CSB 310: Artificial Intelligence

Lab7: Logical Agent - Knowledge Representation

Submitted By:

Name: **UMANG KUMAR**

Roll No: **201210051**

Branch: **CSE**

Semester: **5th Sem**

Submitted To: ***Dr. Chandra Prakash***

Release Date: 24/10/2022 **Submitted Date:** 06/11/2022

NATIONAL INSTITUTE OF TECHNOLOGY DELHI

Department of Computer Science and Engineering

2022

PART A: Exposition Problems

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 well know

Code:

```
well_known(X) :- student_president(X).
student_president(umang).
```

Output:

```
?- well_known(umang)
```

true

```
?- well_known(mohit)
```

false

- If I am the Student President then I am well-known. I am not the Student President. So I am not well-known.

Code:

```
well_known(X) :- student_president(X).
student_president(umang) :- false.
```

Output:

```
?- well_known(umang).
```

false

- If Rajat is the Student President then Rajat is well-known. Rajat is the Student President. So Rajat is well known.

Code:

```
well_known(X) :- student_president(X).
student_president(rajat) :- true.
```

Output:

```
?- well_known(rajat).
true
```

- 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.

  Code:

```
1 treasurer(X).
2 elected_as_VP(asha) :- elected_as_gsec(rajat), treasurer(bharti).
3 elected_as_gsec(rajat) :- false.
```

  Output:

```
?- elected_as_VP(asha).
Singleton variables: [X]
false
```

- 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.

  Code:

```
1 elected_as_gsec(rajat) :- true.
2 treasurer(X).
3 elected_as_VP(asha) :- elected_as_gsec(rajat), treasurer(bharti).
4
```

  Output:

```
?- elected_as_VP(asha).
Singleton variables: [X]
true
```

- Wherever Mary goes, so does the Lamb. Mary goes to School. So the Lamb goes to School.

  Code:

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

Output:

```
?- goes(lamb, school).
true
```

- No contractors are dependable. Some engineers are contractors. Therefore some engineers are not dependable.

  Code:

```
dependable(X) :- contractor(X, false).
engineer(umang, true).
engineet(mohit, true).
contractor(umang, true).
contractor(mohit, false).
```

  Output:

```
?- dependable(umang).
false
```

```
?- dependable(mohit).
true
```

- 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.

  Code:

```
passenger(aditya).
passenger(mohit).
passenger(umang).
secondclass(X) :- wealthy(X), passenger(X).
wealthy(umang).
```

  Output:

```
?- secondclass(umang).
```

true

- All dancers are graceful. Ayesha is a student. Ayesha is a dancer. Therefore some student is graceful.

  Code:

```
graceful(X):-dancer(X),student(X).
dancer(ayesha).
student(ayesha).
```

  Output:

```
?- graceful(ayesha).
true
```

Explanation/Observation:

We defined several facts in prolog using predicate logic. When a statement cannot be represent using propositional gates then we use predicates to represent the meaning of a statement and test if tautology for different cases. Predicate is a combination of facts and rules used in prolog language. In this assignment, I wrote prolog statements for different statements and test their universality.

Prolog doesn't allow most facts or conclusions having existential quantification--that is, statements that there exists some value of a variable, though we don't know what, such that a predicate expression containing it is true. Prolog doesn't allow "not" (negative) facts or conclusions--that is, direct statements that something is false.

PART B: Conceptual Questions

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:

```
parent(t,m).
parent(k,e).
parent(l,f).
parent(m,f).
parent(n,h).
parent(o,i).
parent(p,i).
parent(q,j).
parent(r,j).
parent(s,j).
parent(e,b).
parent(f,b).
parent(g,c).
parent(h,c).
parent(i,c).
parent(j,d).
parent(b,a).
parent(c,a).
parent(d,a).

ancestor(X,Y):-parent(X,Y).
ancestor(X,Y):- X == Y.
ancestor(X,Y):-parent(X,Z),ancestor(Z,Y).
```

Output:

```
?- ancestor(c, c).
true
Next 10 100 1,000 Stop
```

```
?- ancestor(c, a).
true
Next 10 100 1,000 Stop
```

Observations:

- Use " ; " to see more ancestors.
- While printing the list of all the ancestors, at termination condition /point it will return false which implies that there no more ancestors exists.

3. Construct your family tree diagram (start from grandparents to your siblings). and formulate defi- nitions 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:

```
male(umang_kumar).
male(dipender_kumar).
male(rakesh).
male(inder_pal).
male(ankit).
female(neetu_devi).
female(anchal).
parent(dipender_kumar,umang_kumar).
parent(neetu_devi,umang_kumar).
parent(dipender_kumar,anchal).
parent(inder_pal, dipender_kumar).
parent(inder_pal, rakesh).
parent(neetu_devi,anchal).
parent(rakesh, ankit).

%Father and Mother
father(X,Y):-parent(X,Y),male(X).
mother(X,Y):-parent(X,Y),female(X).

%sibling
sibling(X,Y):-mother(Z,Y),mother(Z,X).
sibling(X,Y):-father(Z,Y),father(Z,X).

%GrandParent
grandparent(X,Y):-father(Z,Y),father(X,Z).
grandparent(X,Y):-mother(Z,Y),mother(X,Z).

%GrandFather
```

```
%GrandFather
grandfather(X,Y):-father(Z,Y),father(X,Z).

%grandMother
grandmother(X,Y):-mother(Z,Y),mother(X,Z).

%Cousin
cousin(X,Y):-father(Z,X),father(A,Y),sibling(Z,A).

%Uncle
uncle(X,Y):-father(Z,Y),sibling(Z,X),not(father(X,Y)).

%Aunt
aunt(X,Y):-father(Z,Y),sibling(Z,A),father(A,B),mother(X,B),not(mother
```

