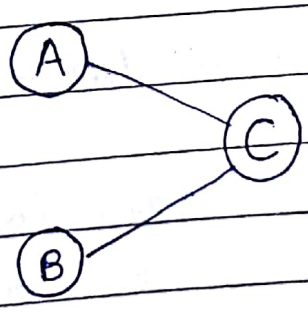
$$= 0.6 + 0.6 \times 0.4$$

$$= 0.6 + 0.24$$

$$= 0.84$$

$$Cf(h, e_1 \wedge e_2 \wedge e_3) = MB(h, e_1 \wedge e_2 \wedge e_3) - MD(h, e_1 \wedge e_2 \wedge e_3)$$

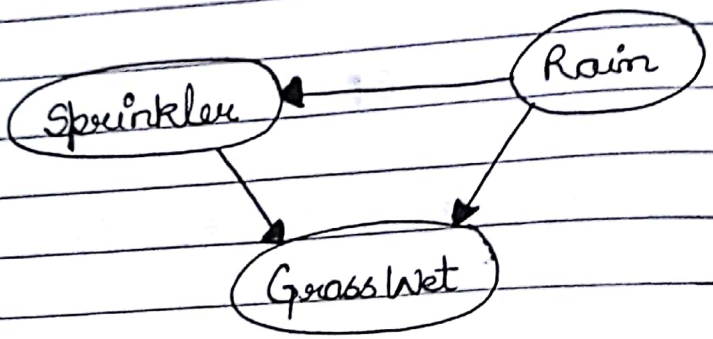
★ Combining two Pieces of Evidence :-



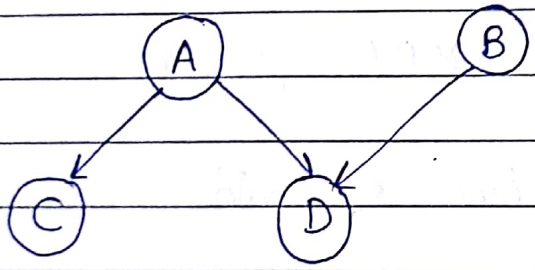
PAGE NO.:
DATE: / /

★ Bayesian Network :—

ex:



The bayesian network is an acyclic directed graph where nodes of the graph represent evidence or hypothesis and arc connecting two nodes represents dependence between them.



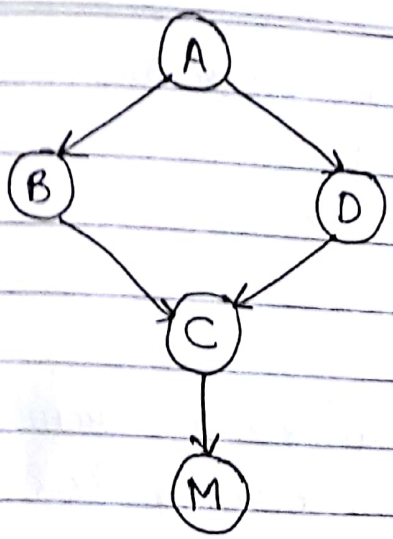
- if nodes having no parent called unconditional node like A, B.
- If nodes having parent called Conditional
- Node with no children called hypothesis and with children

- Joint Probability for above example:

$$P(G,S,R) = P(G/S,R) * P(R) * P(S/R)$$

PAGE NO. :
DATE : / /

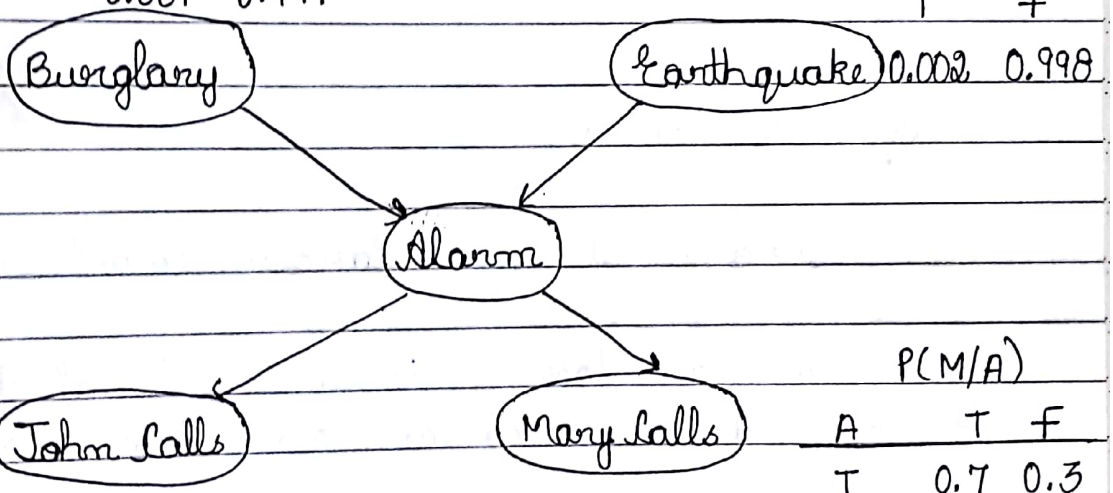
Ex:



$$P(A,B,C,D,M) = P(A) * P(B/A) * P(D/A) * P(C/B,D)$$

$$* P(M/C)$$

Ex:



A	T	F
T	0.90	0.1
F	0.05	0.95

P(M/A)

A	T	F
T	0.7	0.3
F	0.01	0.99

$$P(J,M,A,B,E) = P(J/A) * P(M/A) * P(A/B,E) * P(B) * P(E)$$

$$P(A=T / J=T, M=F, B=T) = \frac{\sum_{E=T,F} P(A=T, J=T, M=F, B=T)}{\sum_{(A,E)=T,F} P(J=T, M=F, B=T)}$$

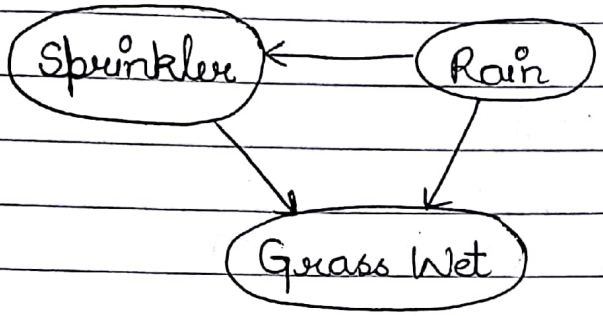
29-03-18

Tutorial

PAGE NO. :
DATE : / /

Artificial Intelligence
INT-404

Ques.



