Expt. No. _____ Date: _____ Page No:

ANUBHAV SINGH
IC-2k16-54
IC-703
ARTIFICIAL INTELLIGENCE LAB
MCA 9th Sem

Anubhav
29 Jan. 2021

Teacher's Signature : _____

Record of Work Done

Date	Expt. No.	Experiment	Page No.	Signature
	1	Lab assignment 1	01-12	
	2	Lab assignment 2	13-25	
	3	Lab assignment 3	26-34	

Anubhav

Expt. No. _____ Lab Assignment #1 Date: _____ Page No: 1

1. Represent the following facts in Prolog:

a. Tom is a singer.

```
singer (Tom).
```

b. Mia and John are married.

```
married (Mia , John).
```

c. Rani is dead.

```
dead (Rani).
```

d. John kills everyone who gives Mia a footmassage.

```
kills (John, X):-
        footmassage (X, mia).
```

e. Mia loves everyone who is a good dancer.

```
loves (Mia, X):-
        gooddancer (X).
```

f. Rahul eats anything that is nutritious or tasty.

```
eats (Rahul (X)):-
        nutritious (X),
        tasty (X).
```

Anubhav

Teacher's Signature : _____

Anubhav

Expt. No. _____ Date: _____ Page No: 2

2. Suppose we are working with the following knowledge base:

```
hasWand(harry).
quidditchPlayer(harry).
wizard(ron).
wizard(X) :- hasBroom(X), hasWand(X).
hasBroom(X) :- quidditchPlayer(X).
```

How does Prolog respond to the following queries?

```
wizard(ron
true

wizard(hermione)
false

wizard(harry)
true

wizard(Y)
Y = ron ;
Y = harry.
```

Anubhav

Teacher's Signature : _____

Anubhav

Expt. No. _____ Date: _____ Page No: 3

3. Define and describe the difference b/w knowledge, belief, hypothesis and data.

i. Knowledge: It can be defined as the body of facts and principles accumulated by humankind or fact or state of knowing.

ii. Belief: It is defined as essentially any meaningful and coherent expression that can be represented. Thus, a belief can be true or false.

iii. Hypothesis: It is defined as justified belief that is not known to be true. Thus a hypothesis is a belief that is backed up with some supporting evidence, but it may still be false.
In other words, it is priliminary assumption or tentative explanation that accounts for a set of facts, taken to be true for the purpose of investigating and testing.

iv. Data: Data in computer terminology means new facts and figures for eg. 'Anubhav', '201654', etc are data. Data are processed to form information.

Anubhav

Teacher's Signature : _____

Anushav

Expt. No. _____ Date: _____ Page No: 4

4. Write Prolog clauses for the following mgt system.

i. Edward manages Smith & Jackson.

manages (Edward, Smith),
manages (Edward, Jackson).

ii. Smith manages Jones, Harris & Peters.

manages (Smith, Jones),
manages (Smith, Harris),
manages (Smith, Peters).

iii. Harris manages carter & Fletcher.

manages (Harris, carter),
manages (Harris, Fletcher).

iv. Jackson manages Harper, Pritchard & Glover.

manages (Jackson, Harper),
manages (Jackson, Pritchard),
manages (Jackson, Glover).

Anushav

Teacher's Signature : _____

Anubhav

Expt. No. _____ Date: _____ Page No: 5

	PERSON	RESPONSIBLE FOR
a.	Jones	Finance
b.	carter	Sales
c.	Fletcher	Purchases
d.	Peters	Personnel
e.	Harper	Transport
f.	Pritchard	communications
g.	Glover	Industrial Relations

responsible (Jones, finance)

responsible (carter, Sales)

responsible (Fletcher, Purchase)

responsible (Peters, Personnel)

responsible (Harper, Transport)

responsible (Pritchard, Communication)

responsible (Glover, Industrial_Relations)

A person responsible for a took if he manages someone who is responsible for that book.

responsible (A, B) :-
manages (A, A),
responsible (A, B).

Anubhav

Teacher's Signature : _____

Anubhav

Expt. No. _____ Date: _____ Page No: 6

Answer the following question:

1. Is Smith responsible for transport.
responsible (Smith, transport).
False

2. Who are responsible for communication.
responsible (X, communication).
X = Pitchard
X = Edward
X = Jackson
False

3. Who are responsible for sales & personal.
responsible (X, sales).
X = Carter
X = Edward
X = Smith
X = Harris
False
responsible (X, personal).
X = Peters
X = Edward
X = Smith
False

Anubhav

Teacher's Signature : _____

Anushan

Expt. No. _____ Date: _____ Page No: 7

5. Implement the following problems in Prolog.

a. The Farmer, Wolf, Goat & Cabbage.

