



University of Asia Pacific

Department Of Computer Science and Engineering

Assignment - 01

Report on:

Implementation of a basic family tree structure of your own family using Prolog.

Submitted To:

Dr. Nasima Begum

Assistant Professor

University of Asia Pacific

Submitted By:

Name: Alma Tanjin

ID: 18101088

Sec : B2

1. Problem Title:

Implementation of a basic family tree structure of my own family using Prolog .

2. Problem Description:

The problem is given implementation of a basic family tree structure of my own family using Prolog where some rules are against degree and removal for up to 3rd degree and twice removed situations. For rules against up to 3rd degree twice removed the family tree's level must have more than level 5. Because if the family tree's level is less than level 6 then this “3rd degree twice removed” rule always returns false. So, for this up to 3rd degree twice removed rule my family tree has level 0 –level 6.

Suppose we want to find 1st degree third removed relation between **X**,**Y** then **X** and **Z** must be 1st cousin and **Z** should be great grandparent of **Y**.

Suppose we want to find 3rd degree twice removed relation between **X** and **Y** then **X** and **Z** must be 3rd cousin and **Z** should be grandparent of **Y**.

		Relatives				
Separation in generation to ancestor $R \rightarrow$		2	3	4	5	6
Subject	↓ S Relation to ancestor	Grandparent	Great grandparent	Great-great grandparent	Great-great-great grandparent	Great-great-great-great grandparent
	2 Grandparent	1st cousin	1st cousin once removed	1st cousin twice removed	1st cousin third removed	1st cousin fourth removed
	3 Great-grandparent	1st cousin once removed	2nd cousin	2nd cousin once removed	2nd cousin + removed	2nd cousin third removed
	4 Great-great grandparent	1st cousin twice removed	2nd cousin once removed	3rd cousin	3rd cousin once removed	3rd cousin twice removed
	5 Great-great-great grandparent	1st cousin third removed	2nd cousin twice removed	3rd cousin once removed	4th cousin	4th cousin once removed
	6 Great-great-great-great grandparent	1st cousin fourth removed	2nd cousin third removed	3rd cousin twice removed	4th cousin once removed	5th cousin