	Sprinkler	
Rain	T	f
T	0.4	0.6
f	0.01	0.99

Rain	
T	f
0.2	0.8

Sprinkler	Rain	Grass Wet	
		T	f
f	f	0.0	1.0
F	T	0.8	0.2
T	f	0.9	0.1
T	T	0.99	0.01

I. Joint Probability

$P(G,S,R) = P(R) * P(S/R) * P(G/S,R)$

Case 1 : $P(G=T, S=T, R=f)$

$P(G=T, S=T, R=f) = P(R=f) * P(S=T/R=f) *$

$P(G=T/S=T, R=f)$

PAGE NO. :
DATE : / /

$= 0.8 * 0.4 * 0.9$

$= 0.288$ Ans.

Case 2: $P(G = T)$

↓

$P(G = T, S = T, R = T)$

$P(G = T, S = T, R = F)$

$P(G = T, S = F, R = T)$

$P(G = T, S = F, R = F)$

$P(G = T, S = T, R = T) = P(R = T) * P(S = T / R = T) * P(G = T / S = T, R = T)$

$= 0.2 * 0.01 * 0.99$

$= 0.00198$

$P(G = T, S = F, R = T) = P(R = T) * P(S = F / R = T) * P(G = T / S = F, R = T)$

$= 0.2 * 0.99 * 0.8$

$= 0.1584$

$P(G = T, S = F, R = F) = P(R = F) * P(S = F / R = F) * P(G = T / S = F, R = F)$

$= 0.8 * 0.6 * 0.0$

$= 0$

$P(G = T) = 0.00198 + 0.288 + 0.1584 + 0$

$= 0.44838$ Ans.

II Conditional Probability

$$P(R/G) = \frac{\sum_{S=T,F} P(R,G)}{\sum_{S,R=T,F} P(G)}$$

$$= \frac{P(G=T,R=T,S=T) + P(G=T,R=T,S=F)}{P(G=T,R=T,S=T) + P(G=T,R=T,S=F) + (P(G=T,R=F,S=T) + P(G=T,R=F,S=F)}$$

$$= \frac{0.00198 + 0.158}{0.00198 + 0.158 + 0.288 + 0}$$

$$= 0.357 \quad \text{Ans.}$$

Ques 1. Consider the following set :

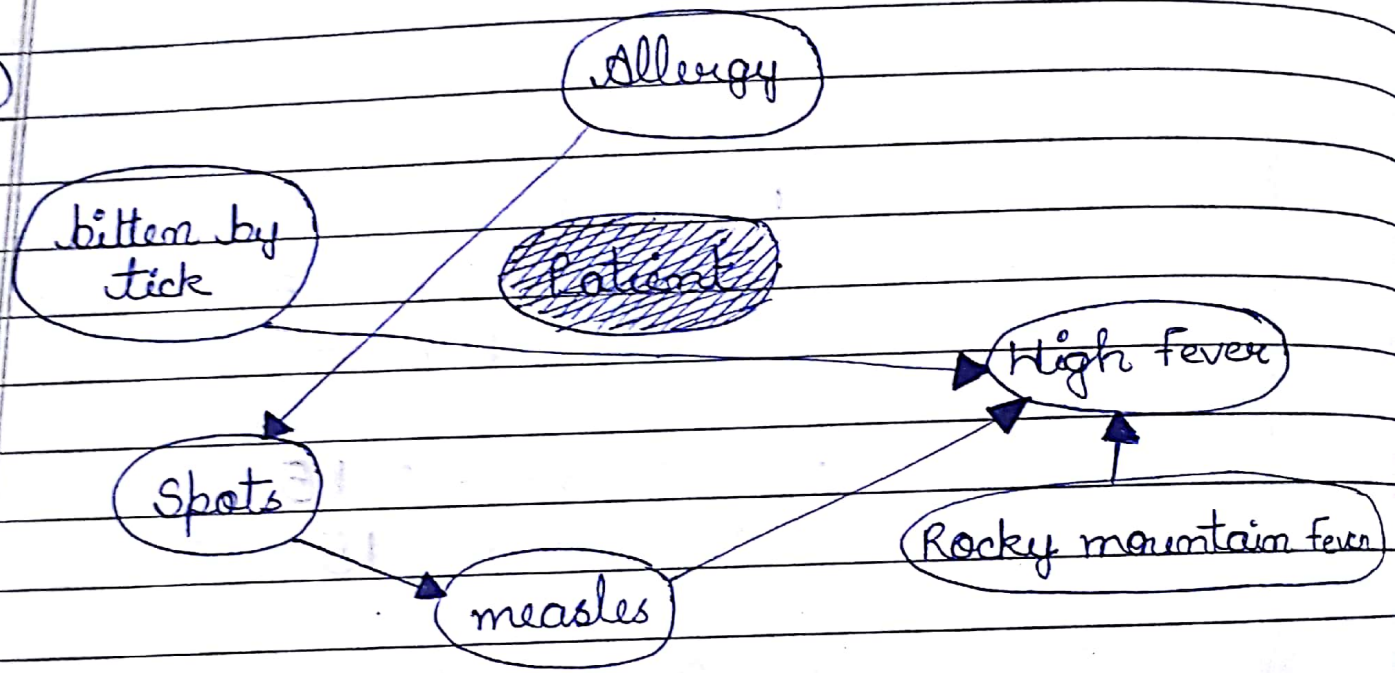
1. Patients has Spots.
2. Patients has measles.
3. Patient has high Feaver.
4. Patient has rocky mountain Spoted feven.
5. Patient has previously being detected against measles.
6. Patient was recently bitten by a tick.
7. Patient has an Allergy.

PAGE NO.:
DATE: / /

Ques a) Create a network that defines the causal connections among these nodes.

Ques b) Make it a bayesian network by constructing the necessary conditional probability matrix.

(a)



(b)

PAGE NO. :
DATE : / /

Ques 2. Suppose we have following rule R1 if (P1 and P2 and P3) or (P4 and not P5) then C1 (0.7) C2 (0.5)

CF(P1) = 0.8
CF(P2) = 0.7
(P3) = 0.6
(P4) = 0.9
(P5) = -0.5

What are the certainity factors associated with conditions C1 and C2 after using rule R1.

