Exercise: Simulate the working of inference in a rule based engine

**Tool**: SWI-Prolog

**Sample Problem**:

Given information about a Family Tree, implement a system to generate inferences for queries based on Facts and Rules about the family.

**Input Type**: Queries

parent\_of(kamladevi,arun).

parent\_of(kamladevi,abha).

child(arun,kamladevi).

grandparent(kamladevi,abha).

sibling(rakesh,leena).

brother(rakesh,leena).

cousin(abha,gautam).

**Logic**: Resolution, Unification, Backward Chaining

**Output Type**: True or False based on the execution result of the query

**Implementation**

male(karamSingh).

male(arun).

male(rakesh).

male(gautam).

female(kamladevi).

female(abha).

female(anu).

female(saroj).

female(leena).

parent\_of(karamSingh,arun).

parent\_of(karamSingh,leena).

parent\_of(karamSingh,rakesh).

parent\_of(kamladevi,arun).

parent\_of(kamladevi,rakesh).

parent\_of(kamladevi, leena).

parent\_of(arun,abha).

parent\_of(saroj,abha).

parent\_of(rakesh,gautam).

parent\_of(leena,anu).

/\* Rules \*/

parents(X,Y):- parent\_of(X,Y).

child(X, Y) :-

parent\_of(Y, X).

grandparent(X, Y) :-

parent\_of(X, Z),

parent\_of(Z, Y).

sibling(X, Y) :-

parent\_of(Z, X),

parent\_of(Z, Y),

X \= Y.

sister(X, Y) :-

sibling(X, Y),

female(X),

X \= Y.

brother(X, Y) :-

sibling(X, Y),

male(X),

X \= Y.

uncle(X, Y) :-

brother(X, Z),

child(Y, Z).

aunt(X, Y) :-

sister(X, Z),

child(Y, Z).

cousin(X, Y) :-

grandparent(Z, X),

grandparent(Z, Y),

\+sibling(X, Y),

X \= Y.

**Output**:

?- parent\_of(kamladevi,arun).

true.

?- parent\_of(kamladevi,abha).

false.

?- child(arun,kamladevi).

true.

?- grandparent(kamladevi,abha).

true .

?- sibling(rakesh,leena).

true .

?- sister(akesh,leena).

false.

?- sister(rakesh,leena).

false.

?- sister(leena,rakesh).

true .

?- brother(rakesh,leena).

true .

?- cousin(abha,gautam).

true .

**Screenshot**



**Lab Exercises**

**Implement the following rules:**

1. Father
2. Mother
3. Son
4. Daughter
5. Ancestor