⇒ change (e, w)
⇒ change (w, e)

```
move([X;x, Goat, cabbage], wolf [Y;Y, Goat,
      cabbage]) :-
            change (X, Y).

move([x, wolf, X, cabbage], Goat [Y, wolf; Y,
      cabbage]):-
            change (X, Y).

move([x, wolf, goat, X], cabbage, [Y, Wolf, Goat,
     Y]) :-
            change (X, Y).

move ([X, Wolf, Goat, cabbage], nothing, [Y, Wolf,
  Goat, cabbage]):-
        change(X, Y).

oneEq(x, x, _).
oneEq(X, _, X).
```

Anushan

Teacher's Signature : _____

Anuvhav

Expt. No. _____

Date:

Page No: 8

```
safe([Man, Wolf, Goat, Cabbage]):-
    oneEq(Man, Goat, Wolf),
    oneEq(Man, Goat, Cabbage).

solution([e,e,e,e],[]).
solution(Config, [FirstMove | OtherMoves]):-
    move(Config, Move, NextConfig),
    safe(NextConfig),
    solution(NextConfig, OtherMoves).
```

Anuvhav

Teacher's Signature : _____

Anubhav

Expt. No. _____ Date: _____ Page No: 9

2. The Monkey & the Banana :-

```
domains:
    X = string

predicates

 take (X, X)
 move (X, X)
 get_on (X, X)
 hit (X, X, X)

 /* go clause */
 go :-
 take ("Monkey", "Stick"),
 move ("Monkey", "Chair"),
 get_on ("Monkey", "Chair"),
 hit ("Monkey", "Stick", "Banana"),
 write("The Monkey has hit the banana").

 go:-
 write ("The monkey could not reach to the
         banana").
 take (Animal, Object):-
 write("Does the ", "Animal", take the ", Object, "?"(Y/N")),
 readchar (Reply),
 Reply = 'y'.
```

Anubhav

Teacher's Signature : _____

Anushar

Expt. No. _____ Date: _____ Page No: 10

```
move(Animal, Object):-
write("Does the", Animal, "moves the", Object, "?(Y/N)"),
readchar(Reply),
Reply = 'Y'.

geton(Animal, Object):-
write("Does the", Animal, "Get on", Object, "?(Y/N)"),
readchar(Reply),
Reply = 'Y'.

hit(Animal, Object, Fruit):-
write("Does the, "Animal, "hit the, "Fruit", with
the", Object, "?(Y/N)"),
readchar(Reply),
Reply = 'Y'.
```

Anushar

Teacher's Signature : _____

Anubhav

Expt. No. _____ Date: _____ Page No: 11

c. Waterjug problem.

domain
x, y = Integers.

predicates
f(x,y)

clauses.

```
f(2,Y):- write("\n2 0 \n"),
f(X,2):- write("\n0 2 \n").

f(X,Y):- Y=0, X<>4!,
write("\n", X, "3"),
f(X,3).

f(X,Y):- X=0, Y<>3!,
write("\n 4", Y),
f(4,Y).

f(X,Y):- X+Y>=4; Y>0, X<>4, !,
    Y1 = Y-4+X.

write("\nX  4", Y2),
f(4,Y2); X+Y>=4, Y>0, Y<>3, !;
    Y2 = Y-4+X,
write("\n 4", Y1),
f(4,Y1).
```