(P1 and P2 and P3)

Cf = min (P1 and P2 and P3)
CF = min (0.8, 0.7, 0.6)
Cf = 0.6

(P4 and P5)

Cf = min (0.9, ¬P5)
= min (0.9, 0.5)
= 0.5

Cf = 0.6 or 0.5
Cf = max (0.6, 0.5)
Cf(R1) = 0.6

$CF_{C1}\ R1 = 0.6 \times 0.7 = 0.42$

$CF_{C2}\ R1 = 0.6 \times 0.5 = 0.3$

Lecture

★ Dempster-Shafer Theory :—

Dempster Shafer Theory Considers the Set of propositions and assigns to each of them an interval in which degree of belief must lie.

[Belief, Plausibility]

Belief denoted as Bel measures the Strength of the evidence in favour of a Set of propositions. It ranges from 0 (no evidence) to 1 (definite Certainty)

Plausibility (Pl) is $PL(s) = 1 - Bel(-s)$

It also ranges from 0 to 1 and measures the extent to which evidence in favour of (not)s, Leaves room for belief in s.

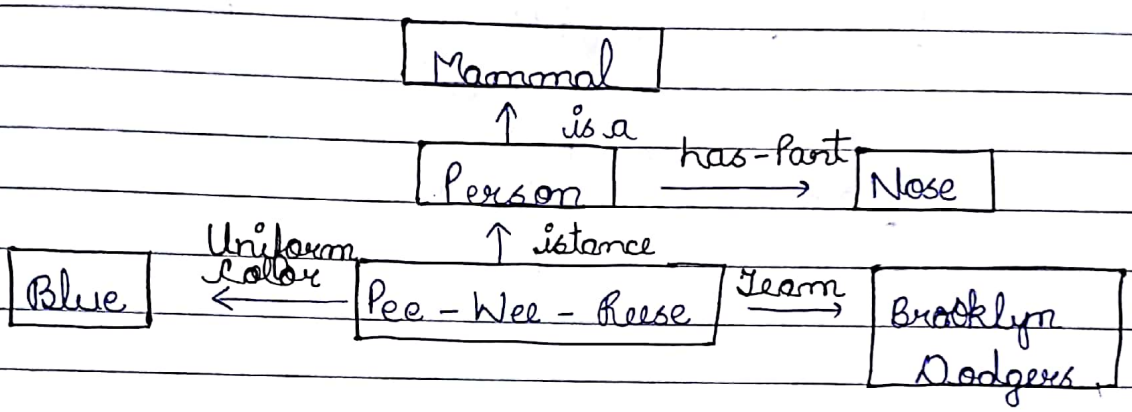
PAGE NO. :
DATE : / /

8-04-18

Artifical Intelligence

UNIT- 5

★ **SEMANTIC NETWORKS :—**



→ **Representing Non Binary Predicates :—**

- Unary Predicates can be rewritten as binary one.

  man (Marcus)

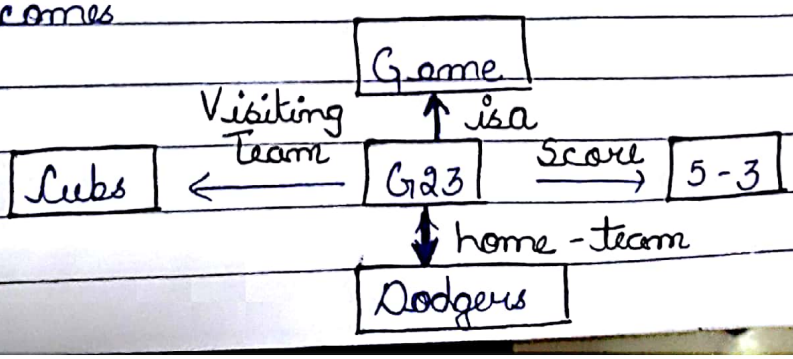
  Could be rewritten as

  instance (Marcus, Man)

- N-Place Predicates

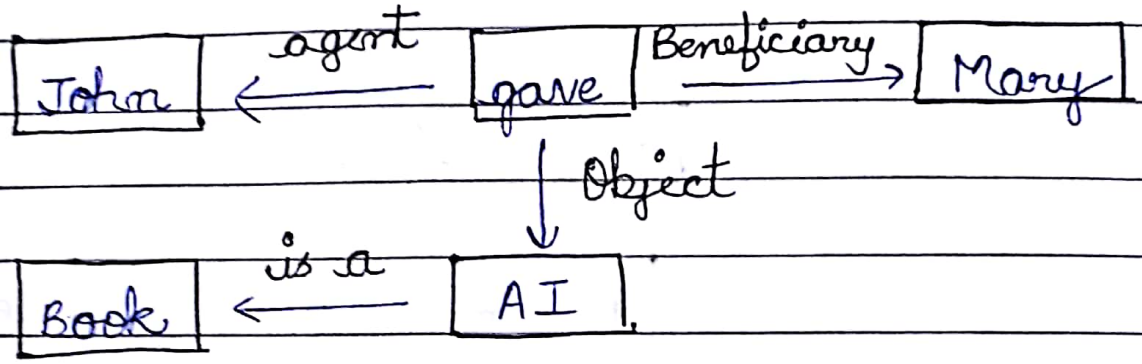
  Score (Cubs, Dodgers, 5-3)

  becomes



→ A Semantic Net representing a Sentence :-

- John gave the book to Mary.