Output:

```
?- sibling(umang_kumar, anchal).
true
Next 10 100 1,000 Stop

?- parent(dipender_kumar, umang_kumar)
true
Next 10 100 1,000 Stop

?- uncle(rakesh, umang_kumar).
true
Next 10 100 1,000 Stop
```

Observation:

- Don't forget to add "X\=Y" condition in case of same parent, since X,Y have same parent and prolog will check all the possible combinations.
- Facts should be clear while writing code, and check the rules properly.

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: criminial(X)?
Draw a resolution tree to find the answer of par

Answer:

Knowledge base in FOL

- it is crime for a american to sell weapons tto hostile nations

  american(x) ^ weapon(y) ^ sells(x,y,z) ^ hostile(z) => crimina(x)

- Nono has some missiles
  E x owns(none,x) ^ missile(x)
  owns(nono,m) and missile(M)

- all of its missile were sold to it by colonoel west
  missile(x) ^ owns(nono,x) => sells(west,x,nono)

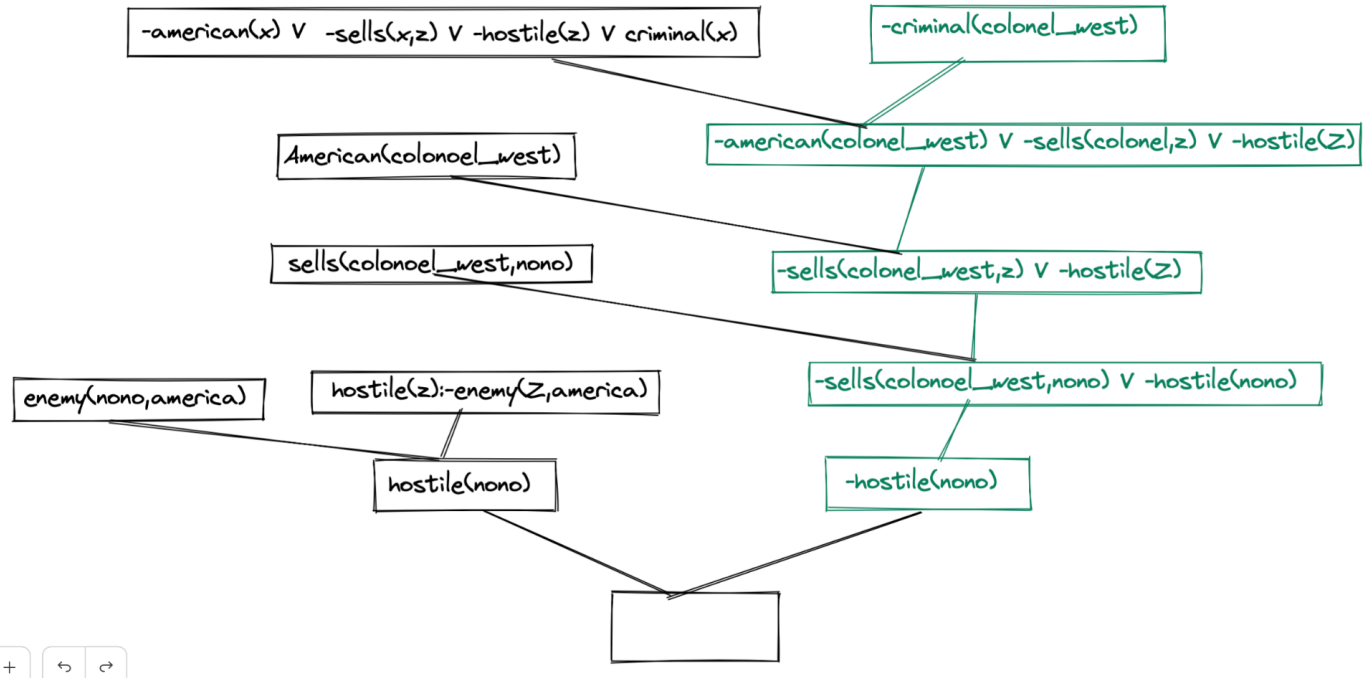
- Missiles are weapons
  Missiles(x) => Weapons(x)

- Enemy of America counts as "hostile"
  Enemy(x, America) => Hostile(x)

- West, who is American...
  American(west)

- The Country, Nono is an enemy of America
  Enemy(Nono, America)

RESOLUTION TREE



Code:

```
american(west).
missile(missile1).
weapon(Y):-missile(Y).
enemy(nono,america).
hostile(Z):-enemy(Z,america).
owns(nono,missile1).
sells(west,X,nono):- missile(X),owns(nono,X).
criminal(X) :- american(X),weapon(Y),sells(X,Y,Z),hostile(Z).
```

Output:

```
?- criminal(west).
```

true

```
?- criminal(X).
```

X = west

Observation:

We will start with taking negation of the required. We assume, west is not a criminal by taking negation of goal, but resolution tree gives us the NULL value. At the end our assumption was wrong. So we can say that, west is criminal. Hence proved.