Lab Report on Propositional Logic

Sumit Yadav (076BCT088)
Institute of Engineering, Pulchowk Campus

February 4, 2023

Abstract

The text presents five sets of premises with corresponding questions or goals to be proved. The first set of premises deals with the relationship between horses, cows, pigs, and offspring in terms of mammal classification. The question is to determine if Charlie is a horse. The second set of premises focuses on the relationship between poverty, intelligence, reading, and happiness, with the goal of finding if anyone has an exciting life. The third set of premises revolves around loyalty and hatred towards Caesar among the Romans, with the goal of determining if Marcus hated Caesar. The fourth set of premises deals with Bhogendra's food preferences and Shailendra's relationship with Bhogendra, with the goal of finding out if Shailendra likes chicken. The fifth set of premises deals with a dancing club and the members' ability to waltz and jive, with the goal of proving that there is a member who can't jive.

1 Introduction

1.1 Propositional Logics

Logic is the basis of all mathematical reasoning, and of all automated reasoning. A proposition is the basic building block of logic. It is defined as a declarative sentence that is either True or False, but not both. For Example,

1. The sun revolves around the Earth
2. 1 + 1 = 2
3. Mount Everest is the highest peak in the world

All of the above sentences are propositions, where the last two are Valid (True) and the first one is Invalid (False). Some sentences that do not have a truth value or may have more than one truth value are not propositions. For Example,

1. What is your name?
2. 4 + x = 8

The above sentences are not propositions, as the first one does not have a truth value, and the second one may be true or false. The area of logic which deals with propositions is called propositional calculus or propositional logic. It also includes producing new propositions using existing ones. Propositions constructed using one or more propositions are called compound propo- sitions. The propositions are combined together using Logical Connectives or Logical Operators.

1.2 First Order Preidcates Logics

First Order Predicate Logic is a way of knowledge representation in artificial intelligence. It is an extension to propositional logic. FOPL is sufficiently expressive to represent the natural language statements in a concise way. It is a powerful language that develops information about the objects in a more easy way and can also express the relationship between those objects. For example, let us consider a statement "X is a man." which has two parts; first the variable X, is the subject of the statement and the second part "is a man" is called predicate which represents the property that the subject of the statement can have. It may be denoted as man(X). Once a variable has been assigned to the propositional function man(X), it becomes propositional logic and has a associated truth value.

1.3 Quantifiers in First Order Predicate Logic

A quantifier is a language element which generates quantification, and quantification specifies the quantity of specimen in the universe of discourse. These are the symbols that permit to determine or identify the range and scope of the variable in the logical expression. There are two types of quantifier:

1. Universal Quantifier, (for all, everyone, everything)
2. Existential Quantifier, (for some, at least one)

1.3.1 Universal Quantifier

Universal quantifier is a symbol of logical representation, which specifies that the statement within its range is true for everything or every instance of a particular thing. The Universal quantifier is represented by a symbol $\forall$.

1.3.2 Existential Quantifier

Existential quantifiers are the type of quantifiers, which express that the statement within its scope is true for at least one instance of something. The Existential quantifier is represented by a symbol $\exists$.

2 Experimental Setup

For this lab assignment, the experiment setup consisted of using a laptop with the following specifications:

1. Processor: Intel i5 8th generation
2. RAM: 8 GB
3. Operating System: Linux

The experiments were conducted on this laptop using a Prolog interpreter. The laptop was used to write, compile, and run the Prolog programs for the various tasks in the lab assignment.

3 Analysis

3.1 Question 1

Premises:

1. Horses, cows, pigs are mammals.
2. An offspring of a horse is a horse.
3. Bluebeard is a horse.
4. Bluebeard is Charlie's parent.
5. Offspring and parent are inverse relations.
6. Every mammal has a parent.

Query:

1. Is Charlie a horse?

```
PREDICATES
    horse(STRING)
    mammal(STRING)
    cow(STRING)
    pig(STRING)
    is_offspring(STRING, STRING)
    is_parent(STRING, STRING)
CLAUSE
    mammal(X) :- leftarrow horse(X), cow(X), pig(X), is_parent(_, X).
    horse(Y):- is_parent(X, Y), horse(X), horse("Bluebeard").
    is_offspring(X, Y):- is_parent(Y, X)
    is_parent("Bluebeard", "Charlie")}.cow("_").
    pig("_").

GOAL
    horse("Charlie").
```

The output is:
Yes

3.2 Question 2

Premises:

1. All people who are not poor and smart are happy.
2. Those people who are not stupid.
3. John can read and is wealthy.
4. Happy people have exciting lives.