Lecture

PAGE NO. :
DATE : / /

5-04-18

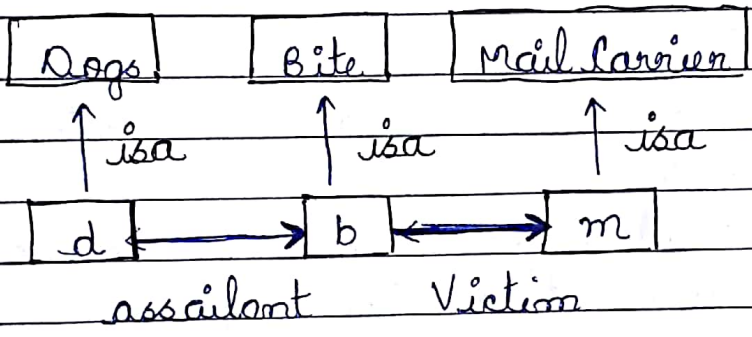
INT-404

Artifical Intelligence

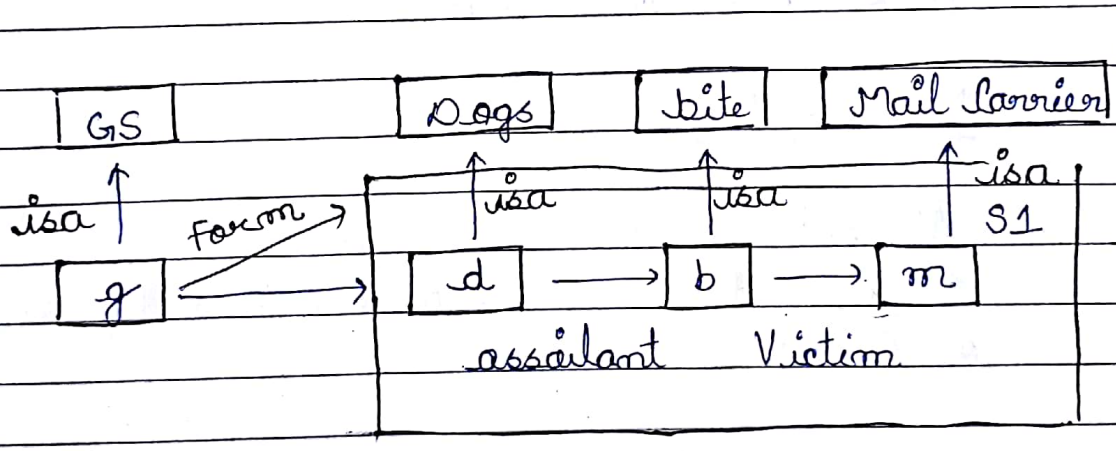
* Partitioned Semantic Nets :—

(a) The dog bit the mail carrier.
b) Every dog has bitten a mail carrier.
c) Every dog in town has bitten the constables.
d) Every dog has bitten every mail carrier.

a)

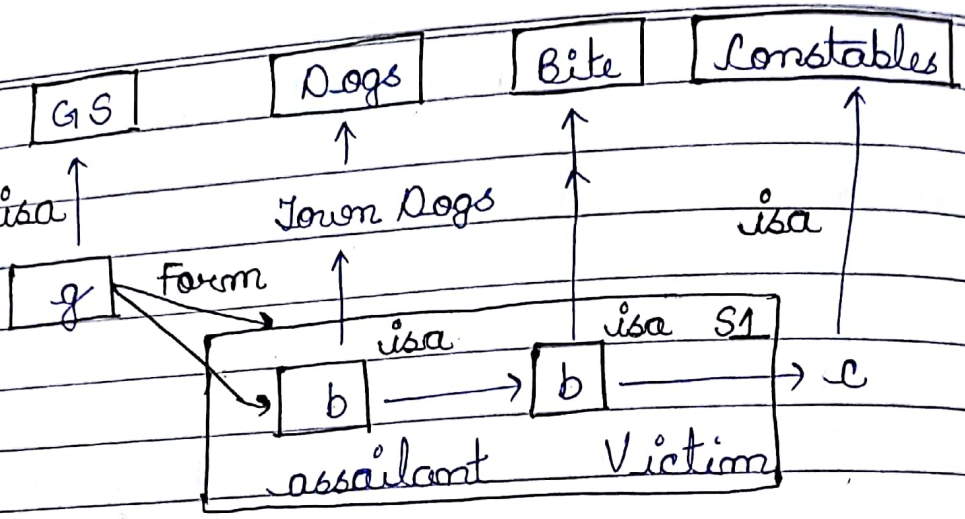


(b)



GA → General Statement.
SA → Space corresponding to Whole event.
S1 → Space corresponding to own event.

(c)



* FRAMES :—

→ represents an entity as a set of slots and associated values.

→ The concept of a frame is defined by a collection of slots. Each slot describe a particular attribute or operation of a frame.

Frame System

→ Attributes about the set itself.

* → Attributes that are to be inherited by each element of the set.

Regular class :—

Those classes whose elements are individual entities.

Meta class :—

Those classes whose elements are themselves classes.

PAGE NO. :
DATE : / /

Regular class : {E_1, E_2, E_3} → instance

Meta class : {C_1, C_2, C_3} → is-a

Tangled Hierarchy :—

To Support flexible representations of knowledge about the world. It is necessary to allow the hierarchy to be an arbitary directed acyclic graph. Hierarchies that are not trees are called Tangled Hierarchies

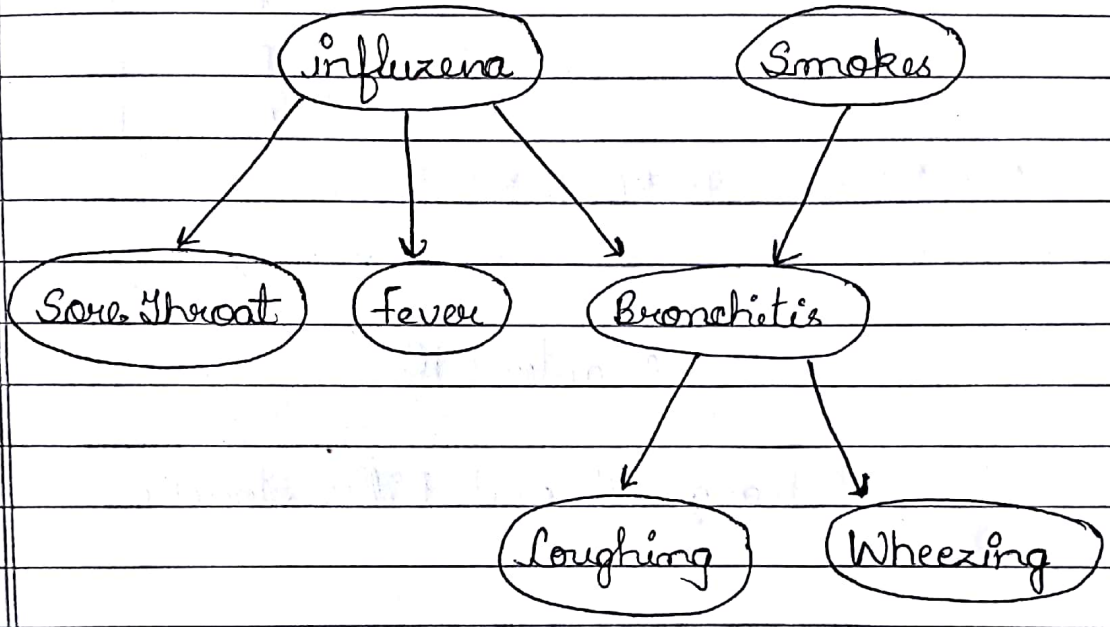
Tutorial

PAGE NO.:
DATE: / /

12-04-18

Artifical Intelligance
INT-404

Ques: Find the Probability of Wheezing, given that influenza is true, fever is true, bronchitis is true, Sore throat is false.



