Sardar Patel Institute of Technology
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058, India
(Autonomous College Affiliated to University of Mumbai)

Experiment 05

Aim: To design and implement an intelligent system, incorporating the matching algorithm and the rule language.

1. It should provide a fact base updating function.
2. It should provide a function that checks the rules LHS and return which rules were matched.
3. It should support firing RHS according to matches. Using SWISH Prolog or Java or Python or any other open-source tool

Problem: Determine parent, child and ancestral relationship

Theory:

- **Prolog**

Intelligent systems in AI are designed to mimic human intelligence and perform tasks that typically require human-like cognitive abilities, such as learning, reasoning, problem-solving, and decision-making.

Prolog stands for programming in logic. In the logic programming paradigm, prolog language is most widely available.

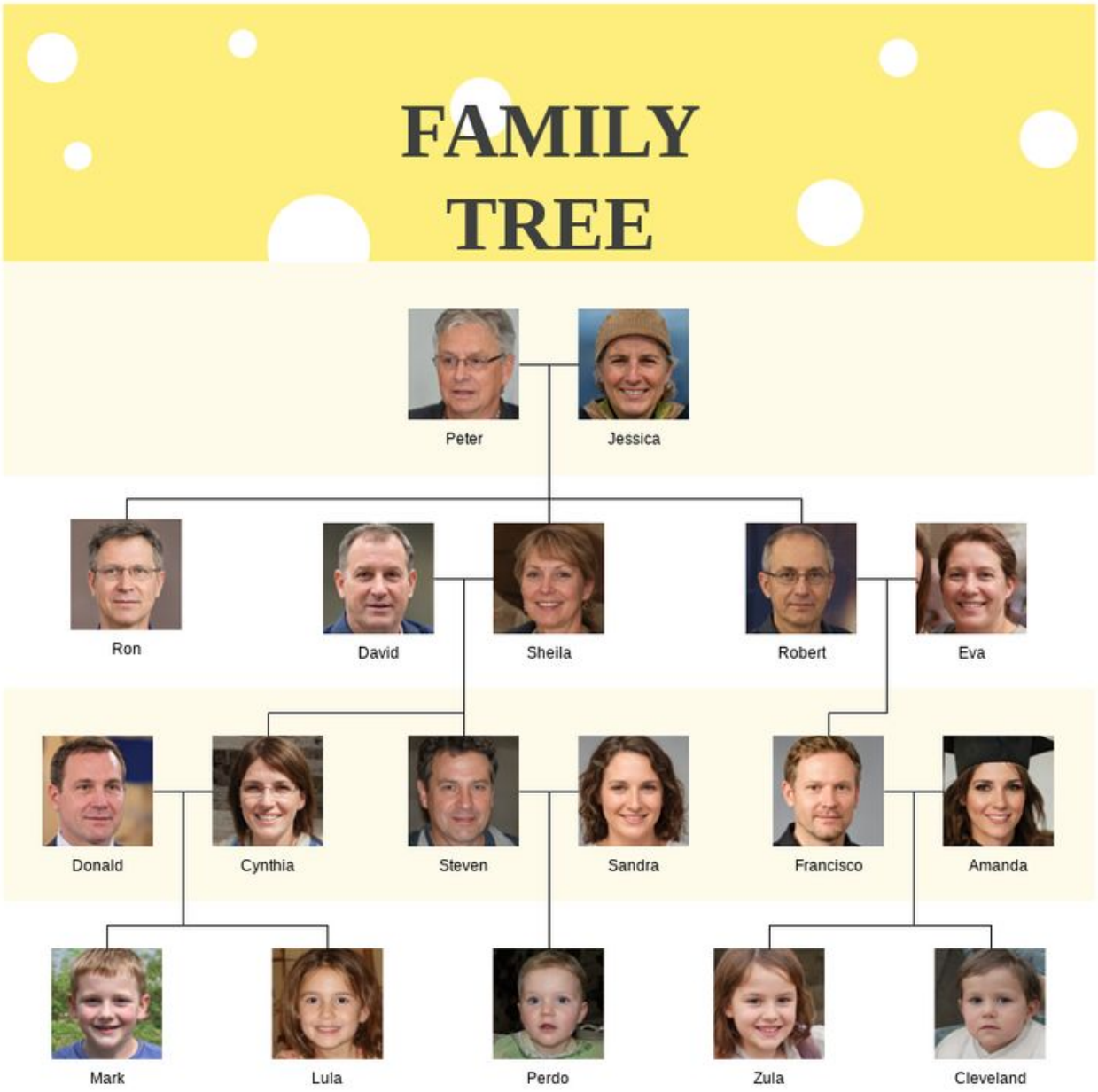
- **Knowledge representation**

Encoding information in a format that an intelligent system can understand and use. Prolog excels at knowledge representation through the use of facts and rules.

- **Inference and REasoning**

Intelligent systems need to make inferences based on available knowledge. Prolog's logical inference engine enables rule-based reasoning.

