Ahsanullah University of Science and Technology

Department of Computer Science and Engineering

CSE4108
Artificial Intelligence Lab

Term Assignment: 01

Submitted By:

Name : Devopriya Tirtho
Id :16.02.04.033

Date of Submission: **1st September, 2020**

Term Assignment 1: Reasoning with Propositional Logic.

Introduction: We are living in an age of modern science. Science is evolving day by day. At present, we have come to a point of science where **'Artificial Intelligence'** is ruling the world. 'Artificial Intelligence' means the intelligence which is given to a machine and by using the intelligence it can train itself to do further tasks. Now a days, computer science is more focused on developing 'Artificial Intelligence' than other fields. To understand and to do well in this field, we have to learn from the scratch. **'Propositional Logic'** is one of the root topics of this field. 'Propositional Logic' helps us to understand a problem solely and then it becomes easy to infer and make reasonings.

Propositional Logic: 'Propositional Logic' is one kind of logic which is based on true or false result. It is also called as **'Boolean Logic'**. A proposition can have only one value and the value has to be either true or false. This is the most widely used form of logical system and also the most basic one. In this system, logical operators combine propositions to create other propositions followed by some rules.

Propositional Logic in Artificial Intelligence: In 'Artificial Intelligence', propositional logic works as the root of many problems. It leads to the basic setup of 'Machine Learning'. There are some other fields, where propositional logic works. These are:

- **Planning**
- **Problem-solving**
- **Intelligent Control**
- **Decision-making**

Reasoning with Propositional Logic: Propositional logic is used to formally reason any proposition. Propositional logic works on a knowledgebase. From the knowledgebase, we can finally get some facts which is formally reasoned by the help of propositional logic. To make reason with propositional logic, we have to maintain a chunk of information, from which any conclusion of result can be derived. As we want to know about reasoning with propositional logic, we need to learn about some terms and we need to understand the workflow for reasoning.
These terms are:

- **Knowledge Base (KB):** Knowledge base is a storage system of storing given information or the information we have to solve a problem. The 'KB' keeps the records of all information.
  Example: Rahim is the father of Karim. Karim is the father of Hasib. If anyone's father has a father, that person is the grandfather of him/her.
  We can state it as a knowledgebase.

Facts about Propositional Logic: A propositional logic has some key facts. These are:

- In propositional logic, we use symbolic variables to represent the logic, and we can use any symbol for a representing a proposition, such A, B, C, P, Q, R, etc.
- Propositions can be either true or false, but it cannot be both.
- Propositional logic consists of an object, relations or function, and **logical connectives**.
- The propositions and connectives are the basic elements of the propositional logic.
- Connectives can be said as a logical operator which connects two sentences.
- Statements which are questions, commands, or opinions are not propositions such as "**Where is Rohini**", "**How are you**", "**What is your name**", are not propositions.
- A propositional logic can have a 'Premise' and 'Conclusion'. From the 'Premise'→'Conclusion' relationship, it says if the premise is true, the conclusion has to be true or when the premise is false, the conclusion must be false.
- Propositions are sometimes connected by connectives. Here is some information about connectives:

Connective symbols	Word	Technical term	Example
∧	AND	Conjunction	A ∧ B
∨	OR	Disjunction	A ∨ B
→	Implies	Implication	A → B
⇔	If and only if	Biconditional	A⇔ B
¬ or ~	Not	Negation	¬ A or ¬ B

Types of Propositional Logic: There are two types of 'Propositional Logic'

1. **Atomic Proposition:** Atomic propositions are the simple propositions. It consists of a single proposition symbol. These are the sentences which must be either true or false.

   Example: 2+2 is 4, it is an atomic proposition as it is a **true** fact.

2. Compound Proposition: Compound propositions are constructed by combining simpler or atomic propositions, using parenthesis and logical connectives.
   Example: I go to school and he comes to my house.