$$P(W, C, B, S, St, f, I) = P(C/B) * P(W/B) * P(B/I, S) * P(f/I) * P(St/I) * P(S) * P(I)$$

$$P(W = T / I = T, f = T, B = T, St = f) = \sum_{C,S \in T,f} P(W=T, I=T, f=T, B=T, St=f)$$

$$\sum_{(C,S,W) \in (T,f)} P(I=T, f=T, B=T, St=f)$$

$P(St/I) = 0.3$, $P(\neg St/I) = (1-0.3) = 0.7$

$P(I=T, F=T, B=T, St=F, C=T, S=T, W=T)$

$= P(C=T/B=T) * P(W=T/B=T) * P(B=T/I=T, S=T) * P(F=T/I=T) * P(St=F/I=T) * P(S=T) * P(I=T)$

C	S	W
T	T	T
T	T	f
T	f	T
T	f	f
f	T	T
f	T	f
f	f	T
F	f	f

$= 0.8 * 0.6 * 0.99 * 0.9 * 0.7$

Chapter = 10

Strong Slot and Filler Structure

Strong Slot & Filler Structure

- Conceptual Dependency
- Script

→ The Probability of getting the answers from Strong Slot (CD and Script) are more as Compared to Weak Slot & Filler Structure.

PAGE NO. :
DATE : / /

★ Conceptual Dependency :—

is a theory to represent a kind of knowledge about event that is usually contained in Natural Language Sentance.

A Simple Conceptual Dependency Representation

"I gave the man a book"

CD Primitive Actions

1. ATRANS : Transfer of an abstract relationship
   ex : give
2. PTRANS : Transfer of the physical Location of Object
   ex : go
3. PROPEL : Application of physical force to an Object
   ex : Push
4. MOVE : Movement of a body part by its owner.
   ex : Kick
5. GRASP : Grasping of an object by an Actor
   ex : clutch.
6. INGEST : Ingestion of an object by an Animal
   ex : Eat
7. EXPEL : Expulsion of Something from body
   ex : Cry

8. MTRANS : Transfer of mental information.
Ex : Tell

9. MBUILD : Building new info. out of old.
Ex : Decide

10. SPEAK : Production of Sounds
Ex : Say

11. ATTEND : Focusing of a Sense toward a

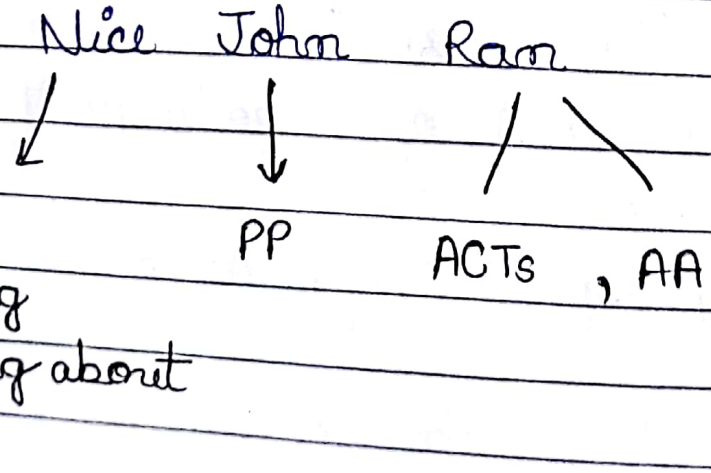
CD Primitive Conceptual Categories

ACTs Actions

PPs Objects (picture producers)

AAs Modifiers of actions (action aiders)

PAs Modifiers of PPs (picture aiders)



PAGE NO. :
DATE : / /

The Dependencies of CD

1. $\Longleftrightarrow$ used to represent the relation b/w Object & Action.

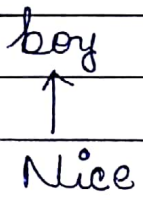
John ran — John $\overset{P}{\Longleftrightarrow}$ PTRANS

2. $\Longleftrightarrow$ used to represent is a relation.
i.e b/w class - class
class - Instance

John is a doctor — John $\Longleftrightarrow$ doctor

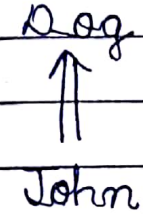
3. PP ↑ PP used to represent relation b/w PA and PP.

Nice boy

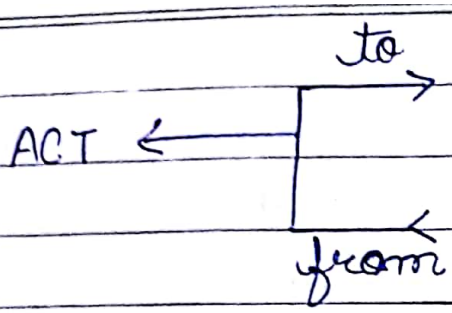


4. PP ⇑ PP used to represent some possesion

John's Dog

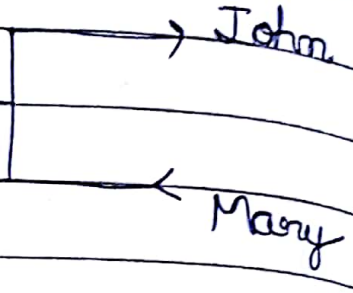


5.



represent give & take

John took the book from Mary



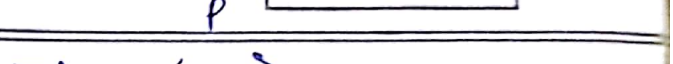
CD Dependencies Tense

p	Past
f	Future
t	Transition
ts	Start Transition
tf	Finished transition
K	Continuing
?	Interrogative
/	Negative
nil	Present
delta	Timeless
c	Conditional

