Name: Selas Moro **Roll no.:** 39 **PID:** 182074 **Class:** BE CMPN B

ST.FRANCIS INSTITUTE OF TECHNOLOGY

Mount Poinsur, S.V.P. Road, Borivli (West), Mumbai - 400103

Computer Engineering Department

Academic Year: 2021-2022 **Class/Branch**: BE CMPN

Subject: CSC 703 Artificial Intelligence & Soft Computing **Semester**: VII

Experiment No. 1: Introduction to Prolog

Aim: To study programming language of Prolog

Theory:

Prolog is a logical and declarative programming language. The name itself, Prolog, is short for PROgramming in LOGic. Prolog is a logic language that is particularly suited to programs that involve symbolic or non-numeric computation. It is a frequently used language in Artificial Intelligence where manipulation of symbols and inference about them is a common task.

Introduction to Prolog

Prolog program specifies relationships among objects and properties of objects. The relationships can be rules. Rules can find relations which are not explicitly defined. A rule is written as follows:

A and B are brothers if
- A and B are both male
- They have same father
- They have same mother
- A is not the same as B

In prolog, program consists of a series of rules and facts. A program is run by presenting some query and seeing if this can be proved against these known rules and facts.

Facts:

Facts describe explicit relationships between objects and the properties that objects might have.
e.g. Hair is black

Rules:

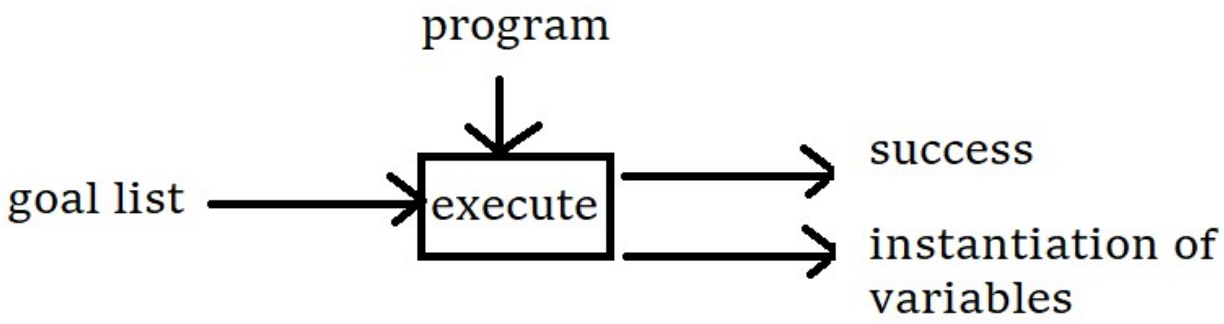
Rules define implicit relationships between objects (e.g. brother relationship) and/or rules defining implicit object properties (A is a child of B if B is parent of A).

Queries:

One then asks queries to the system like
- Is hair black?
- Is B a parent of A?

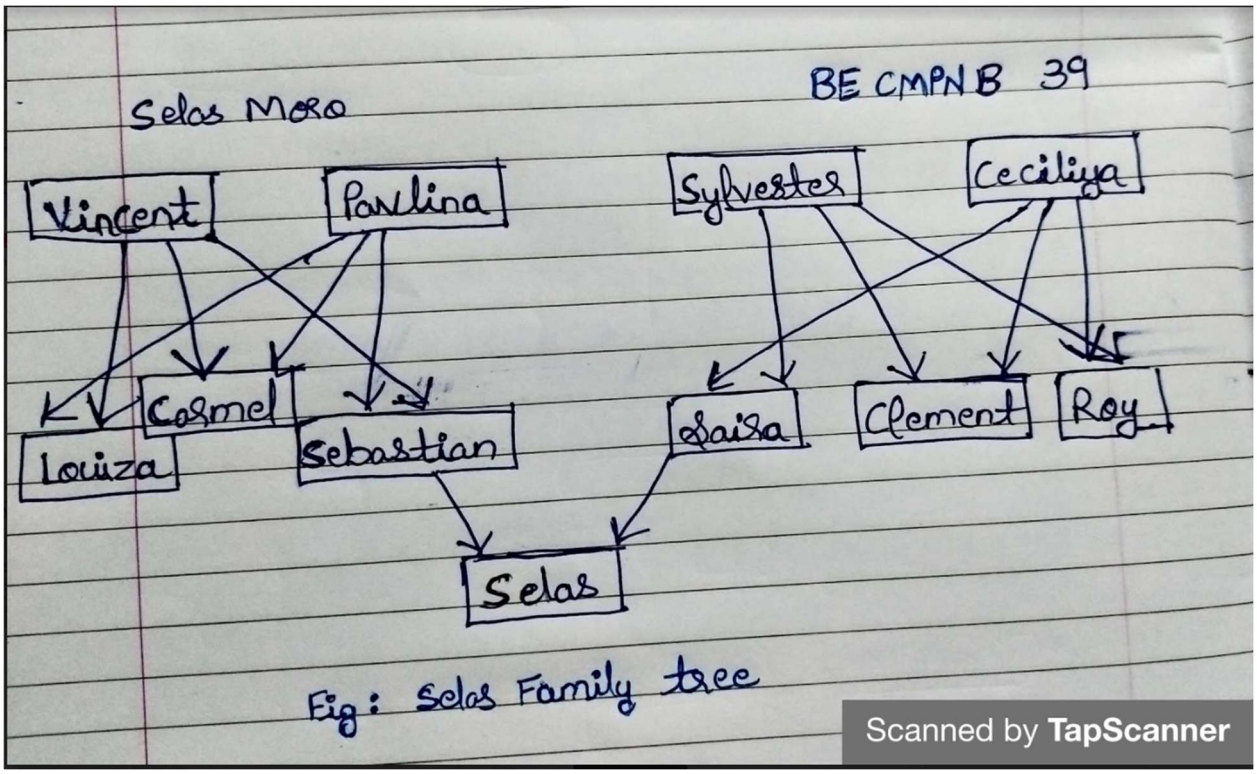
Syntax of a clause or rule

":-" This means if . it is also called the neck symbol. Left side of the neck is called head. The right hand side of the body is called the body. comma stands for and and colon stands for or
A procedure is the set of clauses or rules for the same relation.