Anubhav

Teacher's Signature : _____

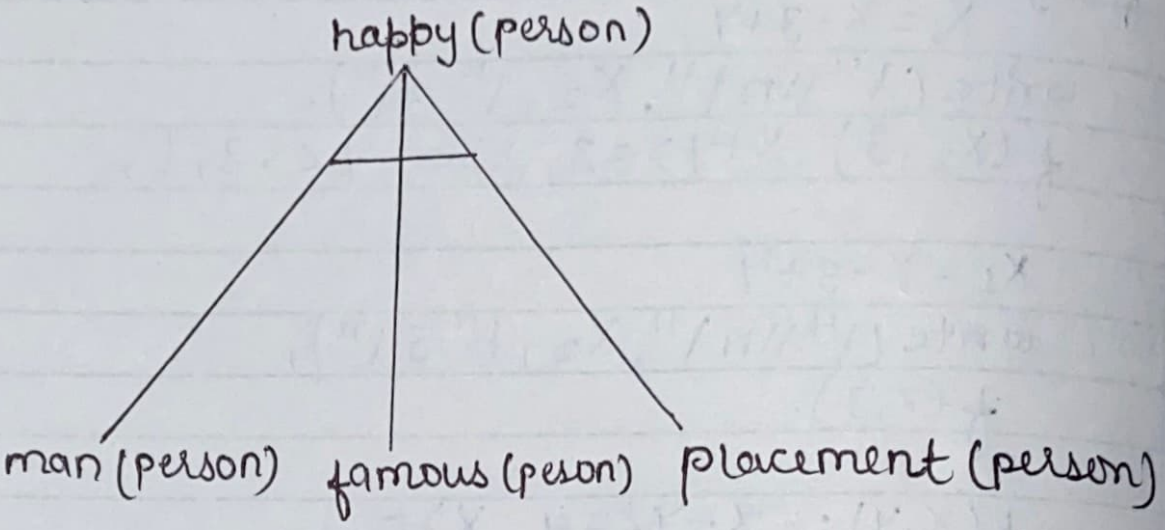
Anubhav

Expt. No. _____ Date: _____ Page No: 12

```
f(x,y):- x+y>=3, x>0, x<>4, !,
        X1 = X-3+Y,
    write("\n", X2, " 3"),
    f(X2,3), X+Y>=3, X>0, Y<>3, !,

    X1 = X-3+Y
    write("\n", X2, " 3"),
     f(X,3).

f(X,Y):- X+Y<=4, X>0, !,
       X1 = X+Y;
    write("\n", X1 " 0"),
     f(X,0).

f(X1,Y):- X+Y<=3, X>0, !,
     Y1 = X+Y
    write("\n 0", Y1),
     f(0,Y1)

f(X1,Y):- X>0, !,
    write("\n 0", Y)
     f(0,Y)

f(X,Y):- Y>0, !,
    write("\n", X, " 0"),
     f(X,0).
```

Anubhav

Teacher's Signature : _____



AND Tree

(fig 1)

Anubhav

Expt. No. _____ LAB ASSIGNMENT #2 Date: _____ Page No: 13

1. Explain the rules of conjuction & disjuction with a example & draw AND/OR tree.

Rule of conjuction:

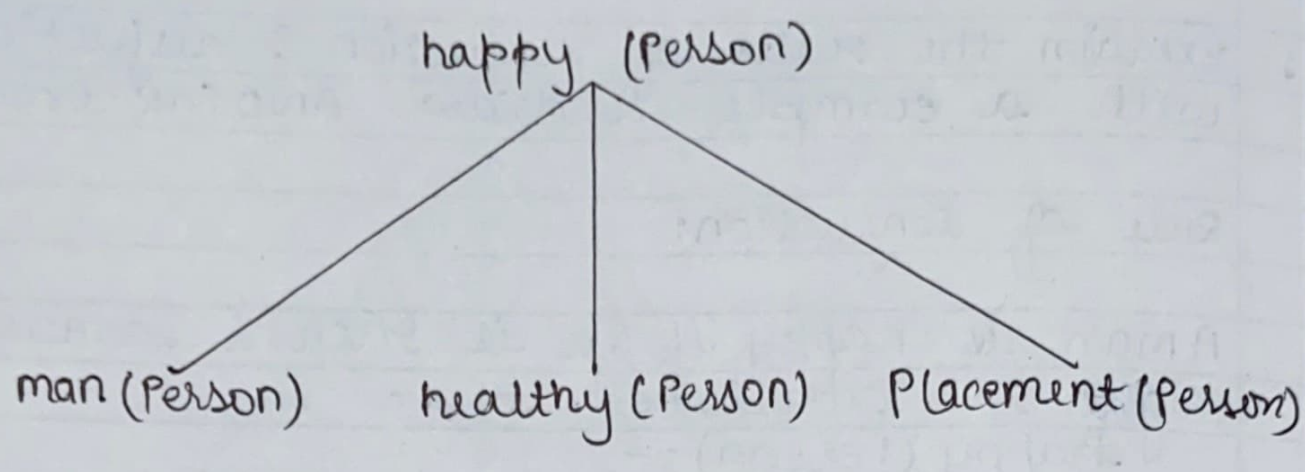
A man is happy if he is rich & famous might be translated to :-

happy (Person) :-
man (Person),
famous (Person),
placement (Person).