Workflow with an Example:

Knowledge Base (KB):

There lives a lion in a jungle. There lives a tiger also in the jungle. A friend of the tiger is an elephant, who lives in another jungle. When there is a tiger and a lion are present in the jungle, a fight happens. The fight is initiated by the lion. If the lion starts fighting, the tiger also starts. When the fight happens between a lion and a tiger, always the mightier one wins, which means the lion wins. But when the lion wins against the tiger, this news spreads throughout the jungle and to the neighboring jungle also. The elephant cannot resist himself/herself. The elephant comes if the tiger is defeated by the lion. When there is an elephant and a lion are present at the jungle, a fight happens again. This time, both the elephant and the lion start fighting. In this fight, the elephant wins. In a jungle, the last survivor becomes the king. Therefore, the elephant becomes the king of the jungle.

Representation in 'Propositional Logic':

1. **A [A tiger lives in a jungle]**
2. **B [A lion lives in a jungle]**
3. **C [An elephant lives in another jungle]**
4. **A ∧ B → D [If a tiger and a lion lives in a jungle, then a lion starts fighting]**
5. **D → E [If a lion starts fighting, then a tiger starts fighting]**
6. **D ∧ E → F [If a lion starts fighting and a tiger starts fighting, then the lion wins]**
7. **F ∧ C → G [If a lion wins and there is an elephant in another jungle, then the elephants comes to the jungle]**
8. **B ∧ G → H [If a lion is in the jungle and an elephant is in the jungle, then the elephant starts fighting]**
9. **B ∧ G → D [If a lion is in the jungle and an elephant is in the jungle, then the lion starts fighting]**
10. **D ∧ H → I [If a lion starts fighting and an elephant starts fighting, then the elephant wins]**
11. **I → K [If the elephant wins, then the elephant becomes the king of the jungle]**

Here **A (1), B (2), C (3)** are facts and **4, 5, 6, 7, 8, 9, 10, 11** numbered propositions are rules. Here, we have to find that if the elephant is the king of the jungle is or not by reasoning.

Procedure: As, A, B and C are facts. These are included to the KB.

- Then we take, **A ∧ B → D.**

Proof by Truth Table:

A	B	A ∧ B	D
0	0	0	0
0	1	0	0
1	0	0	0
1	1	1	1

So, here we can say that only when A is true and B is true, then D will be true.
So, after reasoning we can add 'D' to the 'KB'.

- Then, **D → E:**
  As, we prove that 'D' is True, so 'E' has to be true. So, after reasoning 'E' is added to the knowledgebase.

- Then, **D ∧ E → F:**
  We have already proven that, D and E both are added to the knowledgebase as fact. So, from the premise, we can say that as D and E are true, so F has to be true. After reasoning, 'F' is added to the 'KB'.

- **F ∧ C → G:**
  As 'F' is added to the 'KB' and 'C' is already given as a fact. So, from F ∧ C it can be said that, if F and C both are true, then 'G' will be true. So, after reasoning 'G' is added to the knowledgebase.

- **B ∧ G → H:**
  Here, B is a fact and G has already been added to the knowledgebase. So, H must be true. After reasoning, H is added to the 'KB'.

- **B ∧ G → D:**
  Here, B is a fact and G has already been added to the knowledgebase. So, D must be true. D has already been added to the knowledgebase.