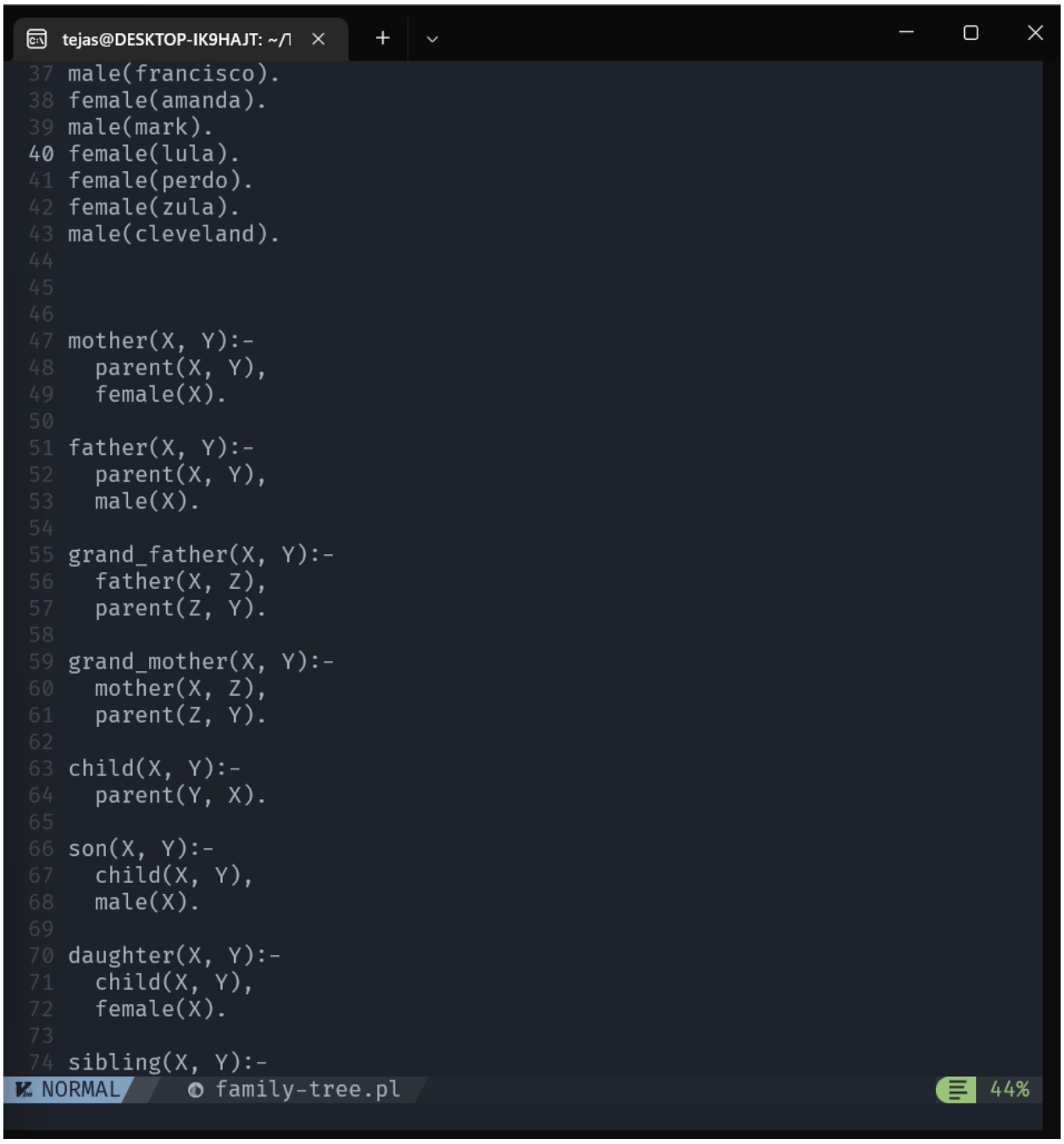
Family Tree:


FAMILY
TREE
Peter
Jessica
Ron
David
Sheila
Robert
Eva
Donald
Cynthia
Steven
Sandra
Francisco
Amanda
Mark
Lula
Perdo
Zula
Cleveland

