Lab 1: Introduction to Prolog and Basic Syntax

Objective: The objective of this lab is to understand the basic syntax and structure of Prolog. The lab will cover defining facts, rules, and running queries to perform logical inferences.

Theory

Prolog Basics:

Prolog (Programming in Logic) is a declarative programming language, where the logic of the computation is expressed in terms of relations, represented as facts and rules. A Prolog program consists of a series of facts and rules that describe relationships and logical connections between data. Queries are used to retrieve information or infer new data based on the given facts and rules.

Key Concepts:

1. **Facts:** Represent statements that are unconditionally true.
   - Example: `parent(john, mary).` Meaning that John is the parent of Mary.
2. **Rules:** Represent conditional statements that define relations between facts.
   - Example: `grandparent(X, Y) :- parent(X, Z), parent(Z, Y).` States that X is a grandparent of Y if X is a parent of Z and Z is a parent of Y.
3. **Queries:** Questions that Prolog answers based on the facts and rules.
   - Example: `?- parent(john, mary).` asks whether John is the parent of Mary.
4. **Variables:** Represent unknown values in Prolog, starting with an uppercase letter.
   - Example: `X` in `parent(X, mary)` is a variable.

Facts

A little Prolog program consisting of four facts:

```
likes(john,prolog).

likes(john,football).

likes(john,mary).

likes(mary,football).
```

Queries After compilation we can query the Prolog system:

```
?- likes(john,prolog).

true

?- likes(mary,john).

false
```

For query “is there something that both john and mary likes”

?-likes(john,X),likes(mary,X).

X=football (It uses the backtracking to solve the above problem)

The order of fact and queries is important.

Prolog would solve the above query faster if it is adjusted slightly to:

?- likes(mary,X),likes(john,X).

X=football

This is succeed without any backtracking

Conclusion:

Lab 2: Family Tree Problem in Prolog

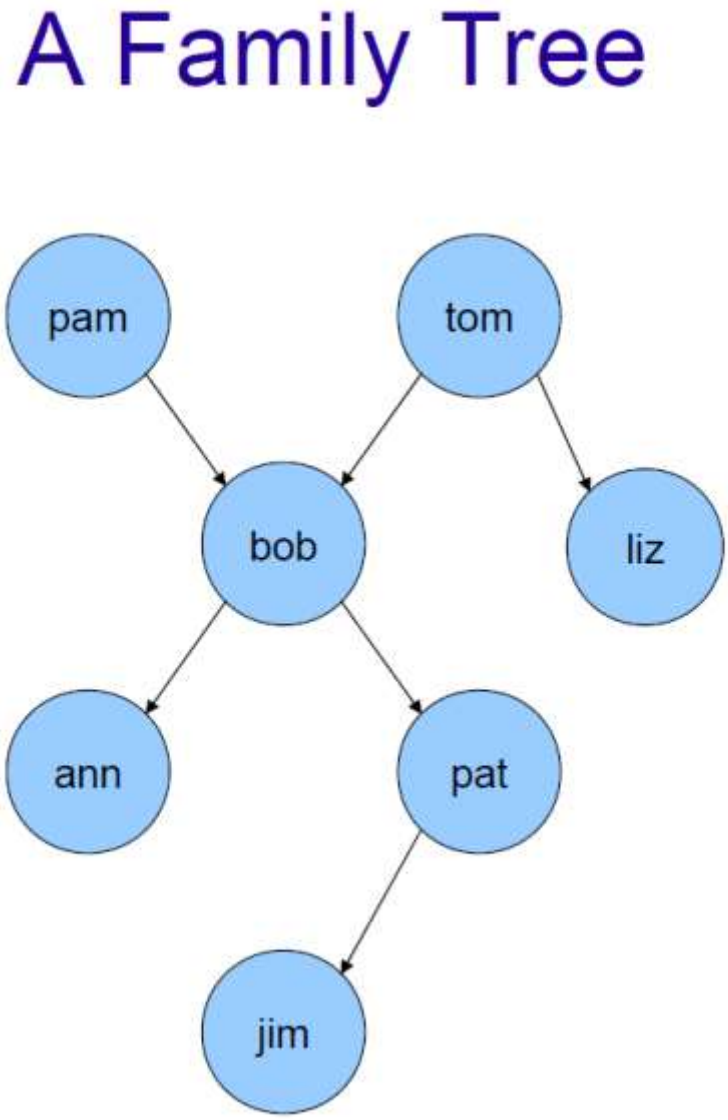
Objective: The objective of this lab is to model a family tree using Prolog, define family relationships through facts and rules, and query relationships.