Table 1: R S in the separation

Degree: $(\min(R, S) - 1)$

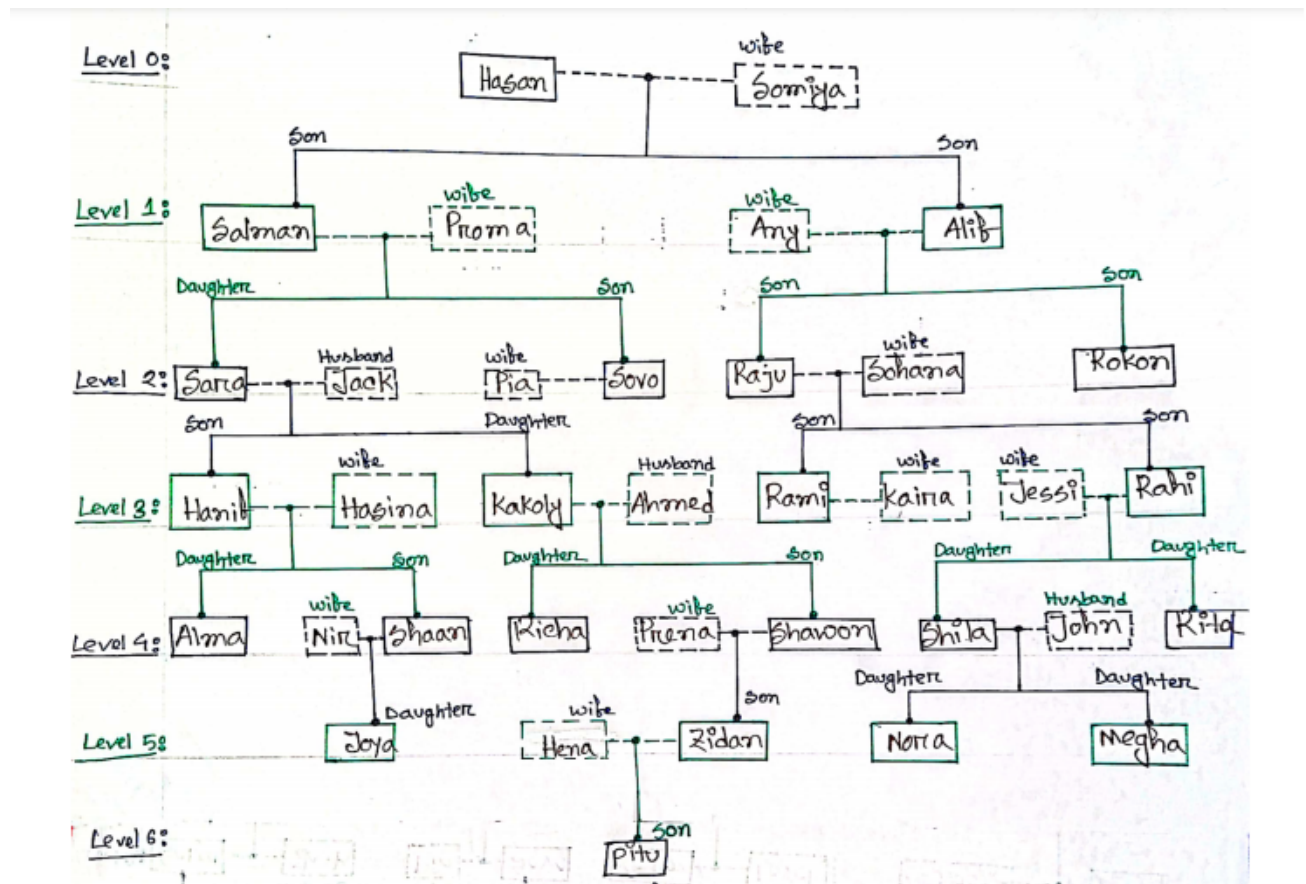
Removal: $|R - S|$

3. Tools and Languages Used:

For programming I used Prolog .

4. Diagram:

It's a simple family tree diagram with level 0 – level 6.



5. Sample Input/Output:

Some related result's screenshots are given below:

First_cousin

```
% e:/AI LAB/assignment.pl compiled
?- first_cousin(X,Y).
X = sara,
Y = rokon ;
X = sara,
Y = raju ;
X = sovo,
Y = rokon ;
X = sovo,
Y = raju ;
X = rokon,
Y = sara ;
X = rokon,
Y = sovo ;
X = raju,
Y = sara ;
X = raju,
Y = sovo ;
X = sara,
Y = rokon ;
X = sara,
Y = raju ;
X = sovo,
Y = rokon ;
X = sovo,
Y = raju ;
X = rokon,
Y = sara ;
X = rokon,
Y = sovo ;
X = raju,
Y = sara ;
X = raju,
Y = sovo ;
X = alma,
Y = shawon .
```

Second_cousin

```
?- second_cousin(X,Y).
X = hanif,
Y = rami ;
X = hanif,
Y = rahi ;
X = hanif,
Y = rami ;
X = hanif,
Y = rahi ;
X = kakoly,
Y = rami ;
X = kakoly,
Y = rahi ;
X = kakoly,
Y = rami ;
X = kakoly,
Y = rahi ;
X = rami,
Y = hanif ;
X = rami,
Y = kakoly ;
X = rami,
Y = hanif ;
X = rami,
Y = kakoly ;
X = rahi,
Y = hanif ;
X = rahi,
Y = kakoly ;
X = rahi,
Y = kakoly ;
X = rahi,
Y = hanif ;
X = rahi,
Y = kakoly ;
X = joya,
Y = zidan .
```

Cousin

```
?- first_cousin(Who,alma).
Who = shawon ;
Who = richa ;
Who = shawon ;
Who = richa ;
false.

?- third_cousin(Who,alma).
Who = shila ;
Who = shila ;
Who = rita ;
Who = rita ;
false.
```

Grandparent

```
?- great_greatgrandparent(Who,alma).
Who = hasan ;
Who = somiya .

?- great_greatgrandmother(Who,alma).
Who = somiya .

?- great_greatgrandfather(Who,alma).
Who = hasan .

?- greatgrandchild(alma,Who).
Who = salman ;
Who = proma .
```

```

?- second_cousin(Who,hanif).      ?- granddaughter(alma,Who).
Who = rami ;                      Who = sara ;
Who = rami ;                      Who = jack ;
Who = rahi ;
Who = rahi ;
false.

