TRIBHUVAN UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY

HIMALAYA COLLEGE OF ENGINEERING

CHYASAL,LALITPUR

Lab Report No:- 3

Title:- Realizing family Tree using FOPL in prolog.

Submitted by:-

Name:- Swajan Shrestha

Roll No:- 38

Submitted To:- Rubas Mali

Department Of CSIT

Checked by:- Rubas

Date of submission:- 2081/04/09

TITLE: Realizing family Tree using first order Predicate Logic (FOPL) in prolog

- OBJECTIVE:

→ The primary objective of this lab is to create and understand a family tree using FOPL in prolog. The report will cover the following:
  * Introduction to family trees.
  * Concepts of predicate logic & propositional Logic.
  * Implementation of a family tree in prolog using FOPL.
  * Writing and understanding prolog rules for family relationships.
  * Executing and validating queries on the family tree.

- THEORY:

* family Tree:

↳ A family tree is a graphical representation of familial relationships in a structured tree format. It shows how family members are related to each other. In a family tree, each node represents a family member, and the connections between nodes represent the relationships (such as parent-child, siblings, spouses, etc)

Predicate logic :

→ It is also known as First-order logic (FOL) or FOPL, extends propositional logic by dealing with predicates and quantifiers. It allows us to express statements involving objects and their properties, relationships and quantification over objects.

• Basic Concepts of predicate logic :

→ predicates : function that return true or false, such as 'male(x)', 'female(y)', etc.

→ Terms : objects or constants, such as 'dashrath', 'ram', etc.

→ variables : symbols that can represent any object, such as 'x', 'y', etc.

→ Quantifiers : '∀' (universal quantifier) and '∃' (existential quantifier).

→ Logical connectives : '∧' (and), '∨' (or), '→' (implies), '¬' (not).

• propositional logic :

→ propositional logic, or Boolean logic, deals with propositions that can either be true or false. It uses logical connectives to form complex logical statements from simpler ones.

- **Basic Concepts of propositional logic:**
  * propositions: statements that are either true or false.
  * Logical connectives: '^', 'v', '→', '¬'.

- **Knowledge-Based systems:**

→ A knowledge-based system is an AI system that uses knowledge about a specific domain to act intelligently. It consists of:

* Knowledge Base: Contains domain-specific facts and rules.
* Inference Engine: uses logical reasoning to infer new information from the knowledge base.

In this lab, prolog will serve as both the knowledge base and the inference engine, allowing us to model and query the family tree.

OBSERVATION:

→ Implementation in prolog

* code:

```
% facts used----
male (dashrath).
male (ram).
male (laxman).
male (bharat).
```

```
male (luv).
male (kush).

female (kaushalya).
female (sita).
female (urmila).
female (daughter_of_dashrath).

father (dashrath, ram).
father (dashrath, laxman).
father (dashrath, bharat).
father (ram, luv).
father (ram, kush).
father (laxman, son_of_laxman).
father (dashrath, daughter_of_dashrath).

husband (dashrath, kaushalya).
husband (ram, sita).
husband (laxman, urmila).

% RULES - - - -
grandfather (X, Y) :-
        father (X, Z),
        father (Z, Y).
mother (X, Y) :-
        father (Z, Y),
        husband (Z, X).
brother (X, Y) :-
    father (Z, X),
    father (Z, Y),
    male (X),
```

```
X \=Y.

uncle(X,Y):-
    brother(X,Z),
    father(Z,Y).

listbrother(X):-
    brother(Z,X),
    write(Z).

aunty(X,Y):
    sister(X,Z),
    father(Z,Y).

Aunty(X,Y):
    wife(X,Z),
    uncle(Z,Y).

sister(X,Y):-
    father(Z,X),
    father(Z,Y),
    female(X),
    X\=Y.

wife(X,Y):-
    husband(Y,X).

cousin(X,Y):-
    father(Z,X),
    (uncle(Z,Y); aunty(Z,Y)).
```

Queries:

→ we can run various queries to infer relationships within the family tree:

a, grandfathers of all:

↳ prolog:

```
grandfather(A,B).
```

grandfather(A,B).

A	B
dashrath	luv
dashrath	kush
dashrath	son_of_laxman

false

```
?- grandfather(A,B).
```

b, All the mother's of :

→ prolog:

```
mother(A,B).
```

mother(A,B).

A	B
kaushalya	ram
kaushalya	laxman
kaushalya	bharat
sita	luv
sita	kush
urmila	son_of_laxman
kaushalya	daughter_of_dashrath

```
?- mother(A,B).
```

c> All the brother's :

→ prolog :

```
brother(A,B).
```

brother(A,B).

A	B
ram	laxman
ram	bharat
ram	daughter_of_dashrath
laxman	ram
laxman	bharat
laxman	daughter_of_dashrath
bharat	ram
bharat	laxman
bharat	daughter_of_dashrath
luv	kush
kush	luv

false

```
?- brother(A,B).
```

d. All the uncles :

→ prolog :

```
uncle(A,B).
```

uncle(A,B).

A	B
ram	son_of_laxman
laxman	luv
laxman	kush
bharat	luv
bharat	kush
bharat	son_of_laxman

```
?- uncle(A,B).
```

e> All the Aunty :

→ prolog :

```
Aunty(A,B).
```

aunty(A,B).

A	B
daughter_of_dashrath	luv
daughter_of_dashrath	kush
daughter_of_dashrath	son_of_laxman
sita	son_of_laxman
urmila	luv
urmila	kush

```
?- aunty(A,B).
```

5) All the sisters:

→ Prolog:

```
sister(B,A).
```

sister(B,A).

B	A
daughter_of_dashrath	ram
daughter_of_dashrath	laxman
daughter_of_dashrath	bharat

```
?- sister(B,A).
```

6) All the cousins:

→ Prolog:

```
cousin(A,B).
```

cousin(A,B)

A	B
luv	son_of_laxman
kush	son_of_laxman
son_of_laxman	luv
son_of_laxman	kush

```
?- cousin(A,B).
```

DISCUSSION:

→ This lab demonstrates how to model a family tree using first-order predicate logic in prolog. By defining facts and rules, we can infer various familial relationships and validate them through queries. This approach showcases the power of the predicate logic and knowledge-based systems in representing and reasoning about complex domains.

CONCLUSION:

→ we successfully created a family tree using first-order predicate logic in prolog, defined various familial relationships through facts and rules, and validated these relationships through prolog queries. This exercise provided a comprehensive understanding of predicate logic, propositional logic and knowledge-based systems in AI.

This concludes the detailed lab report on realizing a family tree using FOPL in prolog.

Checked