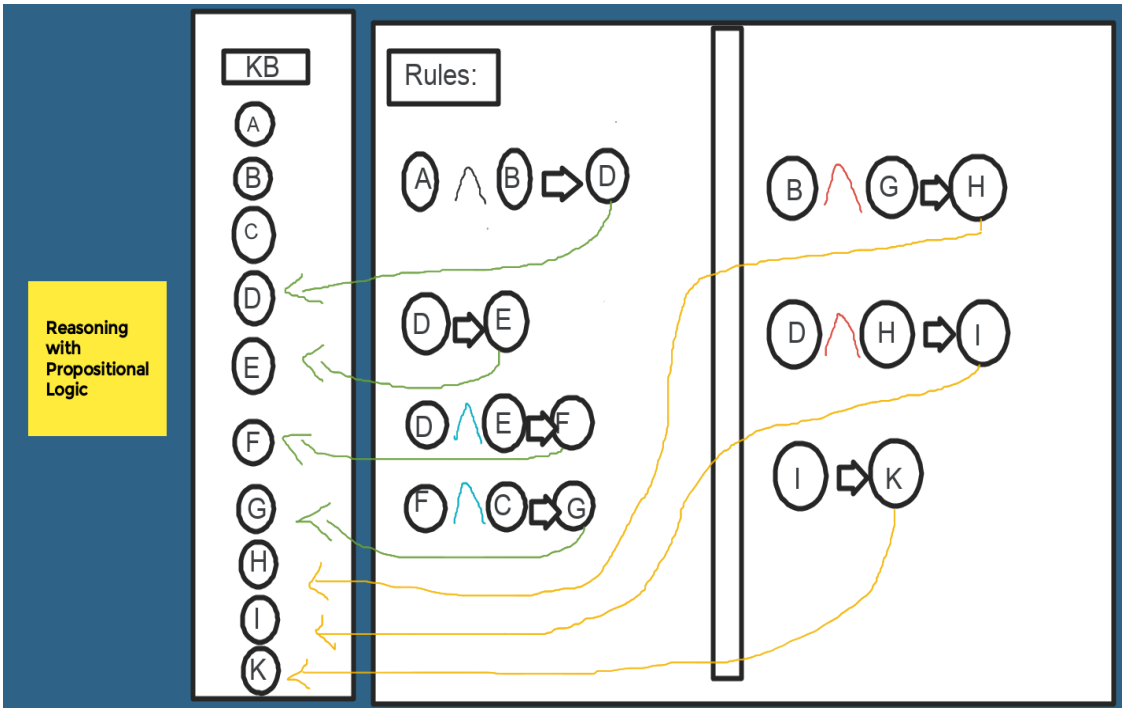
- **D ∧ H → I:**
  As, D and H both are present in the 'KB'. So, after reasoning I has been added to the knowledgebase.

- **I → K:**
  As, I is true, then K is true. After reasoning K is added to the knowledgebase.

If we take the conclusion first, the we can follow some steps for reasoning. These are:

- As, **I → K** is given, we have to find that I is true or not. For searching this, we have to backtrack.
- Then we find, **D ∧ H → I**. Here, D and H both are unavailable in the knowledgebase. So, we have to backtrack for D and H.
- Then We get **B ∧ G → D**, Here B is present in the 'KB' but G is not present in the 'KB'. So, again we backtrack.
- We find**, F ∧ C → G**, here C is present in the 'KB' but F is not. So, we search again.
- **D ∧ E → F**, here, D and E both are absent from KB. So, we have to look back again.
- Then we find D → E, here, D is absent in the KB. So, we search back.
- We see **A ∧ B → D**, here, A and B both are present in the KB, so D is true. Now we can look forward. After reasoning D is added to the KB.
- From **D → E**, E can be added to the KB.
- By going forward, we progressively add F, G, I and lastly K to the KB.

The whole process can be visualized as this picture:



Input of The Program:

1. **A**
2. **B**
3. **C**
4. **A ∧ B → D**
5. **D → E**
6. **D ∧ E → F**
7. **F ∧ C → G**
8. **B ∧ G → H**
9. **B ∧ G → D**
10. **D ∧ H → I**
11. **I → K**

Output of The Program:
The Final Knowledgebase will be:
A, B, C, D, E, F, G, H, I, K