Code:

tejas@DESKTOP-IK9HAJT: ~/

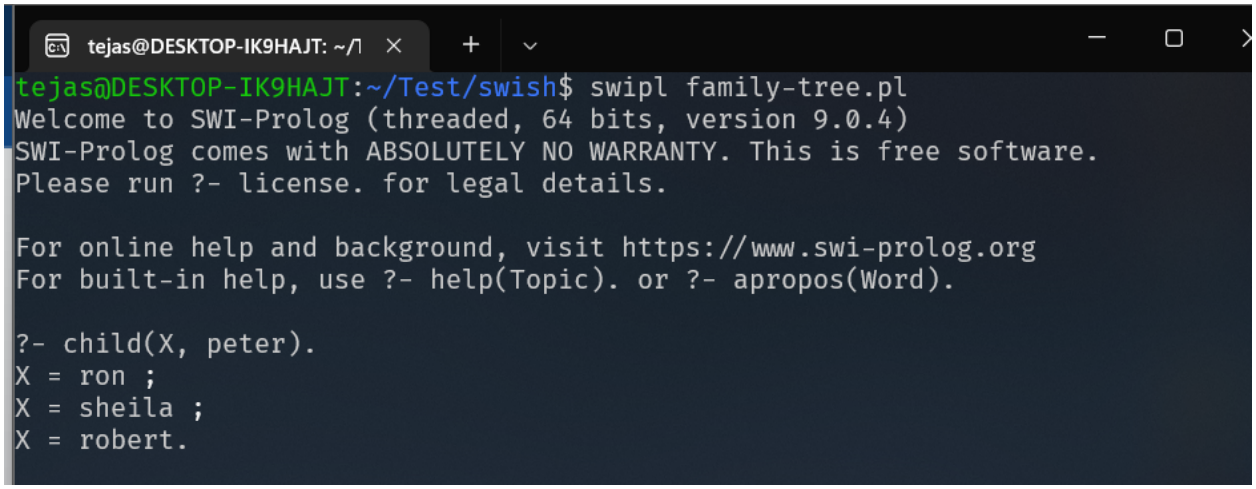
```
parent(peter, ron).
parent(jessica, ron).
parent(peter, sheila).
parent(jessica, sheila).
parent(peter, robert).
parent(jessica, robert).
parent(david, cynthia).
parent(sheila, cynthia).
parent(david, steven).
parent(sheila, steven).
parent(robert, francisco).
parent(eva, francisco).
parent(donald, mark).
parent(cynthia, mark).
parent(donald, lula).
parent(cynthia, lula).
parent(steven, perdo).
parent(sandra, perdo).
parent(francisco, zula).
parent(amanda, zula).
parent(francisco, cleveland).
parent(amanda, cleveland).

:- discontiguous male/1.
:- discontiguous female/1.
male(peter).
female(jessica).
male(ron).
male(david).
female(sheila).
male(robert).
female(eva).
male(donald).
female(cynthia).
male(steven).
female(sandra).
male(francisco).
```


tejas@DESKTOP-IK9HAJT: ~/
male(francisco).
female(amanda).
male(mark).
female(lula).
female(perdo).
female(zula).
male(cleveland).
mother(X, Y):-
parent(X, Y),
female(X).
father(X, Y):-
parent(X, Y),
male(X).
grand_father(X, Y):-
father(X, Z),
parent(Z, Y).
grand_mother(X, Y):-
mother(X, Z),
parent(Z, Y).
child(X, Y):-
parent(Y, X).
son(X, Y):-
child(X, Y),
male(X).
daughter(X, Y):-
child(X, Y),
female(X).
sibling(X, Y):-
NORMAL
family-tree.pl
44%


tejas@DESKTOP-IK9HAJT: ~/
73
74 sibling(X, Y):-
75 parent(Z, X),
76 parent(Z, Y),
77 diff(X, Y).
78
79 sister(X, Y):-
80 sibling(X, Y),
81 female(X).
82
83 brother(X, Y):-
84 sibling(X, Y),
85 male(X).
86
87 ancestor(X, Y):- parent(X, Y).
88 ancestor(X, Y):- parent(X, A), ancestor(A, Y).
89
90
NORMAL
family-tree.pl
87%

Output:

1) Find all children of Peter



2) Find sister of ron

```
?- sister(X, ron).
X = sheila .
```

3) Parent of zula

```
?- parent(X, zula).
X = francisco ;
X = amanda.
```

4) Grandfather of zula

```
?- grand_father(X, zula).
X = robert ;
```

5) Ancestor of zula

```
?- ancestor(X, zula).
X = francisco ;
X = amanda ;
X = peter ;
X = jessica ;
X = robert ;
X = eva ;
```

6) Female ancestor of cleveland

```
?- ancestor(X, cleveland), female(X).
X = amanda ;
X = jessica ;
X = eva ;
```

7) Cleveland's mother

```
?- mother(X, cleveland).
X = amanda.
```

8) Sheila's grandkids

```
?- grand_mother(sheila, X).
X = mark ;
X = lula ;
X = perdo.
```

9) Perdo's father's sister.

```
?- father(X, perdo), sister(Y, X).
X = steven,
Y = cynthia .
```

10) Ron's father's grandchild's daughters.

```
?- father(X, ron), grand_father(X, Y), daughter(Z, Y).
X = peter,
Y = cynthia,
Z = lula ;
X = peter,
Y = steven,
Z = perdo ;
X = peter,
Y = francisco,
Z = zula ;
```

Conclusion:

After performing this experiment, I have learnt how to use SWISH prolog to intelligently establish relationships and query data based on it. Swish prolog can answer even the most complex logical questions.