7/6/2017

FIRST ORDER PREDICATE LOGIC

Artificial Intelligence Lab Report: 03

Ankit Shrestha

071/BCT/505

Submitted to:

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

Central Campus, Pulchowk

I.O.E.

T.U.

Theory:

Introduction to First Order Predicate Logic (FOPL)

The use of symbolic logic to represent knowledge is not new in that it predates the modern computer by a number of decades. Even so, the application of logic as a practical means of representing and manipulating knowledge was not demonstrated until the early 1960s. Today First Order Predicate Logic (FOPL) or predicate calculus has assumed one of the important roles in AI for representing the knowledge.

The understanding of FOPL for AI student has several benefits. One, logic offers the formal approach to reasoning that has a sound theoretical foundation. Next, the structure of FOPL is flexible enough to permit the accurate representation of the natural language reasonably well.

For Example:

- Ram loves all animals.
  $\forall x Animals(x) \Rightarrow Loves(ram,x)$
- Poppy is a dog.
  $Dog(Poppy)$
- Grandparent is a parent of one's parent
  $\forall x, y Grandparent(x, y) \Leftrightarrow \exists z Parent(x,z) \cap Parent(z, y)$
- Parent and child are inverse relation.
  $\forall x, y Parent(x, y) \Leftrightarrow Child(y,x)$
- Rules combine facts to increase knowledge of the system
  son(X,Y):- male(X),child(X).
  X is a son of Y if X is male and X is a child of Y.

Monkey- Banana Problem

Monkey-Banana Problem is the famous problem in AI. Where there is a room containing a monkey, a chair, and bananas that have been hung from the center of the ceiling of the room; out of reach from monkey. If the monkey is clever enough, he can reach the bananas by placing the chair directly below the bananas and climbing on the top of the chair.

Now the problem is to use FOPL to represent this monkey-banana problem and prove that monkey can reach the bananas.

PROGRAM:
Example-I:

```
PREDICATES in_room(symbol)
dexterous(symbol)
tall(symbol)
can_move(symbol,symbol,symbol)
can_reach(symbol,symbol)
get_on(symbol,symbol)
can_climb(symbol,symbol)
close(symbol,symbol)
under(symbol,symbol)
can_climb(symbol,symbol)
   CLAUSES in_room(bananas).
in_room(chair).
in_room(monkey).
dexterous(monkey). tall(chair).
can_move(monkey,chair,bananas).
can_climb(monkey,chair).

can_reach(X,Y):-
dexterous(X),close(X,Y).
 close(X,Z):-
     get_on(X,Y),
under(Y,Z),    tall(Y).
get_on(X,Y):-
can_climb(X,Y).
 under(Y,Z):-
     in_room(X),
in_room(Y),    in_room(Z),
can_move(X,Y,Z).
 GOAL
can_reach(monkey,appple)
```

Output:

No.

When `GOAL`
`can_reach(monkey,banana)`

Output:

Yes.

Assignment 1:

- Every American who sells weapons to hostile nations is a criminal.
  $\forall x,y$ (American(x) ∩ sells_missiles(x,y) ∩ hostile(y)) ⇒ Criminal(x)
- Every enemy of America is a hostile.
  $\forall x$ enemy_of_america(x) ⇒hostile(x)
- Iraque has some missiles.
  $\exists x$ missile(x) ⇒belongs(x,Iraq)
- All missiles of Iraque were sold by George.
  $\forall x,y$ (George(x) ∩ missile_of_iraq(y)) ⇒ sells_missile(x,y)
- George is an American. American(George) • Iraque is a country. Country(Iraq)
- Iraque is the enemy of America.
  Enemy_of_america("Iraq")
- Missiles are weapens.
  weapon(missile)

```
PREDICATES
hostile(STRING)
enemy_of_america(STRING)
american(STRING)
criminal(STRING)
sells_missiles(STRING, STRING)
has_missile(STRING)
country(STRING)
 CLAUSES criminal(X):-   american(X),
sells_missiles(X, Y), hostile(Y).
enemy_of_america(X) :-
hostile(X).
enemy_of_america("Iraq").
hostile(X):-   country(X).
has_missile("Iraq").
sells_missiles("George", "Iraq").
american("George"). country("Iraq").

GOAL
```

```
criminal("George").
```

Output:

Yes.

Assignment 2:

- Horses are mammals.
  ∀x horse(x) ⇒mammal(x)
- An offspring of a horse is a horse.
  ∀x,y(horse(x) ∩ offspring(y,x))⇒horse(y)
- Bluebeard is a Charlie's parent.
  Parent(bluebeard,Charlie)
- Offspring and parents are inverse relations.
  ∀x, yParent(x, y) ⇔ offsring(y,x)
- Every mammal has a parent.
  ∀x mammal(x) ⇔∃yParent(y,x)
- Bluebeard is a horse.
  Horse(Bluebeard)

```
PREDICATES
horse(STRING)
mammals(STRING)
offspring(STRING, STRING)
parent(STRING, STRING)

CLAUSES
parent("Bluebeard", "Charlie").
horse("Bluebeard").
horse(X):-
     mammals(X), offspring(X,Y), horse(Y).
mammals(X):-
     parent(Y,X), offspring(X,Y).
offspring(X,Y):-
     parent(Y,X).
```

```
GOAL
horse("Charlie").
```

Output:

Yes.

Discussion:

First Order Predicate Language (FOPL) has been used in the examples and assignments of this lab to model natural language problems into statements for logical solution. FOPL has been used here for an accurate representation of the natural language problems by forming predicates for various statements. FOPL provides a formal approach for reasoning and has a very sound theoretical foundation.

Prolog has been used in all the examples and assignments of the lab for the implementation of FOPL. In the assignments, FOPL was first designed for all the given statements of the natural language problem and the FOPL was used for developing logic for the program to solve the given problem.

Conclusion:

Hence, from this lab we familiarized ourselves with First Order Predicate Logic (FOPL) and its implementation in Prolog and understood its basic concepts.