Key Concepts:

1. **Facts:** Direct relationships in the family (e.g., parent-child).
   - Example: `parent(john, mary).`
2. **Rules:** Logical inferences based on facts.
   - Example: `grandparent(X, Y) :- parent(X, Z), parent(Z, Y).` (X is Y's grandparent if X is the parent of Z and Z is the parent of Y).
3. **Queries:** Used to ask about relationships

Code Implementation

Defining the Family Tree

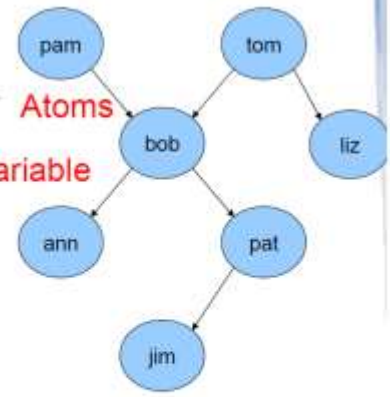


Insert the following clause:

```
parent(pam, bob).
parent(tom, bob).
parent(tom, liz).
parent(bob, ann).
parent(bob, pat).
parent(pat, jim).
```

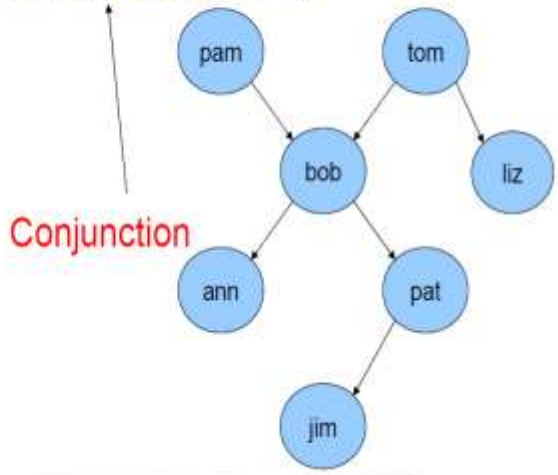
A Family Tree

- ?- parent(bob, pat).
  - yes
- ?- parent(liz, pat).
  - no
- ?- parent(X, liz).
  - X = tom
- ?- parent(bob, X)
  - X = ann;
  - X = pat;
  - no



Who is a grandparent of jim?

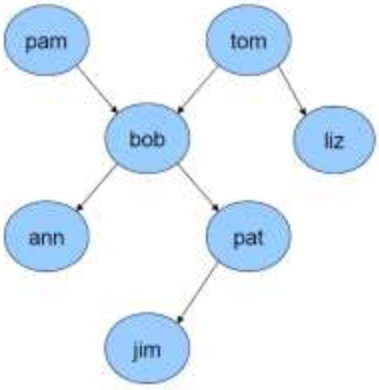
- ?- parent(Y, jim), parent(X, Y).
- X = bob
- Y = pat



Conjunction

Now Add new fact:

female(pam).
male(tom).
male(bob).
female(liz).
female(pat).
female(ann).
male(jim).



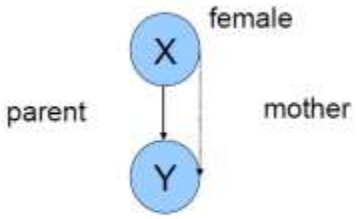
- Offspring
  - For all X and Y,
    - Y is an offspring of X if
    - X is a parent of Y.
- Prolog Rule:
  - offspring(Y, X) :- parent(X, Y).

conclusion condition

Add the above rule.

Now You can query.

- Mother
  - mother(X, Y) :- parent(X, Y), female(X).



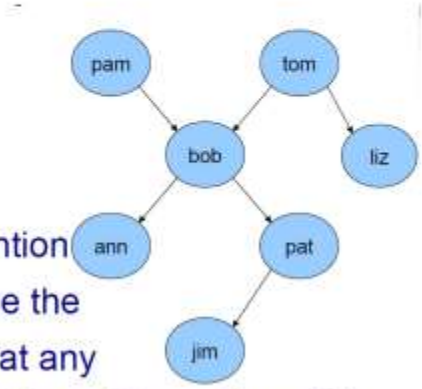
Then you can query as:

?-mother(A,B).

Or

?-mother(A,jim).

- sister(X, Y) :-
  parent(Z, X),
  parent(Z, Y),
  female(X).



- sister(X, pat).
  - X = ann;
  - X = pat
- Our rule does not mention that X and Y must not be the same and so will find that any female who has a parent is a sister of herself!

Conclusion: