

CSL 303 : Artificial Intelligence

TUTORIAL ASSIGNMENT 7

Logical Agent - Knowledge representation

Date Assigned: 24th September, 2021

Date Submitted : 2nd October, 2021

Submitted by:

Name: S. Bhavyesh

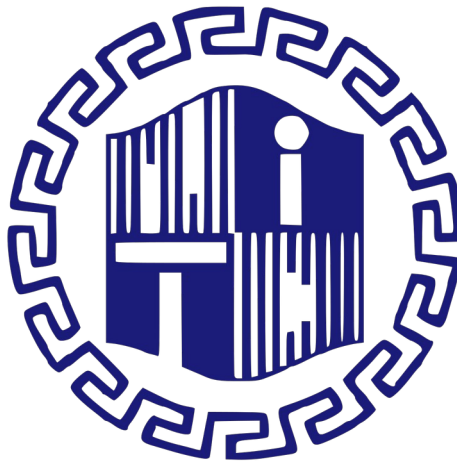
Roll Number: 191210045

Branch: Computer Science and Engineering

Semester: 5th

Submitted to: Dr. Chandra Prakash

Department of Computer Science and Engineering



NATIONAL INSTITUTE OF TECHNOLOGY DELHI

A-7, Institutional Area, near Satyawadi Raja Harish Chandra Hospital, New Delhi, Delhi 110040

1. Write prolog program to find if the given sentences is valid or not:

- If I am the Student President then I am well-known. I am the Student President. So I am wellknown.
- If I am the Student President then I am well-known. I am not the Student President. So I am not well-known.
- If Rajat is the Student President then Rajat is well-known. Rajat is the Student President. So Rajat is well known.
- If Asha is elected VP then Rajat is chosen as G-Sec and Bharati is chosen as Treasurer. Rajat is not chosen as G-Sec. Therefore Asha is not elected VP.
- If Asha is elected VP then Rajat is chosen as G-Sec and Bharati is chosen as Treasurer. Rajat is chosen as G-Sec. Therefore Asha is elected VP.
- Wherever Mary goes, so does the Lamb. Mary goes to School. So the Lamb goes to School.
- No contractors are dependable. Some engineers are contractors. Therefore some engineers are not dependable.
- Every passenger is either in first class or second class. Each passenger is in second class if and only if the passenger is not wealthy. Some passengers are wealthy. Not all passengers are wealthy. Therefore some passengers are in second class.
- All dancers are graceful. Ayesha is a student. Ayesha is a dancer. Therefore some student is graceful.

CODE:

```
1  % Part 1
2  well_known1(X):-stud_prez1(X).
3  stud_prez1(me).
4
5  % Part 2
6  well_known2(X):-stud_prez2(X).
7  stud_prez2(me):-false.
8
9  % Part 3
10 well_known3(X):-stud_prez3(X).
11 stud_prez3(rajat).
12
13 % Part 4
14 g_sec1(rajat):-vp1(asha).
15 treasurer1(bharati):-vp1(asha).
16 not_vp1(asha):-not_g_sec1(rajat).
17 not_g_sec1(rajat).
18
19 % Part 5
20 g_sec2(rajat).
21 g_sec2(rajat):-vp2(asha).
22 treasurer2(bharati):-vp2(asha).
```

```
% Part 6
goes(lamb,X):-goes(mary,X).
goes(mary,school).

% Part 7
not_dependable(X):-contractor(X).
engineer(m).
contractor(m).

% Part 8
first_or_second(X):-passenger(X).
second_class(X):-not_wealthy(X).
wealthy(a).
not_wealthy(b).

% Part 9
graceful(X):-dancer(X).
student(ayesha).
dancer(ayesha).
```

OUTPUTS:

```
bridges@bridges-CF-C2AHCCZC7:~/OS codes$ swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- ['/home/bridges/TA_7_Q1.pl'].
true.

?- well_known1(me).
true.

?- well_known2(me).
false.

?- well_known3(rajat).
true.

?- not_vp1(asha).
true.

?- vp2(asha).
ERROR: Unknown procedure: vp2/1 (DWIM could not correct goal)
?-
```

```
bridges@bridges-CF-C2AHCCZC7:~/OS codes$ swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- ['/home/bridges/TA_7_191210045_Q1_6789.pl'].
true.

?- goes(lamb,school).
true.

?- not_dependable(m).
true.

?- second_class(b).
true.

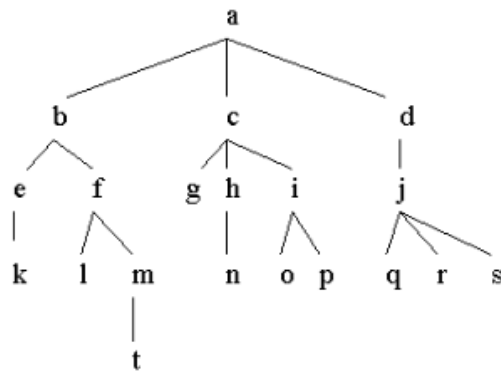
?- graceful(ayesha).
true.

?-
```

OBSERVATION/COMMENTS

1. In part 5, I was unable to find any way out of my problem, because you cannot create a PROLOG program that can test a statement's validity for multiple input combinations.
2. In parts 6, 7, 8, and 9, wherever the phrase 'some' is used, it means that if we prove the truth of the clause for even one value, we are done.
3. PROLOG is not readily usable for theorem-proving. We may have to include contrapositive statements too.

2. From the diagram below, write a Prolog definition for 'ancestor(X,Y)' with the intended meaning that "X is an ancestor of Y in the tree".



CODE:

```
1  % parent(A,B) = A is parent of B.
2
3  % children of each node.
4  parent(a,b).
5  parent(a,c).
6  parent(a,d).
7
8  parent(b,e).
9  parent(b,f).
10
11 parent(c,g).
12 parent(c,h).
13 parent(c,i).
14
15 parent(d,j).
16
17 parent(e,k).
18
19 parent(f,l).
20 parent(f,m).
21
22 parent(h,n).
23
24 parent(i,o).
25 parent(i,p).
26
27 parent(j,q).
28 parent(j,r).
29 parent(j,s).
30
31 parent(m,t).
32
33 ancestor(X,Y):- parent(X,Y).
34 ancestor(X,Y):- parent(X,Z),ancestor(Z,Y).
```

OUTPUTS:

```
bridges@bridges-CF-C2AHCCZC7:~$ swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- ['/home/bridges/TA_7_191210045_Q2.pl'].
true.

?- ancestor(a,n).
true.

?- ancestor(X,f).
X = b ;
X = a ;
false.

?-
```

OBSERVATION/COMMENTS

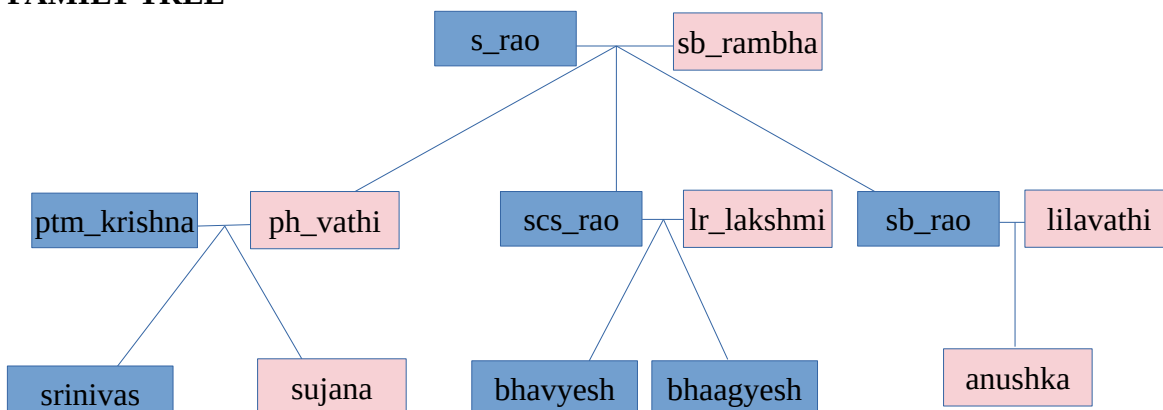
The ancestor definition has two statements, one captures the obvious parent relationship, the other is a recursive procedure. In fact the first statement is the base step, the second line is the recursion step.

3. Construct your family tree diagram (start from grandparents to your siblings). and formulate definitions for a human family tree using relations 'male', 'female', 'parent', 'father', 'mother', 'sibling', 'grandparent', 'grandmother', 'grandfather', 'cousin', 'aunt', and 'uncle'. Let 'male', 'female', 'parent' be the fundamental relations and define the others in terms of these. Write your information in facts in English.

CODE:

```
1  male(s_rao).
2  male(ptm_krishna).
3  male(srinivas).
4  male(scs_rao).
5  male(bhavyesh).
6  male(bhaagyesh).
7  male(sb_rao).
8
9  female(sb_rambha).
10 female(ph_vathi).
11 female(sujana).
12 female(lr_lakshmi).
13 female(k_lilavathi).
14 female(anushka).
15
16 parent(s_rao, ph_vathi).
17 parent(s_rao, scs_rao).
18 parent(s_rao, sb_rao).
19
20 parent(sb_rambha, ph_vathi).
21 parent(sb_rambha, scs_rao).
22 parent(sb_rambha, sb_rao).
23
24 parent(ph_vathi, sujana).
25 parent(ph_vathi, srinivas).
26 parent(ptm_krishna, sujana).
27 parent(ptm_krishna, srinivas).
28
29 parent(scs_rao, bhavyesh).
30 parent(scs_rao, bhaagyesh).
31 parent(lr_lakshmi, bhavyesh).
32 parent(lr_lakshmi, bhaagyesh).
33
34 parent(sb_rao, anushka).
35 parent(k_lilavathi, anushka).
36
37 father(X,Y):- parent(X,Y), male(X).
38 mother(X,Y):- parent(X,Y), female(X).
39 sibling(X,Y):- father(Z,X), father(Z,Y), mother(W,X), mother(W,Y), X\=Y.
40 grandparent(X,Y):- parent(X,Z), parent(Z,Y).
41 grandfather(X,Y):- father(X,Z), (father(Z,Y) ; mother(Z,Y)).
42 grandmother(X,Y):- mother(X,Z), (father(Z,Y) ; mother(Z,Y)).
43 cousin(X,Y):- grandparent(Z,X),grandparent(Z,Y), not(sibling(X,Y)), X\=Y.
44 uncle(X,Y):-male(X), parent(Z,Y), sibling(Z,X).
45 aunt(X,Y):-female(X), parent(Z,Y), sibling(Z,X).
```

FAMILY TREE



OUTPUTS:

```
bridges@bridges-CF-C2AHCCZC7:~$ swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- ['/home/bridges/TA_7_191210045_Q3.pl'].
true.

?- cousin(bhavyesh,anushka).
true.

?- trace.
true.

[trace] ?- grandparent(s_rao,bhavyesh).
  Call: (10) grandparent(s_rao, bhavyesh) ? creep
  Call: (11) parent(s_rao, _10392) ? creep
  Exit: (11) parent(s_rao, ph_vathi) ? creep
  Call: (11) parent(ph_vathi, bhavyesh) ? creep
  Fail: (11) parent(ph_vathi, bhavyesh) ? creep
  Redo: (11) parent(s_rao, _10568) ? creep
  Exit: (11) parent(s_rao, scs_rao) ? creep
  Call: (11) parent(scs_rao, bhavyesh) ? creep
  Exit: (11) parent(scs_rao, bhavyesh) ? creep
  Exit: (10) grandparent(s_rao, bhavyesh) ? creep
true.

[trace] ?- 
```

OBSERVATION/COMMENTS

1. I did not write the facts in English. Instead, I have drawn a family tree for easy visualization.
2. The trace() command is useful if one wants to know about the internal workings of PROLOG.

4. Consider the following facts/statements.

The law says that it is a crime for an American to sell weapons to hostile nations. The country Nono, an enemy of America, has some missiles, and all of its missiles were sold to it by Colonel West, who is American.

Formulate this knowledge in First Order Logic. And use prolog program to execute following queries:

a)Query : criminal(west)?

b)Query: criminal(X)?