The ',' indicates the conjuction & is roughly equivalent to a predicate calculas. Therefore read ',' as 'AND'. Subgoals are cojoined.
We can represent conjunction using an AND tree. Here is an AND tree that represents above example. The way in which we descriminate b/w OR tree & AND tree, is the use of horizontal bar to link the rule goal. We need this distinction because we are going to seperate the structure of a program using a compiled AND/OR tree.

Anubhav

Teacher's Signature : _____



OR Tree

(fig 2)

Anubhav

Expt. No. _____ Date: _____ Page No: 14

Rule of Disjunction

Someone is happy if he is man, healthy or has placement.
Translate to:

happy (Person) :- man (Person)
happy (Person) :- healthy (Person)
happy (Person) :- Placement (Person)

Note that we have rewrite these stmts.
Someone is happy if they are men. OR
Someone is happy if they are wealthy. OR
Someone is happy if they have placements.

The predicate name "Happy" is known as function.
The function happy has one argument.
We describe a predicate with name 'predname' with argument 'n' as predname /n. It has one argument so we can say that, its asity is 1.

The predicate happy/1 is defined by 3 clauses.

The AND/OR tree is represented as given

Anubhav

Teacher's Signature : _____

Anubhav

Expt. No. _____ Date: _____ Page No: 15

2. Explain logical variables. Explain the diff. b/w logical variable & variables used in procedural language.

In prolog, logical variable enable you to write general facts & ask general questions. To represent a variable in prolog, captitative first letter of it.

Ex:- likes ("Rama", X) if likes ("bill", X).
The object x is a logical variable.
It might be able to match anything that Bill likes.

Logical variable V/S variable

Prolog has no assignment attachment, this is a significant distinction b/w them.

i. Variables in prolog get their value by being matched to constants in facts or rules, you can't store information in logical variables, unlike procedural language.

ii. Anonymus variables can be used in place of any other logical variables.

Anubhav

Teacher's Signature : _____

Anubhav

Expt. No. _____ Date: _____

Page No: 16

3. Write short note on following :-

a. First order predicate logic :-

First order predicate logic is a powerful language that develops info about the object in a more easy way & can also express relationship b/w the objects first order logic didn't assume that the world contains facts like propositional logic world contains facts the following :-

i. Object
ii. Relations
iii. Functions

As a natural language, first order also has two main facts :-

i. Syntax
ii. Sementics.

Basic elements of first order logic are:

i. Contact
ii. Variable
iii. Predicate
iv. Connecting equality.

Anubhav

Teacher's Signature : _____

Anubhav

Expt. No. _____ Date: _____

Page No: 17

b. Prepositional & predicate logic:-

Prepositional logic is the study of prepositions where a preposition is a statement that is either true or false. Preposition logic may be used to encode simple arguments that are expressed in natural language. It uses truth table & inference rule.

Predicate rules logic allows complex facts world to be represented & new fact may be determined via deduction reasoning predicate calculus include predicate variable & quantities.

c. Forward chaining vs backward chaining

Forward chaining starts from the known fact & more forward by applying inference rule to extract more data & it continues until it reaches to the goal, whereas backward by using inference rule to determine the fact that satisfy the goal.

Anubhav

Teacher's Signature : .....

Anuhar

Expt. No. _____ Date: _____

Page No: 18

- Forward chaining in data driven whereas backward chaining is goal driven.

- Forward chaining uses BFS strategy whereas backward chaining used DFS.

d. Unification :- Unification is a general method of matching a query to a fact each of which may contain variables. In order to determine contradictories, we need a matching procedure that compare 2 litrals & discourages weather there exists a set of substitutions that makes them identicle. This is unification Algorithm.

e. Fail Predicate :- If we want to force a rule to fail under certain condition Then we built in predicate called failure used in prolog. Predicate fail takes prolog interpreter to fail a predicate goal & subsequently forcd batch tracking.

Anuhar

Teacher's Signature : .....

Anubhav

Expt. No. _____ Date: _____

Page No: 19

f. Input Predicate: Prolog can read & write terms from one or several files. There thus form the stream of data that are either input or output.