```

Third_cousin

```

?- third_cousin(X,Y).
X = alma,
Y = shila ;
X = alma,
Y = rita ;
X = alma,
Y = shila ;
X = alma,
Y = rita ;
X = shaan,
Y = shila ;
X = shaan,
Y = rita ;
X = shaan,
Y = shila ;
X = shaan,
Y = rita ;
X = shawon,
Y = shila ;
X = shawon,
Y = rita ;
X = shawon,
Y = shila ;
X = shawon,
Y = rita ;
X = shila,
Y = alma ;
X = shila,
Y = shaan ;
X = shila,
Y = shawon ;

```

First_degree_once_removed

```

?- first_degree_once_removed(X,Y).
X = sara,
Y = rami ;
X = sara,
Y = rahi ;
X = sovo,
Y = rami ;
X = sovo,
Y = rahi ;
X = rokon,
Y = hanif ;
X = rokon,
Y = kakoly ;
X = raju,
Y = hanif ;
X = raju,
Y = kakoly ;
X = sara,
Y = rami ;
X = sara,
Y = rahi ;
X = sovo,
Y = rami ;
X = sovo,
Y = rahi ;
X = rokon,
Y = hanif ;
X = rokon,
Y = kakoly ;
X = raju,
Y = hanif ;
X = raju,
Y = kakoly ;

```

```

?- first_degree_once_removed(Who,alma).
Who = zidan ;
Who = zidan ;
false.

```

```

?- first_degree_twice_removed(Who,alma).
Who = rokon ;
Who = raju ;
Who = rokon ;
Who = raju ;
Who = pitu ;
Who = pitu ;
false.

```

First_degree_twice_remove

```
?- first_degree_twice_removed(X,Y).
X = sara,
Y = shila ;
X = sara,
Y = rita ;
X = sovo,
Y = shila ;
X = sovo,
Y = rita ;
X = rokon,
Y = alma ;
X = rokon,
Y = shaan ;
X = rokon,
Y = shawon ;
X = rokon,
Y = richa ;
X = raju,
Y = alma ;
X = raju,
Y = shaan ;
X = raju,
Y = shawon ;
X = raju,
Y = richa ;
X = sara,
Y = shila ;
X = sara,
Y = rita ;
X = sovo,
Y = shila ;
X = sovo,
Y = rita ;
X = rokon,
Y = alma ;
X = rokon,
Y = shaan ;
X = rokon,
Y = shawon ;
X = rokon,
Y = richa ;
```

First_degree_third_removed

```
?- first_degree_third_removed(X,Y).
X = sara,
Y = nora ;
X = sara,
Y = megha ;
X = sovo,
Y = nora ;
X = sovo,
Y = megha ;
X = rokon,
Y = joya ;
X = rokon,
Y = zidan ;
X = raju,
Y = joya ;
X = raju,
Y = zidan ;
X = sara,
Y = nora ;
X = sara,
Y = megha ;
X = sovo,
Y = nora ;
X = sovo,
Y = megha ;
X = rokon,
Y = joya ;
X = rokon,
Y = zidan ;
X = raju,
Y = joya ;
X = raju,
Y = zidan ;
X = nora,
Y = sara ;
```

First_degree_fourth_removed

```
?- first_degree_fourth_removed(X,Y).
X = rokon,
Y = pitu ;
X = raju,
Y = pitu ;
X = rokon,
Y = pitu ;
X = raju,
Y = pitu ;
X = pitu,
Y = rokon ;
X = pitu,
Y = raju ;
X = pitu,
Y = rokon ;
X = pitu,
Y = raju ;
false.
```

Second_degree_third_removed

```
?- second_degree_third_removed(X,Y).
X = rami,
Y = pitu ;
X = rahi,
Y = pitu ;
X = rami,
Y = pitu ;
X = rahi,
Y = pitu ;
X = pitu,
Y = rami ;
X = pitu,
Y = rahi ;
X = pitu,
Y = rami ;
X = pitu,
Y = rahi ;
false.
```