Solution Code: The Code is written in ‘Python’.

```
# -*- coding: utf-8 -*-
"""
Creasoningeated on Tue Sep  1 03:50:11 2020

@authoreasoning: Hp
"""


class reasoning:
    def __init__(self, pr1, pr2, conslusion, t):
        self.pr1 = pr1
        self.pr2 = pr2
        self.conslusion = conslusion
        self.t = t


list = []
T = 'True'

list.append(reasoning('A', 'B', 'D', 0))
list.append(reasoning('D', 'T', 'E', 0))
list.append(reasoning('D', 'E', 'F', 0))
list.append(reasoning('F', 'C', 'G', 0))
list.append(reasoning('B', 'G', 'H', 0))
list.append(reasoning('B', 'G', 'D', 0))
list.append(reasoning('D', 'H', 'I', 0))
list.append(reasoning('I', 'T', 'K', 0))


kb = dict()
kb['A'] = 'True'
kb['B'] = 'True'
kb['C'] = 'True'
kb['T'] = 'True'
kb['D'] = 'False'
kb['E'] = 'False'
kb['F'] = 'False'
kb['G'] = 'False'
kb['H'] = 'False'
kb['I'] = 'False'
kb['K'] = 'False'
```

```
print("Propositions are taken into action one by one:")
print("\n")


for i in list:
   if kb[i.pr1] == 'True' and kb[i.pr2] == 'True' and i.conslusion == 'D':
      kb['D'] = 'True'

flag = 0
begin = 0
storage = ''
l = []


while 1:
   for i in list:
      if kb[i.pr1] == 'True' and kb[i.pr2] == 'True' and i.t == 1 and i.conslusion == 'E':
         flag = 1
         break
      if i.conslusion == 'K' and begin == 0:

         if kb[i.pr1] == 'False':

            storage = i.pr1
            l.append(i.conslusion)
            l.append(storage)
         else:
            storage = i.pr2
            l.append(storage)
         print(i.pr1 + " & " + i.pr2 + " -> " + i.conslusion)
         i.t = 1
         begin = 1
      elif i.conslusion == storage:
         if kb[i.pr1] == 'False':
            storage = i.pr1
            l.append(storage)
         elif kb[i.pr2] == "False":
            storage = i.pr2
            l.append(storage)

         i.t = 1
         print(i.pr1 + " & " + i.pr2 + " -> " + i.conslusion)
```

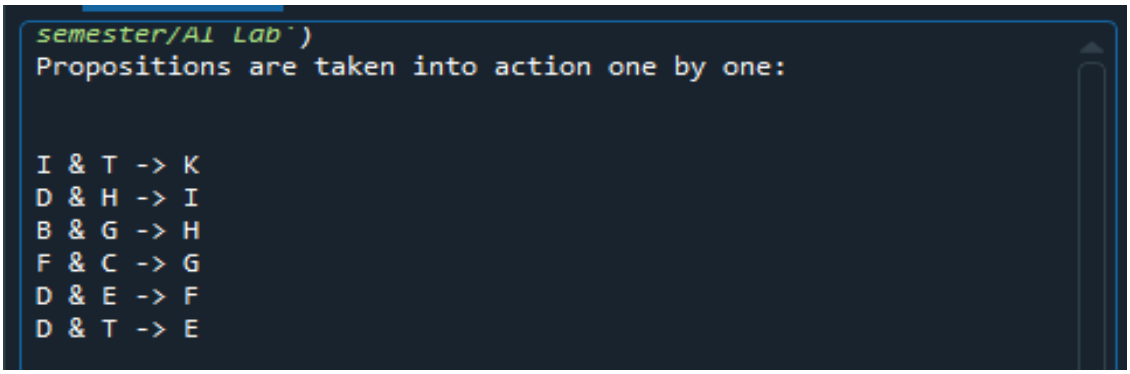
```
    if flag == 1:
        l.append('D')
        l.append('C')
        l.append('B')
        l.append('A')
        break


print("\n")

sz = len(l) -1
while sz >= 0:
    kb[l[sz]] = "True"
    print("After reasoning "+ l[sz]+" "+"is added to the KnowledgeBase")
    sz = sz - 1


print("\n")
print("The Final KnowledgeBase is: \n")


sz = len(l) -1
while sz >= 0:
    kb[l[sz]] = "True"
    print(l[sz] +" " +kb[l[sz]] +": is a fact")
    sz = sz - 1


a = str(input('Give the first input of a rule: '))
b = str(input('Give the second input of a rule: '))
print("\n")

if a == 'A' and b == 'B':
    print('A & B -> D       : If a tiger and a lion lives in a jungle, then a lion starts fighting')
elif a == 'D' and b == 'T':
    print("D -> E           :If a lion starts fighting, then a tiger starts fighting")
elif a == 'D' and b == 'E' :
    print("D & E -> F       :If a lion starts fighting and a tiger starts fighting, then the lion wins")
elif a == 'F' and b == 'C':
    print("F & C -> G      :If a lion wins and there is an elephant in another jungle, then the elephants
comes to the jungle")
elif a == 'B' and b == 'G':
    print("B & G -> H      : If a lion is in the jungle and an elephant is in the jungle, then the elephant
starts fighting")
```

```
elif a == 'D' and b == 'H':
   print("D & H -> I      : If a lion starts fighting and an elephant starts fighting, then the elephant wins")
elif a == 'I' and b == 'T':
   print("I - > k         :If the elephant wins, then the elephant becomes the king of the jungle")
```