The input predicates are:

read(x): read one clause from the current input & unify it with X. If there is no further input, X is unified with end of file.

see (file): Open file as the current input file.

seen: Close the current input file.

g. Backtracking: Backtracking is basically a form of searching. Prolog handles a computation that may have more than one results by backtracking i.e. undoing all work done since a tentative choice was made so an alternative choice can be made or tried. If a particular clause can be satisfied more then once. We know that prolog will go back & try to find all of those solutions.

Anubhav

Teacher's Signature : _____

Anubhav

Expt. No. =

Date:

Page No: 20

h. Computable functions & predicates:

It may be necessary to compute functions as post of fact. In this case a computable predicate is used. A computable predicate may include computable function such as +, −, *, etc. for ex:-
gt (x-y, 10) → bigger (x) contains the computable predicates gt which perform the greater then function. Note this computable predicates uses the computational function substraction.

Anubhav

=

Teacher's Signature :.....

Expt. No. Anubhav

Date:

Page No: 21

4. Assume you have database --- ex :-

beat (rani, rahul)
beat (tom, rani)
beat (jerry, tom)
you define a relation
class (Player, category)
that ranks player into categories,
you just have three categories.

winner: Every player who won all his games is a winner.

fighter: Any player that won some games & lost her games.

sportsman: Every player that has lost all his matches.

Sol: To specify the rules of a fighter.

x is a fighter if
There is some Y such that X beats Y &
There is some Z such that Z beats X.

Anubhav

Teacher's Signature :.....

Anubhav

Expt. No. ____

Date:

Page No: 22

Now rules for the winner:-
X is a winner if
X beat some Y &
Y was not beaten by anybody.

The rules for sportsman are:-
X is a sportman if
There is a Y who beats X &
No one is beaten by X

The formulation contains not which can't be directly expressed with our present prolog facilities. So the formulation of winner appears trickster. The same problem occurs with sportsman. The problem can be circumented by combining the definition of winner with that of fighter & we the otherwise connective.
Such a formation is:

If x beats somebody & X was beaten by somebody, then X is a fighter.
Otherwise, if x beats somebody
then X is a winner

Anubhav

Teacher's Signature :.....

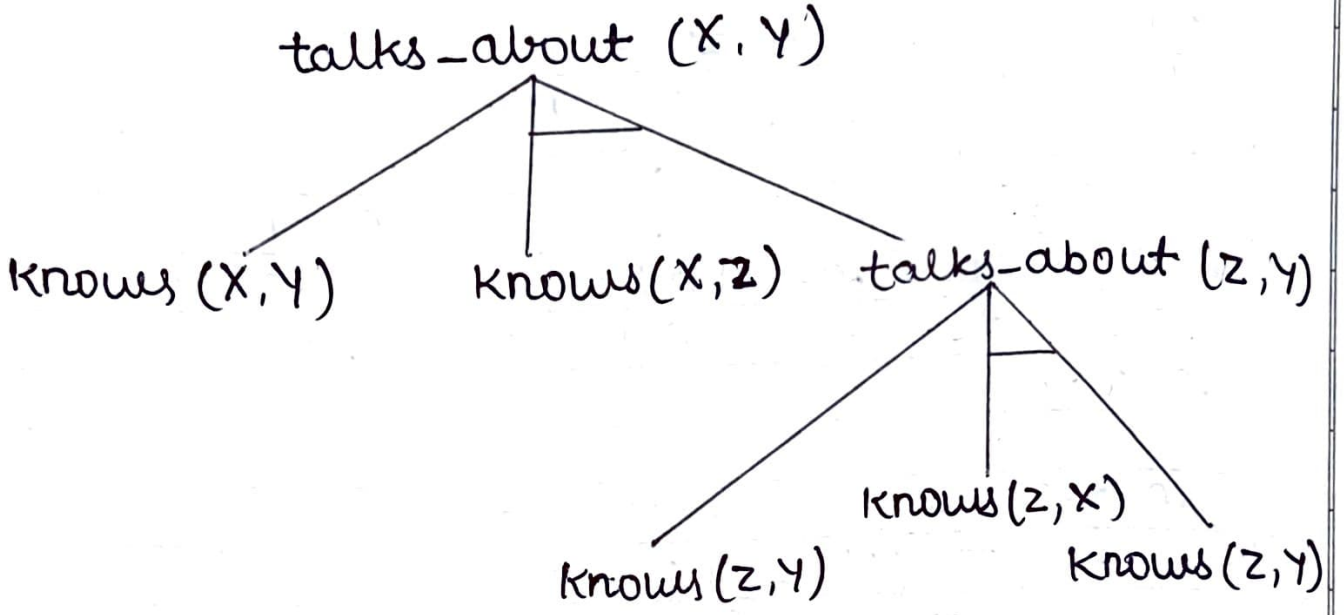
otherwise if x got beaten be somebody then X is sportman.