Query:

1. Can anyone be found with an exciting life?

```
PREDICATES
    not_poor(STRING)
    smart(STRING)
    happy(STRING)
    not_stupid(STRING)
    read(STRING)
    wealthy(STRING)
CLAUSES
    exciting_life(STRING)
    exciting_life(X):- happy(X).
    happy(X):- not_poor(X), smart(X).
    smart(X):- not_stupid(X).
    not_stupid(X):- read(X).
    not_poor(X):- wealthy(X).
    read("John").
    wealthy("John").

GOAL
    exciting_life("John").
```

The output is:
Yes

3.3 Question 3

Premises:

1. All Pompeian are Romans.
2. All Romans were either loyal to Caesar or hated him.
3. Everyone is loyal to someone.
4. People only try to assassinate Caesar.
5. Marcus tried to assassinate Caesar.
6. Marcus was Pompeian.

Query:

1. Did Marcus hate Caesar?

```
PREDICATES
    pompeian(symbol)
    nondeterm roman(symbol)
    nondeterm loyal(symbol, symbol)
    nondeterm hate(symbol, symbol)
    nondeterm assassinate(symbol, symbol)
    nondeterm not_loyal(symbol, symbol)
CLAUSES
    roman (X) :- pompeian (X).
```

```
    assassinate (marcus, caesar).pompeian (marcus)
    hate (X, caesar) :- roman (X), not_loyal (X, caesar)
    loyal (X, caesar) :- roman (X), not (hate (X, caesar))
    not_loyal :- assassinate (X, Y)


GOAL
    hate (marcus, caesar)
```

The output is:
Yes

3.4 Question 4

Premises:

1. Bhogendar likes all kinds of food.
2. Oranges are food.
3. Chicken is food.
4. Anything anyone eats and isn't killed by food.
5. If a person likes food means that person has eaten it.
6. Jogendar eats peanuts and is still alive.
7. Shailendra eats everything Bhogendar eats.

Query:

1. Does Shailendra like chicken?

```
PREDICATES
    nondeterm likes(STRING, STRING)
    nondeterm food(STRING)
    nondeterm eats(STRING)
    nondeterm kill(STRING, STRING)
CLAUSES
    likes(X, Y) :- food(Y).
    food("orange").
    food("chicken").
    food(X) :- likes(Y, X), not(kills(X, Y)).
    eats("sailendra", Y) :- eats("bhogendra", Y).
    eats(X, Y) :- likes(X, Y), food(Y).
    kills(_, _).


GOAL
    likes("sailendra", "chicken").
```

The output is:
No

3.5 Question 5

Premises:

1. Dave and Fred are members of a dancing club in which no member can both waltz and jive.
2. Fred's dad can't waltz, can do whatever Fred can't do.
3. If a child can do something, then their parents can do it also.

Query:

1. There is a member of the dancing club who can't jive.

```
PREDICATES

nondeterm can_do(STRING, STRING)
nondeterm member(STRING, STRING)
nondeterm parent(STRING, STRING)
nondeterm cant_do(STRING, STRING)

CLAUSES

member("dave", "dancingclub").
member("fred", "dancingclub").
cant_do("freddad", "waltz").
cant_do(X, Y) :- parent(Z, X), cant_do(Z, Y).
cant_do(X, "waltz") :- member(X, "dancingclub"), can_do(X, "jive").
cant_do(X, "jive") :- member(X, "dancingclub"), can_do(X, "waltz").
can_do("dave", X) :- cant_do("fred", X).
parent("freddad", "fred").

GOAL

member(X, "dancingclub"), cant_do(X, "jive").
```

The output is:
X - dave
solution 1

4 Conclusion

The above text describes five different scenarios that raise various questions or queries based on the premises given.

1. The first scenario is about Charlie and his relationship to horses, cows, pigs and Bluebeard. The premises describe these animals as mammals and Bluebeard as a horse, as well as Bluebeard being Charlie's parent. The query asks if Charlie is a horse.
2. The second scenario is about people's happiness, wealth, and intelligence. The premises describe conditions for being happy, smart, and wealthy and the relationship between reading and intelligence. The question asks if anyone can be found with an exciting life.

3. The third scenario is about Marcus, Caesar, and the people of Pompeii and Rome. The premises describe the relationships between the Romans, the Pompeians, and their loyalty to Caesar. The question asks if Marcus hated Caesar.

4. The fourth scenario is about Bhogendra, Jogendra, and Shailendra's eating habits. The premises describe Bhogendra's liking for all kinds of food, what is considered food, and how liking a food means a person has eaten it. The question asks if Shailendra likes chicken.

5. The fifth scenario is about a dancing club and its members' ability to waltz and jive. The premises describe the club's rules and the relationship between Dave and Fred and their ability to do these dances. The goal is to prove that there is a member of the club who can't jive.

In conclusion, the text presents various premises and raises questions or goals based on those premises, asking the reader to make conclusions based on the information given. latex format