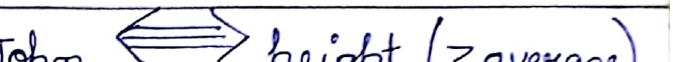
The Dependencies of CD :—

PAGE NO. :
DATE : / /

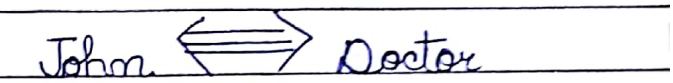
1. John ran



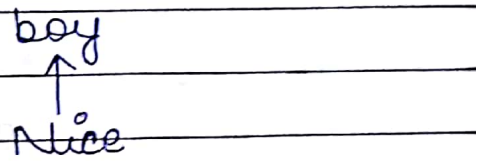
2. John is Tall



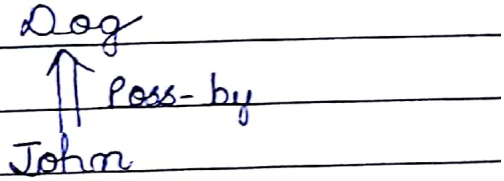
3. John is a doctor



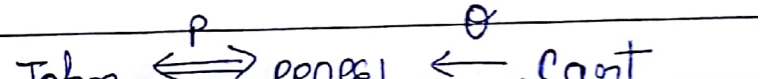
4. A nice boy



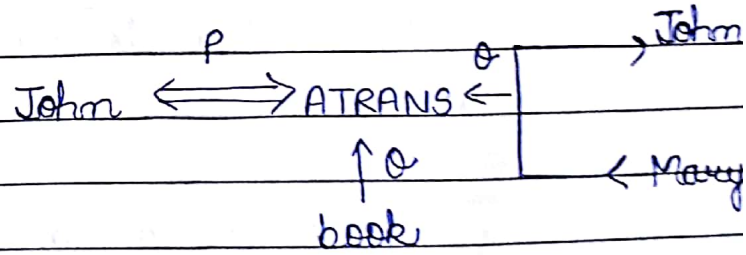
5. John's Dog



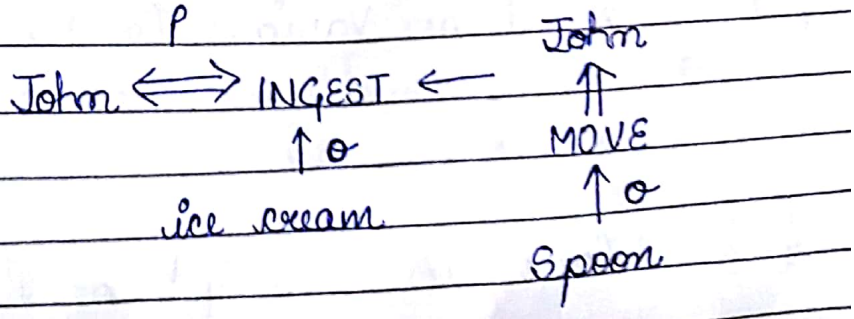
6. John pushed the cart.



7. John took the book from Mary



8. John ate ice cream with a Spoon



PAGE NO. :
DATE : / /

★ SCRIPT :—

Entry Conditions : Conditions that must, in general, be Satisfied before the events described in the Script can occur.

Result Conditions : that will, in general, be true after the events described in the Script have occured.

Props : Slots representing Objects that are involved in the events described in the Script.

Roles : Slots representing people who are involved in the events described in the Script.

Track : The Specific variation on a more general pattern that is represented by this particular Script.

Scenes : The actual Sequences of events that occur. The events are represented in Conceptual dependency formalism.

Ques. What are Various Components of Script?

→ Entry Conditions, Result Conditions, Props, Track, Scenes.

Ques. Construct a Script on Restaurants, Theatre.

PAGE NO. :
DATE : / /

CD Primitive Actions

1. ATRANS → Give, Take, Purchase
2. PTRANS → go, walk, fly
3. PROPEL → Push, Pull, throw
4. MOVE → Kick, hit, throw, Punch
5. GRASP → catch, clutch
6. INGEST → eat, drink, Smoke, breathe
7. MTRANS → Tell, Speak, Sing, read, Remember
8. MBUILD → decide, describe, imagine, Consider, Answer.
9. EXPEL → Weep, Cry, Sweat
10. SPEAK → Say, Tell, Sing
11. ATTEND → Listen, Watch, See, Look