The formulation can be readily translated into prolog. The mutual exclusion of the three alternative categories is indicated by the cuts.

```
class (X, fighter):-
    beat (X, _),
    beat (_, X).

class (X, winner):-
    beat (X, _).

class (X, sportsman):-
    beat (_, X).
```

Anushan

Teacher's Signature :.....



AND/OR Tree for the program

(fig 3)

Expt. No. Anubhav

Date:

Page No: 24

5. Write output & draw AND/OR tree for the following program:-

```
talks_about (A,B):-
        know (A, B).

talks_about (P,R):-
        know (P, Q),
        talks_about (Q,R).
```

consider the following facts

```
know (rahi, ram).
know (ram, pat).
know (rani, fred).
know (fred, rani).
```

Explaination:
You talk about someone either you know them or you know someone who talks about them.

If you look AND/OR tree you see that :-

1. There is a such tree which is the same shape as the whole tree reflecting the simple recursion.

Anubhav

Teacher's Signature :.....

Expt. No. Answer

Date:

Page No: 25

2. The solution of given problem depend on being able to stop recursion at same point. Because the leftmost path down the tree is not infinite in length. It is responsible to help for a solution.

How the prolog respond to the following queries.!.

a. goal : talks_about (X, Y):
X = Rani.
Y = Ram.

b. goal : talks_about (Fred, X):
X = Rani.

