K.G.C.E.
Karjat - Raigad

Page No. :
Date :

Prolog Programming Assignment

Name : Rahul Patil

Roll no : 47

Class : BE - IT.

Subject : AI LAB.

Q1) How does the queries in kb.pl file are executed ?

→ Prolog code for kb.pl :-

```
% Knowledge bases

loves (vincent , mia).
loves (marcellus , mia).
loves (pumpkin , honey_bunny)
loves (honey_bunny , pumpkin).

jealous (X , Y) :-
        loves (X, Z)
        loves (Y, Z)
```

A] Query 1 : ?- loves (X, mia).

Output 1 :- X = vincent
X = marcellus.

Explaination : In above code, the "loves (X, mia)." query initializes a variable 'X'. The query passed two parameters, i.e, 'X' and 'mia' (fixed parameter). The query will return the values of 'X' by checking the value associated by parameter 'Mia'. Therefore, it returns, i) X = vincent and ii) X = marcellus as these values are assiociated ~~by~~ with 'mia'.

B] Query 2 :- ?- jealous(X,Y).

Output 2 :- X = Y , Y = Vincent
X = Vincent
Y = marcellus
X = marcellus
Y = vincent
X = Y , Y = marcellus.
X = Y , Y = pumpkin.
X = Y , Y = honey-bunny.

Explaination :- As there is no fixed parameter in our query. The query will produce output of every jealous(X,Y) pair on our prolog code. The 'jealous()' rule follows :

jealous(X, Y) :- loves(X, Z), loves(Y, Z).

~~The~~ Initially, X and Y both were associated to vincent, i.e. self-association. It then follows Reflexive Property for the rest of the prolog code.

Q2) How does the queries in lists.pl file are executed?

→

Prolog code of lists.pl :-

```
suffix (Xs, Ys) :-
    append (_, Ys, Xs).

prefix (Xs, Ys) :-
    append (Ys, _, Xs).

sublist (Xs, Ys) :-
    suffix (Xs, Zs),
    prefix (Zs, Ys).

nrev ([], []).
nrev ([H|T0], L) :-
    nrev (T0, T),
    append (T, [H], L).
```

A] Query 1 :- ?- sublist ([a,b,c,d,e], [c,d]).

Output 1 :- true
false.

Explaination :- The query passes two parameters to sublist rule in our prolog. The sublist rule calls the suffix and prefix predicates. Using the predicates, the rule returns true if the second parameter is the sublist of first and false, otherwise.

B] Query 2 :- ?- suffix ([a,b,c], Zs)

Output 2 :- Zs = [a,b,c]
Zs = [b,c]
Zs = [c]
Zs = [] .
false.

Explaination :- Suffix removes the first element from the given list, and returns the remaining list. The suffix procedure continues to remove element till the list is empty. It returns when an empty list is passed.

Q.3) Programming create a Prolog code to find the factorial of a number ?.

→

```
factorial (0, 1).

factorial (N, F) :-
    N > 0
    N1 is N - 1,
    factorial (N1, F1),
    F is N * F1
```

~~Output~~ :- Query :- ?- factorial (4, W)

Output :- W = 24.

Q4) In examples data set movies.pl, write query strings and result of query execution of any of 5 testes:

→

a) In which year was the movie American Beauty released?

→

Query : ?- movie(american_beauty, Y)
Output : Y = 1999.

b) Find the movies released in year 2000.

→

Query : ?- movie(M, 2000)
Output : M = down_from_the_mountain.
M = O_brother_where_art_thou
M = ghost_world.

c) Find movies released before 2000

→

Query : ?- movie(M, Y), Y < 2000
Output : M = american_beauty
Y = 1999.
M = anna.
Y = 1987
M = barton_fink
Y = 1991

d) Find the movies released after 1990.

→

Query : ?- movie(M, Y), Y > 1990
Output : M = american_beauty
M = 1999.

d) Output : M = barton-fink.
Y = 1991.

e) Find a director of a movie in which Scarlett Johannson appeared.

→ Query : ?- actress(M, 'Scarlett Johansson'), director(M, D).

Output : D = peter_webber.
M = girl-with-a-pearl-earring.

Q5) Draw a family tree of you / any arbitary family which has the following relations mother, father, daughter, son, grandson, grandmother, sibling, uncle, person, male, female. You need to convert it into KB and write atleast 6 queries and query results on your KB.

→ Diagram :

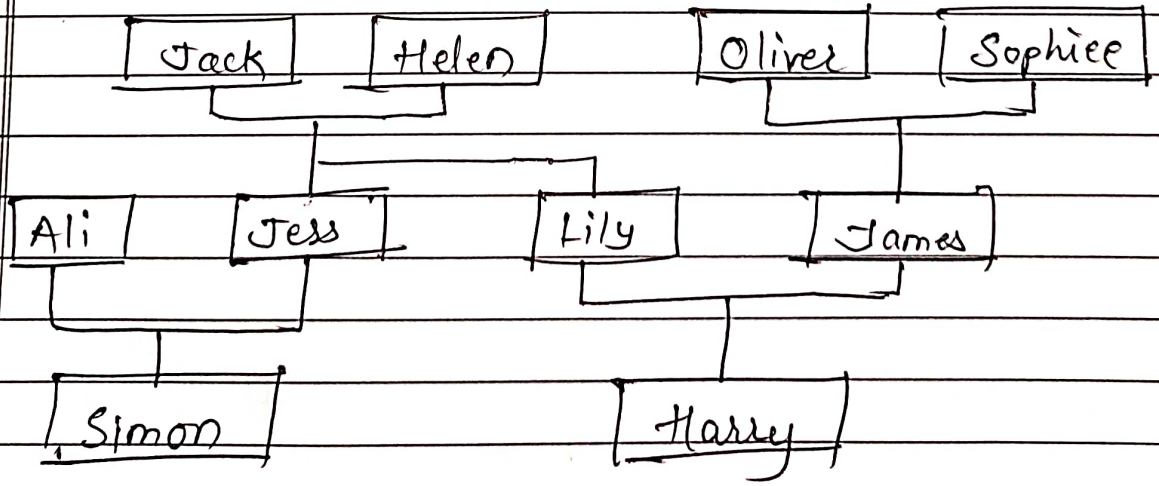

Fig : Family Tree

Query 1 : ?- mother_of(X, jess)
Output 1 : X = helen.

Query 2 : ?- parent_of(X, simon)
Output 2 : X = jess

Query 3 : ?- sister_of(X, lily)
Output 3 : X = jess

Query 4 : ?- parent_of(X, harry)
Output 4 : X = lily
X = james.

Query 5 : ?- aunt_of(X, simon)
Output 5 : X = lily

~~Query 6 : ?- grandmother~~

Query 6 : ?- grandfather_of(X, harry)
Output : X = jack.