Lab - 1

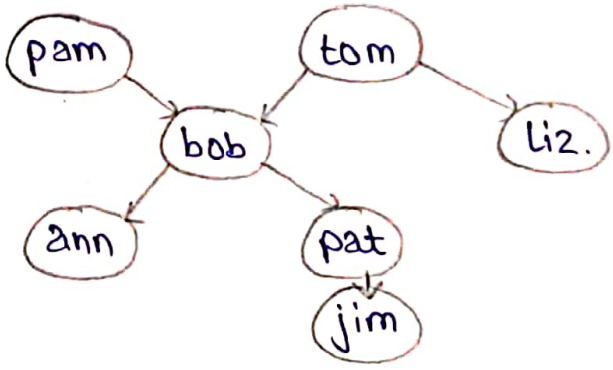
Date:

What is Prolog?

→ Prolog is a logic programming language which is associated with artificial intelligence and computational linguistics.

Prolog has its roots in first order logic, a formal logic, and unlike many other programming languages, Prolog is intended primarily as a declarative programming language: the program logic is expressed in terms of relations, represented as facts and rules. A computation is initiated by running a query over these relations.

Example: Family tree diagram.



The above figure represents a family tree diagram which will be used to develop clauses on which querying will be done.

Clauses:- A clause in Prolog is a unit of information ending with a full stop. A clause may be a fact or it may also be a query to the prolog interpreter.

The relationships developed for the family tree above are as follows:-

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

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

Queries: The prolog interpreter responds to queries about the facts and rules represented in its database. The database is assumed to represent what is true about a particular domain. While making a query we are asking Prolog whether it can prove that our query is true or not. We can also make variable bindings in our query which is represented by a capital letter.

Example:-
```
?- parent(X, jim).
X= pat.
```

Here, X is a variable which gives the parent of jim if present in database else it returns false.

Some of the queries made using the family tree clauses, relationships are:-

①
```
?- parent(pam, bob).
true.
```

②
```
?- parent(bob, pam).
false.
```

③ ?-parent(x, bob).

X= pam ;

X= tom

④ ?- parent(x, jim).

X= pat .

⑤ ?-parent (x, liz).

X= tom

⑥ ?- parent (tom, x)

X= bob ;

X= liz

⑦ ?- parent (bob, x)

X= ann ;

X= pat

⑧ ?- parent(X, Y).

X= pam,

Y= bob ;

X= tom,

Y= bob;

X= tom,

Y= liz;

X= bob

Y= ann;

X= bob,

Y= pat;

X= pat,

Y= jim.

⑨ ?parent(Y, jim), parent(X, Y).

Y= pat,

X= bob.

⑩. ?- parent(Y, jim), parent(X, Y), parent(Z, X).

Y= pat,

X= bob,

Z= pam;

Y= pat,

X= bob,

Z= tom.

(11) ?- female(pam).

true.

(12) ?- female(x).

x = pam;

x = liz;

x = pat;

x = ann.

(13) ?- male(x).

x = tom;

x = bob;

x = jim.

(14) ?- male(bob).

true.

(15) ?- parent(x, jim), female(x).

x = pat.

(16). ?- parent(x, bob), female(x).

x = pam;

false.