Draw a resolution tree to find the answer of par (a)

CODE:

```
1  % The law says that it is a crime for an American to sell weapons to hostile nations.
2  criminal(X):-american(X),weapon(Y),sells(X,Y,Z),hostile(Z).
3  weapon(X):-missile(X).
4  hostile(X):-enemy(X,america).
5
6  % The country Nono, an enemy of America, has some missiles,
7  enemy(nono,america).
8  owns(nono,m).
9  missile(m).
10
11 % and all of its missiles were sold to it by Colonel West, who is American.
12 sells(west,X,nono):-missile(X),owns(nono,X).
13 american(west).
```

OUTPUTS:

```
bridges@bridges-CF-C2AHCCZC7:~$ swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.2.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- ['/home/bridges/TA_7_191210045_Q4.pl'].
true.

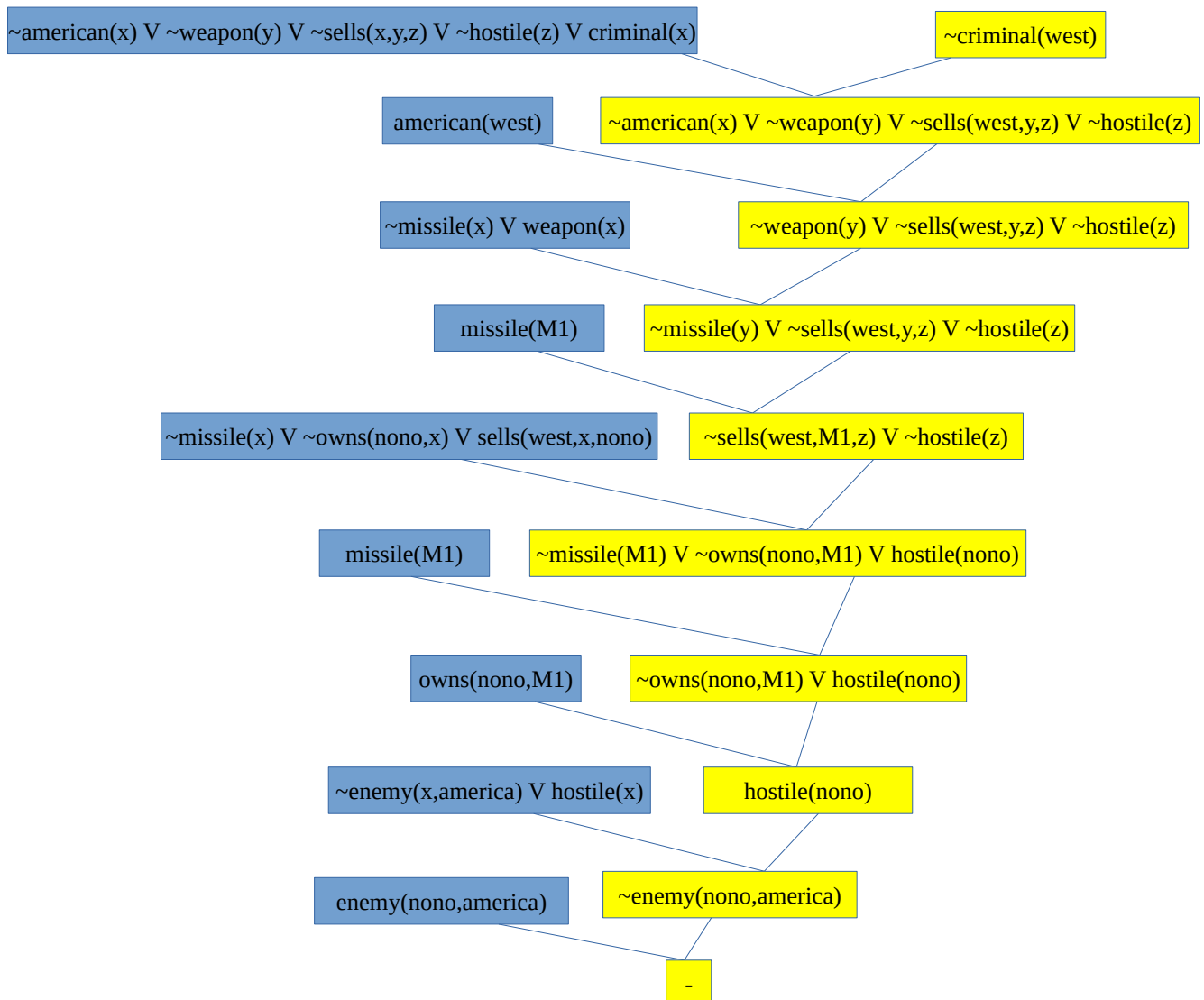
?- criminal(west).
true.

?- criminal(X).
X = west.

?- []
```

OBSERVATION/COMMENTS

The resolution tree for the given problem is as follows:



In this question, we can see the true utility of PROLOG, as a resolution prover, and the fact that almost all clauses in question are Horn clauses means that we have to simply code the statements directly as they are given in the question.