17-04-18

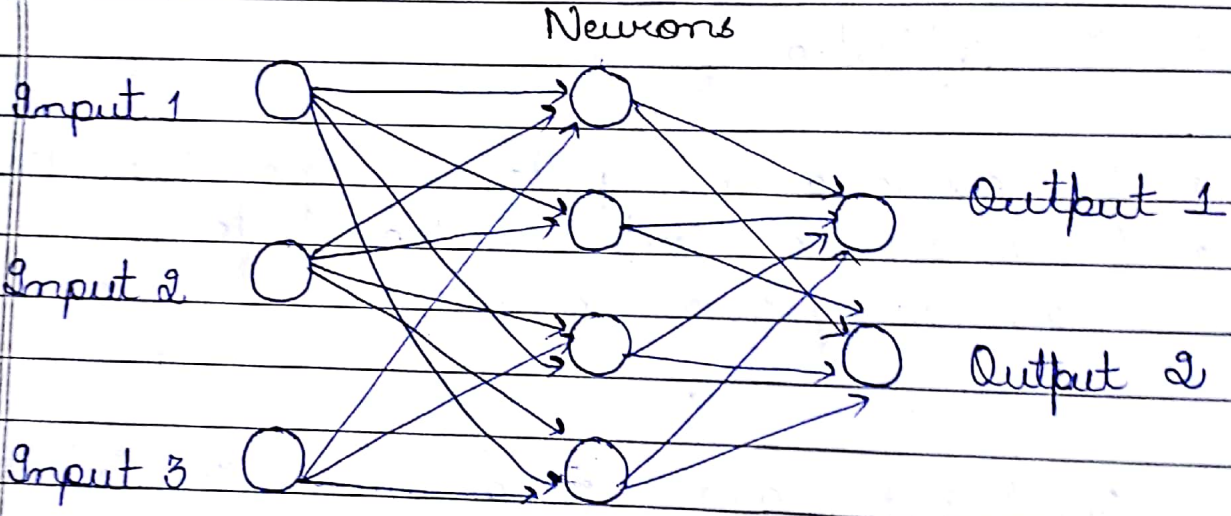
PAGE NO. :
DATE : / /

Artifical Intelligance
INT-404

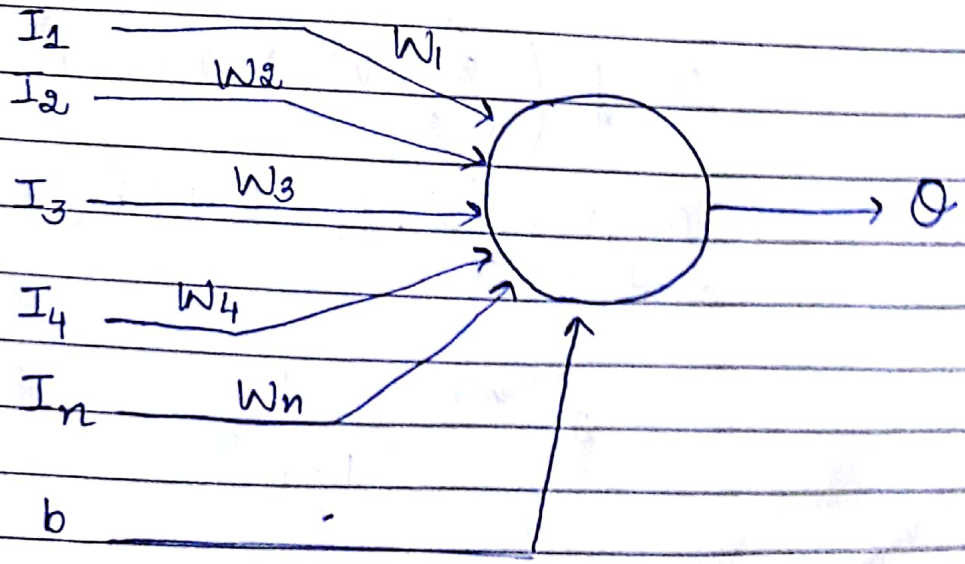
Artifical Neural Network
UNIT-6

★ **ANNs - How They Operate :-**

→ ANN represent a highly connected networks of neurons - the basic processing unit.



Structure of an Artifical Neuron :-



PAGE NO.:
DATE: / /

→ The neuron consists of no. of inputs. The info. is given as inputs via input connections,

→ b = bias
W = Weight
O = Output

→ The Processing of the Neuron :—

$$O = f\left(\sum_{i=1}^{n} W_i * I_i + b\right)$$

→ The Activation function :—

- The activation fun^n performs several imp. tasks.
- One of the most imp. function is to non-linearity of an equation.

Ques. Calculate the output for a neuron. The inputs are (0.10, 0.90, 0.05) and the corresponding weights are (2, 5, 2). Bias is given to be 1. The activation fun^n is Logistic. Also draw the neuron Architecture

$$O = f\left(\sum_{i=1}^{n} W_i * I_i + b\right)$$

where $I_1 = 0.1, W_1 = 2, I_2 = 0.9, W_2 = 5, I_3 = 0.05, W_3 = 2$ and $b = 1$

$$f(x) = \frac{1}{1 - e^{-x}}$$

PAGE NO.:
DATE: / /

$= f(W_1 * I_1 + W_2 * I_2 + W_3 * I_3)$

$= f(2 * 0.1 + 5 * 0.9 + 2 * 0.05)$

$= f(0.2 + 4.5 + 0.1)$

$= 1.008$ Ans.

* Training :—

Two Techq. for Training

1. Supervised Learning

In this kind, both the I/P and O/P are well determined and supplied to the training algo. Hence whenever an I/P is applied, we can calculate the error. We try to adjust the weights in such a way that this error is reduced.

2. Unsupervised Learing

In this type, the target O/P's are unknown. The I/P are applied, and the System is adjusted

PAGE NO. :
DATE : / /

GENETIC ALGORITHM

A genetic algorithm is a Search heuristic that is inspired by charles Darwin theory of natural evolution. This algorithm reflects the process of natural Selection where the fittest individuals are Selected for reproduction in order to produce offspring of the next generation.

Tutorial

PAGE NO. :
DATE : / /

2-04-18

Artificial Intelligence
INT-404

UNIT-5

Partition SN
Script

UNIT-6

ANN
GA
NLP
Game Playing
↳ MIN MAX
↳ α, β

* **Genetic Algorithm :—**

1. Five Phases of Genetic Algorithm

1. Initial Population
2. Fitness Function → how fit an individual is.
3. Selection → to select the fittest individual.
4. Crossover
5. Mutation

→ Termination

PAGE NO.:
DATE: / /

→ Applications of GA :-

1. Clustering
2. Image Processing
3. Data Analysis
4. Telecommunication

→ Steps of Genetic Algorithm :-

1. Initialization / Initial Population

The entire range of possible solution (Search Space) is selected.

2. Apply the fitness function on the generation and select two pair of individual based on fitness score.

if (Max fitness > threshold Value)
return current generation.

3. While (max fitness < threshold)

a.) Perform Crossover
b.) Perform Mutation
c.) update fittness
d.) Compare

4. Termination

Lecture

PAGE NO. :
DATE : / /

(NLP) Natural Language Processing

NLP Problems can be divided into two parts :—

1. Processing Written Text.
2. Processing Spoken Language.

→ **Steps in NLP :—**

1. **Morphological Analysis**

it means whenever we are writting a Sentence, it will extract meaningfull words from that Sentence.

2. **Syntactic Analysis**

after Catergorizing the Words, it will perform Syntactic Analysis. Build a Structure description of Sentence based.

3. **Semantic Analysis**

4. **Discourse Integration**

5. **Pragmatic Analysis**

It refers to intended meaning of Sentences used in different Contexts. The Context affects the interpretation of the Sentence.

In Min Max, we have to compare every Node which is drawback and to solve this we use Alpha Beta Pruning.