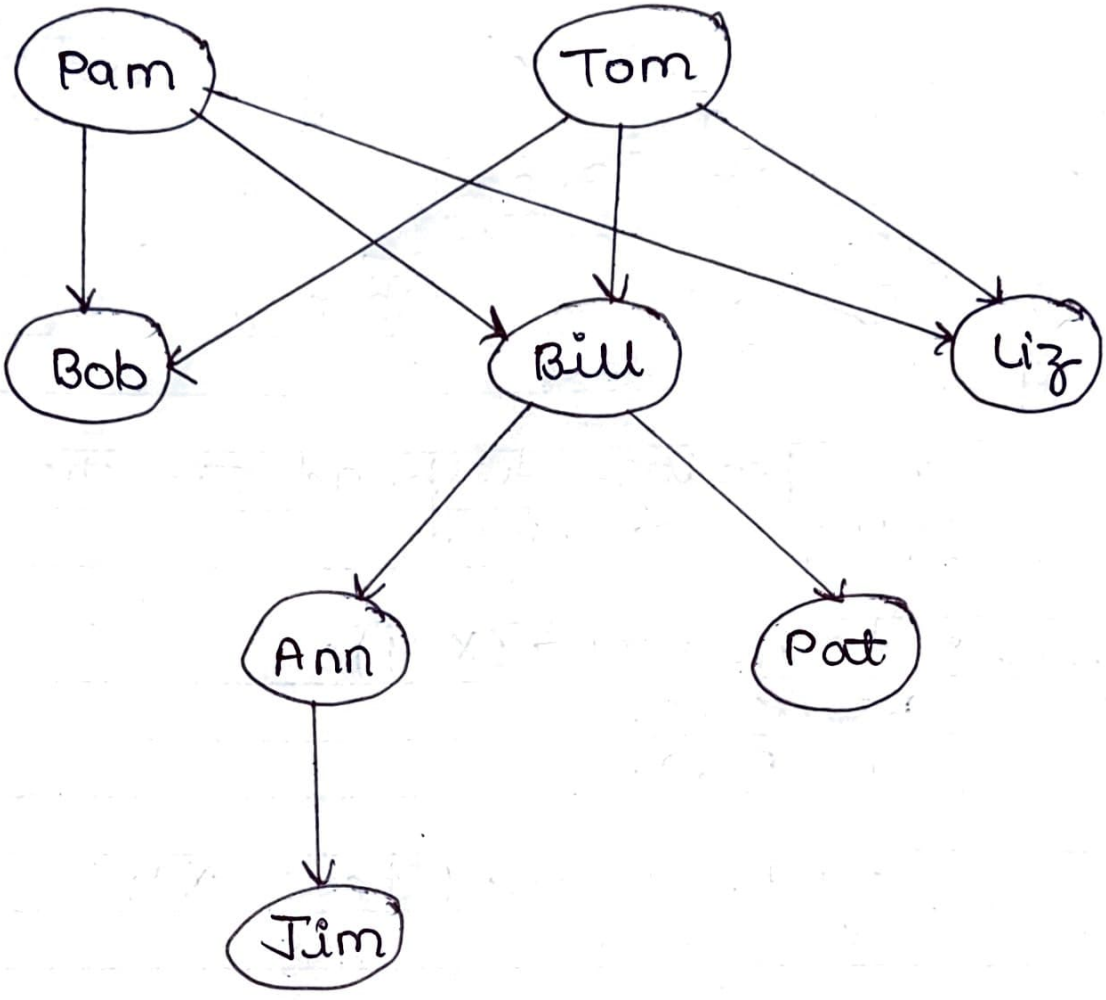
c. goal : talks_about (Ram, _):
True

d. goal : talks_about (_, Rani).
True

e. goal : talks_about (_, _)
True

Answer

Teacher's Signature :.....



(fig 4)

Anubhav

Expt. No. _____ LAB ASSIGNMENT #3 Date: _____

Page No: 26

1. Geneology problem

1.1 Build a knowledge base to represent the parent relationship can be deducted from the tree.

```
parent (child, parent)
parent (bob, pam)
parent (bob, tom)
parent (bill, pam)
parent (bill, tom)
parent (liz, pam)
parent (liz, tom)
parent (ann, bill)
parent (pat, bill)
parent (jim, ann)
```

1.2 Build predicates that describes the following family relationship.

a. grandparent

```
grandparent (grandchild, grandparent) :-
    parent (grandchild, parent),
    parent (parent, grandparent).
```

Anubhav

Teacher's Signature : .....

Anushar

Expt. No. ____

Date:

Page No: 27

b. Mother

```
mother(child, mother):-
    female(mother),
    parent(child, mother).
```

c. Father

```
father(child, father):-
    male(father),
    parent(child, father).
```

d. Brother

```
brother(sibling, bro):-
male(bro),
father(sibling, father),
father(bro, father),
bro = sibling,
mother(sibling, mother),
mother(bro, mother).
```

e. Sister

```
sister(sibling, sis):-
female(sis),
father(sibling, father),
father(sis, father)
sis = sibling,
mother(sibling, mother),
mother(sis, mother).
```

Anushar

Teacher's Signature : .....

Expt. No. Answer

Date:

Page No: 28

f. Aunt

i. Sister of once father or another.

```
aunt (kid, auntie) :-
    female (Auntie),
    parent (kid, parent),
    sister (parent, Auntie).
```

ii. The wife of once uncle.

```
aunt (kid, Auntie) :-
    female (Auntie),
    parent (kid, person),
    brother (person, brother),
    married (Auntie, brother).
```

g. Uncle :-

i. The brother of one's mother or father.

```
uncle (kid, uncle) :-
    male (uncle),
    parent (kid, person),
    brother (parent, uncle).
```

Answer

Teacher's Signature :.....

Expt. No. Anubhav

Date:

Page No: 29

ii. The husband of one's aunt

```
uncle (kid, uncle):-
    male (uncle),
    parent (kid, person),
    sister (person, sister),
    married (uncle, sister).
```

h. Ancestor :-

```
ancestor (Person, Ancestor):-
    parent (person, parent),
    ancestor (parent, ancestor).
```

Anubhav

Teacher's Signature :.....

Expt. No. Anubhav

Date:

Page No: 30

3. Consider the following sentence:-

a. Translate these sentences into formula in predicate logic & prolog clause form

- John likes all kind of food.

$\forall x : food(x) \rightarrow likes(Jones, x)$ /* Predicate logic */

- Apples are food.

```
food (apples).
```

- Dosa is food.

```
food (dosa).
```

- Anything anyone eats & not killed by its food.

$\forall x : \forall y : eats(x,y) \wedge \neg killed(x,y) \rightarrow food(y)$

- Bill eats peanut & still alive.

$eats(bill, peanut) \wedge \neg killed(bill, peanut)$.

Anubhav

Teacher's Signature :.....

Anubhav

Expt. No. _____ Date:

Page No: 31

iv. Liz eats everything bill eats.

$\forall x, food(x) \wedge eats(bill, x) \rightarrow eat(liz, x)$.

