

Prolog Programming Assignment

NAME: Vijay Kumar Prasad

Class: B.E. IT

Roll no. 49

SUBJECT: IS Lab

DOP	DOC	MARKS	SIGN
-----	-----	-------	------

Prolog Programming Assignment

Q.17 How does the queries in kb.pl file are executed?

lives (vincent, mia)

lunes (marcellus, mia).

comes (pumpkin, honey-bunny).

tones (honey-bunny, pump-kin).

jealous (x, y): -

lives (x, y),

$\text{lives}(Y, Z)$.

Query: ? - loves(x, mia)

Output: X - Vincent

X - Marcellus

Explanation: Here it is assumed that X is either Vincent or Marcellus, since Vincent loves Mia and Marcellus loves Mia as well.

Query :

jealous(x, y).

Output:

$X = Y$, $Y = \text{vincent}$

X = vincent

$\gamma = \text{marcellus}$

X = marcelles

Y = Vincent

$X = \gamma$, $Y = \text{marcellus}$

$x = Y$, $Y = \text{pumpkin}$

$x = \gamma$, $y = \text{honey, bunny}$

By referring the knowledge base, we get all possible jealous() variable pairs. This is because the query doesn't contain fixed parameters. Earlier, x and y both were associated to Vincent, and later it follows reflexive property.

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

An

suffix (x_s, y_s): -

append(-, Ys, Xs).

prefix(x_5, y_8): -

`append(Ys, - , Ys).`

`sublist(xs, ys)`:

suffix(x_s, z_s),

prefix(zs, ys)

`nnov([3, 3]).`

$\text{rev}([\text{H}][\text{T03}], \text{L}) :=$

$\text{mer}(T_0, T)$

append (T , $[H]$, L).

Query:

? - sublist([a, b, c, d, e], [c, d]).

Output:

True

Explanation: Here, `sublist()` observes `[c, d]` as a sublist of the main list `[a, b, c, d, e]`, and thus gives the output `true`.

Query:

? - suffix ($[a, b, c]$, zs).

Output: $Z_S = [a, b, c]$

$$Z_S = [b, c]$$

$$ZS = [C]$$

Zs = []

False

false

Explanation : Suffix removes the elements from the front side, thus discarding 'a' from Zs. likewise proceeding further all the elements from the list gets discarded.

q.3) Create a Prolog rule to find factorial of a number

Ans:

factorial(0,1).

factorial(N,F):-

$$\underline{N > 0}$$

N1 is N-1,

factorial(N1, F1)

$F_1 \times N^*F_1$

Query:

? - factorial (3, w)

Output

$$b) = 6$$

Q.4) In examples dataset movies.pl write query strings and results of query execution for any of 5 tasks

Ary:

* Find the movies released in the year 2000.

Query : movie (M, 2000).

Output: M = down from the mountain

M = o-brother where art thou

M-ghost-world

* Find a director of movie in which Scarlett Johansson appeared.

actress (M, scarlett-johansson, -), director (M, D).

Output:

D = peter-webber

M = girl-with-a-pearl-earring

D = sofia - coppola

$n = \text{lost-in-transaction}$

* Find the movies released before 2000.

query : movie(17, y); y < 2000.

Output: M=american beauty

Y = 1999

$$M = \text{anno}$$

Y = 1967

* Find the movies released after 1999

Query : movie(M, Y). Y > 1999.

$$\text{Output} = M = cg$$

Y= 2001

M = down - from the mountain

$y = 2000$

* Find the movies released after 1990

query : ?- movie(M, Y), Y > 1990.

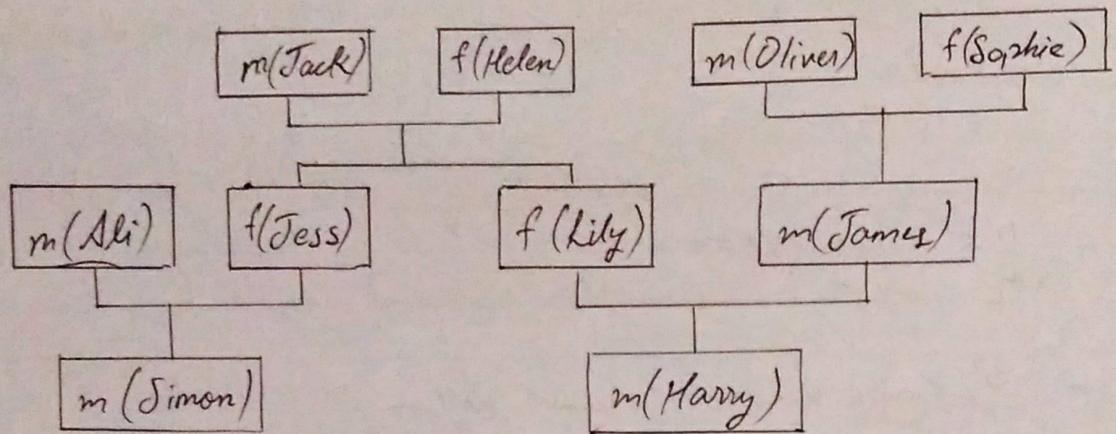
Output: M = american - beauty

γ = 1999

$$M = \text{Garton-Yink}$$

γ = 1991

Family tree



(Q5) Draw a family tree of you/any arbitrary family which has the following relations: mother, father, daughter, son, grandson, grandmother, sibling, uncle, person, male, female. You need to convert it into KB & write at least 6 queries & query results.

Ans: parent_of(jack, jess).

parent of (Jack, Sily).

parent of (heter. jers).

parent. of (helen. lily).

parent-of(oliver, james).

parent-of(sophie, james)

parent-of (jess, simon).

parent-of(ali, simon).

parent of (lily, harry).

parent of James, Harry

2 7/

1st Rules

(* Rules *)

$\text{father_of}(x, y) \equiv \text{male}(x)$.

parent. of (X, Y) .

mother-of(x, y): female(x),

parent-of(X, Y).

grandfather-of (x, y): - male (x),

parent-of(x, z),

parent of $(2, x)$

grandmother of (x, y): - female (x),

parent-of(x, z),

Parent - of (2, Y).

sister-of(x, y) :- % (x, y or y, x) %

female (X).

father-of (F, Y), father-of (F, X), $X \neq Y$.

sister-of(x, y) :- female(x),

$\text{mother_of}(M, Y)$. $\text{mother_of}(M, X)$, $X \neq Y$.

aunt-of(x, y) :- female(x),

parent-of(2, Y), sister-of(2, X), !.

brother_of(x, y) :- $\% (x, y \text{ or } y, x) \%$

male (x),

father-of(F, Y), father-of(F, X), X \neq Y.

brother-of(x, y) :- male(x),

$\text{mother_of}(M, Y)$, $\text{mother_of}(M, X)$, $X \setminus= Y$

uncle-of(x, y): -

$\text{parent-of}(z, y)$, $\text{brother-of}(z, x)$.

ancestor of(x, y) : -parent of(x, y).

ancestor-of(x, y): - parent-of(x, z).

ancestor of $(2, y)$.

Query: ? - mother-of (X_{ijrs}).

Output: $X = \text{Helen}$

Query : ? - parent-of (x, simon)

Output: $x = jess$

Query : ? - sister-of (X , lily).

Output: $X = jess$

Query : ? - parent-of (X , harry).

Output: $X = lily$

$X = james$

Query : ? - grandfather-of (X , harry).

Output: $X = jack$

Query : ? - ancestor-of (X , harry).

Output: $X = lily$

$X = james$

$X = jack$

$X = helen$

$X = oliver$

$X = sophie$