PAGE NO. :
DATE : / /

★ MiN MAX Algorithm :-

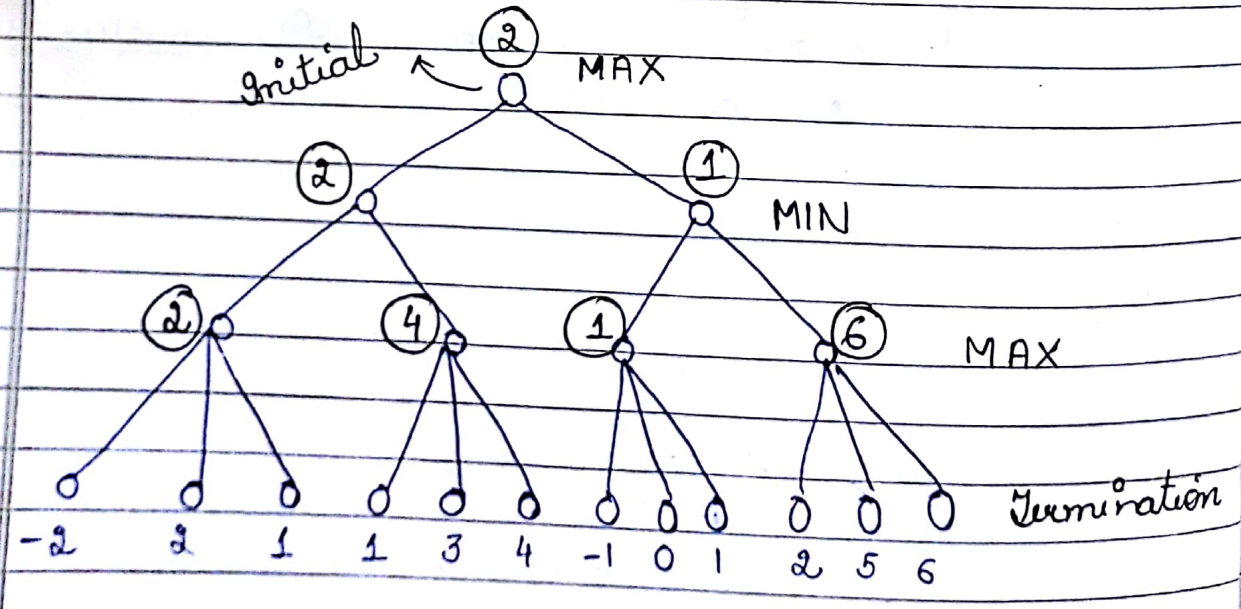
A game can be formally defined as a kind of search problem with the following components :

- The initial State
- A Set of operators
- A terminal Test
- A Utility function (also called Payoff function)

2 Players

1st Player - MAX → will Select the max utility Value - to Win

2nd Player - MIN → will Select the min utility Value



Max = $-\infty$

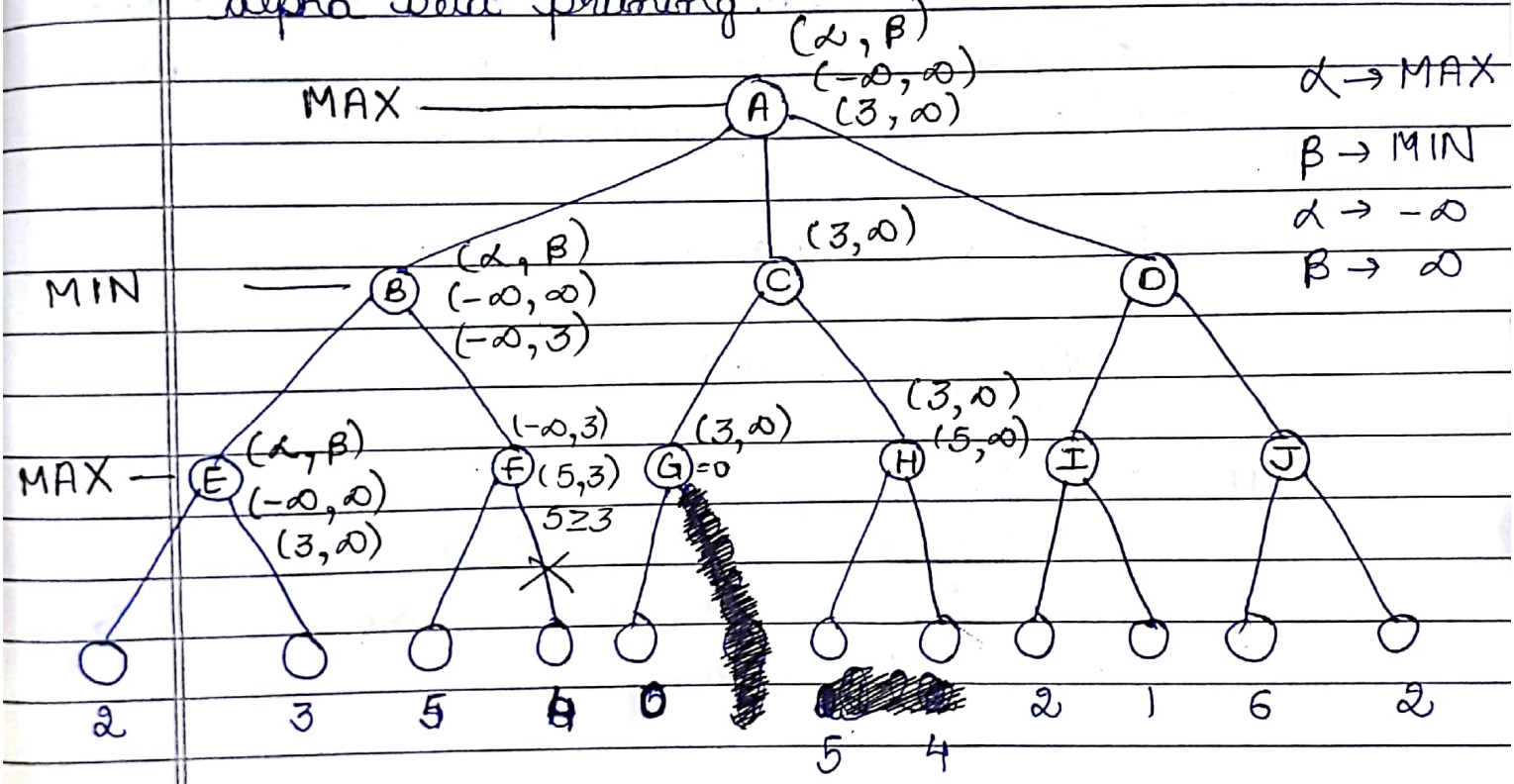
Min = $+\infty$

PAGE NO. :
DATE : / /

★ ALPHA BETA Pruning :—

Follows DFS approch.

The process of eliminating a branch of the Search tree from consideration without examining, it is called Pruning the Search Tree. The particular techq. we will examine is called alpha beta pruning.



Values will always transfered from Top to Bottom

$\alpha \geq \beta$

if its true, will Prun the next node.