CNF form

Rule $P \rightarrow Q = \neg P \vee Q$

i. $\neg food(x_1) \vee likes(John, x_1)$

ii. food (apple)

iii. food (dosa)

iv. $\neg eats(x_2, Y) \wedge \neg killed(x_2, Y) \vee food(Y)$
$= \neg eats(x_2, Y) \vee killed(x_2, Y) \vee food(Y)$.

Rule used $\neg(P \wedge \neg Q) = \neg P \vee Q$

v.a. eats (bill, peanut)

b. $\neg killed$ (bill, peanut)

vi. $\neg food(x_3) \vee \neg eats(bill, x_3) \vee eats(liz, x_3)$.

Rules:- $(A \cap B) \rightarrow C$

$\Rightarrow \neg (A \cap B) \vee C$

$\Rightarrow \neg A \vee \neg B \vee C$

Anubhav

Teacher's Signature : .....

Expt. No. Anubhav

Date:

Page No: 32

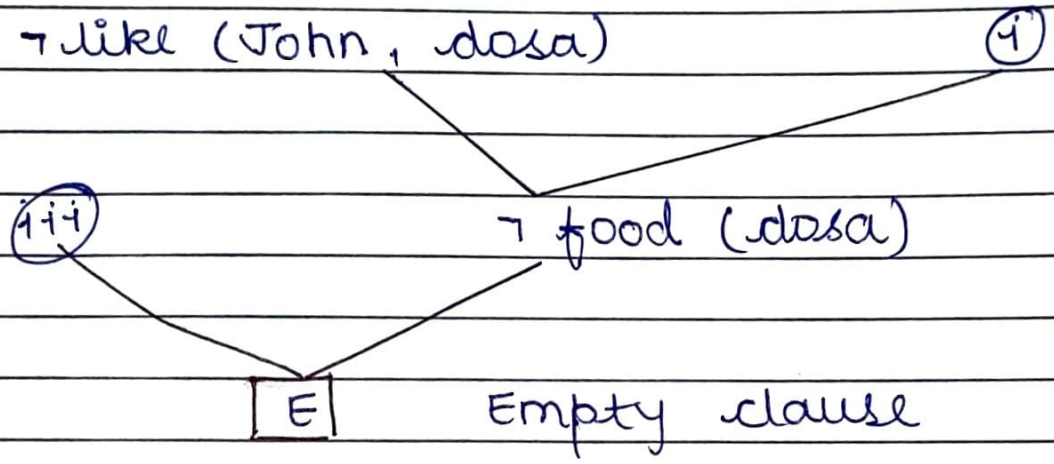
b. To prove:-

John likes dosa.

likes (John, dosa)

let us assume that john doesn't like dosa.

¬ like (John, dosa)



Since, an empty clause occurs means assumption is wrong. Therefore

John likes dosa Proved.

Liz like peanuts.

Suppose liz doesn't like peanuts.

¬ likes (liz, peanuts)

As. no CNF can be used. So. we can say the information is Insufficient.

Anubhav

Teacher's Signature :.....

Expt. No. Anubhav

Date:

Page No: 33

4. Consider the following sentences

a. Translate the sentence into formulas in predicate logic & prolog clause form.

i. Tom only likes hard courses.

hardcourses (x) → likes (Tom, x)

ii. Management courses are easy.

Easy can be converted into not hard

¬hardcourse (management)

iii. All courses in Engineering department are hard.

Eng_dept (x) → hardcourse (x).

iv. Computer engineering is an engg. dept course.

Engg-dept (computer Science)

Anubhav

Teacher's Signature :

Expt. No. Answer

Date:

Page No: 34

CNF FORM :-

Rules used:-

$P \rightarrow Q = \neg P \vee Q$

i. $\neg$hardcourse (x) V likes (Tom, x)

ii. $\neg$hardcourse (management)

iii. $\neg$Engg-dept (x_1) V hard courses (x_2)

iv. Engg-dept (computer Science)

b. Prove that "what course would tom like".

To prove likes (Tom, X) = ?

$\neg$likes (Tom, x) V likes (Tom, x) (i)

$\neg$hardcourse (x) V likes (tom, x_1) (iii)

x/x_2

(iv) $\neg$Engg-dept (x_2) V likes (Tom, x_2)

x_2 = Computer Science

likes (Tom, Computer Science)

Hence, Tome would like Computer Science

Answer

Teacher's Signature :