| EXP.NO : 12(b)  DATE :11/05/2024. PROLOG INTRODUCTION |
| --- |

PROLOG:

To learn PROLOG terminologies and write basic programs.

TERMINOLOGIES

1.Atomic Terms:-

Atomic terms are usually strings made up of lower- and upper case letters, digits, and the underscore, starting with a lowercase letter.

Ex:

dog ab\_c\_321

2.Variables:-

Variables are string , letters, digits, and the underscore, starting with a capital letter or an underscore.

Ex:

Dog Apple\_420

3.CompoundTerms:-

Compound terms are made up of a PROLOG atom and a number of arguments (PROLOG terms,

i.e., atoms, numbers, variables, or other compound terms) enclosed in parentheses and separated by commas.

Ex:

is\_bigger(elephant,X) f(g(X,\_),7)

4.Facts:-

A fact is a predicate followed by a dot.

Ex:

bigger\_animal(whale).

life\_is\_beautiful.

5.Rules:-

A rule consists of a head (apredicate) and a body(asequence of predicates separated by commas).

Ex:

is\_smaller(X,Y):-is\_bigger(Y,X).

aunt(Aunt,Child):-sister(Aunt,Parent),parent(Parent,Child).

SOURCECODE:

KB1:

woman(mia).

woman(jody).

woman(yolanda).

playsAirGuitar(jody).

party.

Query1:?-woman(mia).

Query2:?-playsAirGuitar(mia).

Query3:?-party.

Query4:?-concert

OUTPUT:

KB2:

happy(yolanda).

listens2music(mia).

Listens2music(yolanda):-happy(yolanda).

playsAirGuitar(mia):-listens2music(mia).

playsAirGuitar(Yolanda):-listens2music(yolanda).

OUTPUT:

KB3:

likes(dan,sally). likes(sally,dan). likes(john,brittney).

married(X,Y):- likes(X,Y), likes(Y,X).

friends(X,Y):-likes(X,Y) ;likes(Y,X).

OUT

KB4:

food(burger). food(sandwich). food(pizza). lunch(sandwich). dinner(pizza). meal(X):-food(X).

OUTPUT:

KB5: Find minimum maximum of two numbers

find\_max(X,Y,X):-X>=Y,!.

find\_max(X,Y,Y):-X<Y.

find\_min(X,Y,X):-X=<Y,!.

find\_min(X,Y,Y):-X>Y.

Output:

| ?- find\_max(100,200,Max).

Max = 200

yes

| ?- find\_max(40,10,Max).

Max = 40

yes

| ?- find\_min(40,10,Min).

Min = 10

yes

| ?- find\_min(100,200,Min).

Min = 100

yes

| ?-

KB6:

Here are some simple clauses.

likes(mary,food).

likes(mary,wine).

likes(john,wine).

likes(john,mary).

The following queries yield the specified answers.

| ?- likes(mary,food).

yes.

| ?- likes(john,wine).

yes.

| ?- likes(john,food).

no.

How do you add the following facts?

1. John likes anything that Mary likes

2. John likes anyone who likes wine

3. John likes anyone who likes themselve

% Existing facts

likes(mary, food).

likes(mary, wine).

likes(john, wine).

likes(john, mary).

% New facts

likes(john, X) :- likes(mary, X). % John likes anything that Mary likes

likes(john, Y) :- likes(Y, wine). % John likes anyone who likes wine

likes(john, Z) :- likes(Z, Z). % John likes anyone who likes themselves

% Queries and their answers

% Query: likes(mary, food).

% Answer: yes.

% Explanation: Mary likes food (existing fact).

% Query: likes(john, wine).

% Answer: yes.

% Explanation: John likes wine (existing fact).

% Query: likes(john, food).

% Answer: no.

% Explanation: John does not like food (not explicitly defined).

% Existing facts

likes(mary, food).

likes(mary, wine).

likes(john, wine).

likes(john, mary).

% New facts and rules

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

% John likes anything that Mary likes.

% This rule means that if Mary likes something X, then John also likes X.

Likes(john, Y) :- likes(Y, wine).

% John likes anyone who likes wine.

% This rule means that if someone Y likes wine, then John also likes Y.

likes(john, Z) :- likes(Z, Z).

% John likes anyone who likes themselves.

% This rule means that if someone Z likes themselves, then John also likes Z.

% Queries and their answers

% Query: likes(mary, food).

% Answer: yes.

% Explanation: Mary likes food (existing fact).

% Query: likes(john, wine).

% Answer: yes.

% Explanation: John likes wine (existing fact).

% Query: likes(john, food).

% Answer: no.

% Explanation: John does not like food (not explicitly defined).

RESULT:

Thus to implement the PROLOG has been executed successfully and verified