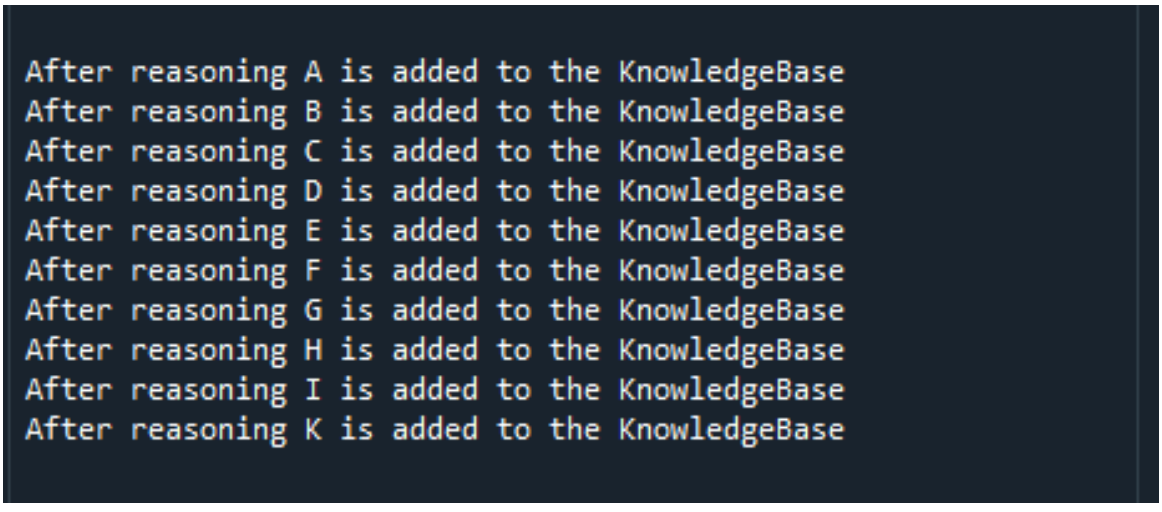
##
##

Output:

Firstly, the output will show the propositions it takes for reasoning one by one.



Secondly, it will show the saved database of iterations. After each reasoning the newly added member of the knowledgebase will be printed.



Then, the output will show the final Knowledgebase.

```
The Final KnowledgeBase is:

A True: is a fact
B True: is a fact
C True: is a fact
D True: is a fact
E True: is a fact
F True: is a fact
G True: is a fact
H True: is a fact
I True: is a fact
K True: is a fact
```

Finally, the program ask user to give input of any proposition to learn what it says.

```
Give the first input of a rule: A

Give the second input of a rule: B
A & B -> D       : If a tiger and a lion lives in a jungle,
then a lion starts fighting
```