Second_degree_once_removed

```
?- second_degree_once_removed(X,Y).
X = rami,
Y = alma ;
X = rami,
Y = shaan ;
X = rahi,
Y = alma ;
X = rahi,
Y = shaan ;
X = rami,
Y = alma ;
X = rami,
Y = shaan ;
X = rahi,
Y = alma ;
X = rahi,
Y = shaan ;
X = rami,
Y = shawon ;
X = rami,
Y = richa ;
X = rahi,
Y = shawon ;
X = rahi,
Y = richa ;
X = rami,
Y = shawon ;
X = rami,
Y = richa ;
X = rahi,
Y = shawon ;
X = rahi,
Y = richa ;
X = hanif,
Y = shila ;
X = hanif,
Y = rita ;
X = kakoly,
Y = shila ;
?
```

Second_degree_twice_removed

```
?- second_degree_twice_removed(X,Y).
X = rami,
Y = joya ;
X = rahi,
Y = joya ;
X = rami,
Y = joya ;
X = rahi,
Y = joya ;
X = rami,
Y = zidan ;
X = rahi,
Y = zidan ;
X = rami,
Y = zidan ;
X = rahi,
Y = zidan ;
X = hanif,
Y = nora ;
X = hanif,
Y = megha ;
X = kakoly,
Y = nora ;
X = kakoly,
Y = megha ;
X = hanif,
Y = nora ;
X = hanif,
Y = megha ;
X = kakoly,
Y = nora ;
?
```

Third_degree_once_removed

```
?- third_degree_once_removed(X,Y).
X = alma,
Y = nora ;
X = alma,
Y = megha ;
X = alma,
Y = nora ;
X = alma,
Y = megha ;
X = shaan,
Y = nora ;
X = shaan,
Y = megha ;
X = shaan,
Y = nora ;
X = shaan,
Y = megha ;
?
```

Third_degree_twice_removed

```
?- third_degree_twice_removed(X,Y).
X = shila,
Y = pitu ;
X = shila,
Y = pitu ;
X = rita,
Y = pitu ;
X = rita,
Y = pitu ;
X = pitu,
Y = shila ;
X = pitu,
Y = shila ;
X = pitu,
Y = rita ;
?
```


Other relations

```

?- sibling(alma,Who).
Who = shaan ;
Who = shaan.

?- brother(Who,alma).
Who = shaan ,
Who = shaan.

?- sister(alma,shaan).
true.

?- sister_in_law(Who,alma).
Who = nir ,
Who = nir.

?- daughter_in_law(nir,Who).
Who = hanif ;
Who = hasina ,
Who = hasina.

?- wife(nir,shaan).
true.

?- mother_in_law(Who,nir).
Who = hasina ,
Who = hasina.

?- father_in_law(Who,nir).
Who = hanif ,
Who = hanif.

?- great_great_gratgrandparent(X,Y).
X = hasan,
Y = joya ;
X = somiya,
Y = joya ;
X = hasan,
Y = zidan ;
X = somiya,
Y = zidan ;
X = hasan,
Y = nora ,
Y = nora.

?- greatgrandson(Who,salman).
Who = shaan ;
Who = shawon ;
false.

?- father(Who,alma).
Who = hanif ;
false.

?- mother(Who,alma).
Who = hasina.

?- couple(hanif,X).
X = hasina.

?- child(Who,hanif).
Who = alma ;
Who = shaan.

?- grandchild(Who,hasina).
Who = joya.

?- grandson(X,salman).
X = hanif ;
false.

?- granddaughter(X,salman).
X = kakoly ;
false.

?- son(Who,shawon).
Who = zidan.

?- sister(Who,hanif).
Who = kakoly ,
Who = kakoly.

?- niece(Who,alma).
Who = joya ,
Who = joya.

?- aunt(Who,alma).
Who = kakoly ,
Who = kakoly.

?- uncle(Who,rami).
Who = rokon ,
Who = rokon.

?- nephew(Who,alif).
Who = sovo ;
Who = sovo ;
false.

```

6. Conclusion:

Prolog is a declarative language .It is one of the most used languages in AI. In the knowledge base we declare some facts and rules and we can query against the knowledge base.

In this problem I wrote more than 50 rules against some relations but rules against degree removal was quiet difficult but I used some rules recursively so it became simple to understand.