Experiment:

1.Create the family tree of yours and write a program in prolog to represent family tree.



2. Run the code and show output of all possible queries related to family tree.

Code:

```
%Facts
```

```
 %Female

female(saira).
female(louiza).
female(carmel).
female(pavlina).
female(ceciliya).

%Male
male(selas).
male(sebastian).
male(vincent).
male(clement).
male(roy).
male(sylvester).

%Parent
parent(selas,saira).
parent(selas,sebastian).
parent(sebastian,vincent).
parent(sebastian,pavlina).
parent(saira,sylvester).
parent(saira,ceciliya).
parent(roy,sylvester).
parent(roy,ceciliya).
parent(clement,sylvester).
parent(clement,ceciliya).
parent(carmel,pavlina).
parent(carmel,vincent).
parent(louiza,pavlina).
parent(louiza,vincent).


%RULES

mother(X,Y):- parent(X,Y),female(Y).
father(X,Y):-parent(X,Y),male(Y).
sister(X,Y):-parent(X,Z),parent(Y,Z),female(X),X\==Y.
brother(X,Y):-parent(X,Z),parent(Y,Z),male(X),X\==Y.
grandparent(X,Y):-parent(X,Z),parent(Z,Y).
grandmother(X,Z):-mother(X,Y),parent(Y,Z).
grandmother(X,Z):-father(X,Y),parent(Y,Z).
grandfather(X,Z):-father(X,Y),parent(Y,Z).
grandfather(X,Z):-mother(X,Y),parent(Y,Z).
wife(X,Y):-parent(Z,X),parent(Z,Y),female(X),male(Y).
uncle(X,Z):-parent(X,Y),brother(Y,Z).
```

```
aunt(X,Z):-parent(X,Y),sister(Y,Z).
```

Output:

GNU Prolog 1.5.0 (64 bits)

Compiled Jul 8 2021, 12:33:56 with cl

| ?- change_directory('C:/Users/Administrator.MAHESHC/Documents/prolog').

yes

| ?- [exp1tree].

compiling C:/Users/Administrator.MAHESHC/Documents/prolog/exp1tree.pl for byte code...

C:/Users/Administrator.MAHESHC/Documents/prolog/exp1tree.pl compiled, 45 lines read - 6434 bytes written, 21 ms

Yes

| ?- **sister(X,Y).**

X = saira

Y = roy ? ;

X = saira

Y = clement ? ;

X = saira

Y = roy ? ;

X = saira

Y = clement ? ;

X = carmel

Y = sebastian ? ;

X = carmel

Y = louiza ? ;

X = carmel

Y = sebastian ? ;

X = carmel

Y = louiza ? ;

X = louiza

Y = sebastian ? ;

X = louiza

Y = carmel ? ;

X = louiza

Y = sebastian ? ;

X = louiza

Y = carmel ? ;

(94 ms) no

| ?- **brother(X,Y).**

X = sebastian

Y = carmel ? ;

X = sebastian

Y = louiza ? ;

X = sebastian

Y = carmel ? ;

X = sebastian

Y = louiza ? ;

X = roy

Y = saira ? ;

X = roy

Y = clement ? ;

X = roy

Y = saira ? ;

X = roy

Y = clement ? ;

X = clement

Y = saira ? ;

X = clement

Y = roy ? ;

X = clement

Y = saira ? ;

X = clement

Y = roy ? ;

(63 ms) no

| ?- **uncle**(X,Y).

X = selas

Y = carmel ? ;

X = selas

Y = louiza ? ;

X = selas

Y = carmel ? ;

X = selas

Y = louiza ? ;

(31 ms) no

| ?- **aunt(X,Y).**

X = selas

Y = roy ? ;

X = selas

Y = clement ? ;

X = selas

Y = roy ? ;

X = selas

Y = clement ? ;

(63 ms) no

Mother and Father Queries:

| ?- mother(selas,saira).

true ?

(16 ms) yes

| ?- mother(saira,ceciliya).

yes

| ?- mother(sebastian,pavlina).

yes

| ?- mother(sebastian,pavlina).

yes

| ?- father(selas,sebastian).

yes

| ?- father(sebastian,vincent).

true ?

yes

| ?- father(saira,sylvester).

true ?

yes

Grandparent Queries:

yes

| ?- grandmother(selas,pavlina).

yes

| ?- grandmother(selas,ceciliya).

true ?

yes

| ?- grandfather(selas,vincent).

true ?

yes

| ?- grandfather(selas,sylvester).

true ?

Wife Queries:

yes

| ?- wife(saira,sebastian).

true ?

yes

| ?- wife(pavlina,vincent).

true ?

yes

| ?- wife(ceciliya,sylvester).

true ?

yes

Conclusion: In this experiment, we have studied on programming language Prolog and implemented our family tree in Prolog.