

CS 820 Assignment 1

Akshat Mahajan

200424131

1 Introduction of Unification Algorithm

The unification problem in first-order logic can be expressed as follows: Given two terms containing some variables, find, if it exists, the simplest substitution (i.e., an assignment of some term to every variable) which makes the two terms equal. The resulting substitution is called the most general unifier .

2 Code Description

In order to run the code you need to unzip the folder. Then you will need to run the solution.py class. The algorithm is coded using python 2.7 which is compatible with Hercules. The code consists of 4 python files :

- solution.py : The function of the python class is to take input from the user and remove the spaces from the input entered by user.
- unification.py : The function of the class is to check the elements present in the input if they are constant, variable or a function. This class in turn calls the unifyWithOccursCheck method of unify class.
- unify.py : The main function of the class is to check the occurrence check in the algorithm whether or not the element we are unifying are present in the element to be unified or not.
- unifyFunctions.py : The function of the python class is to see if the input can be unified or not.

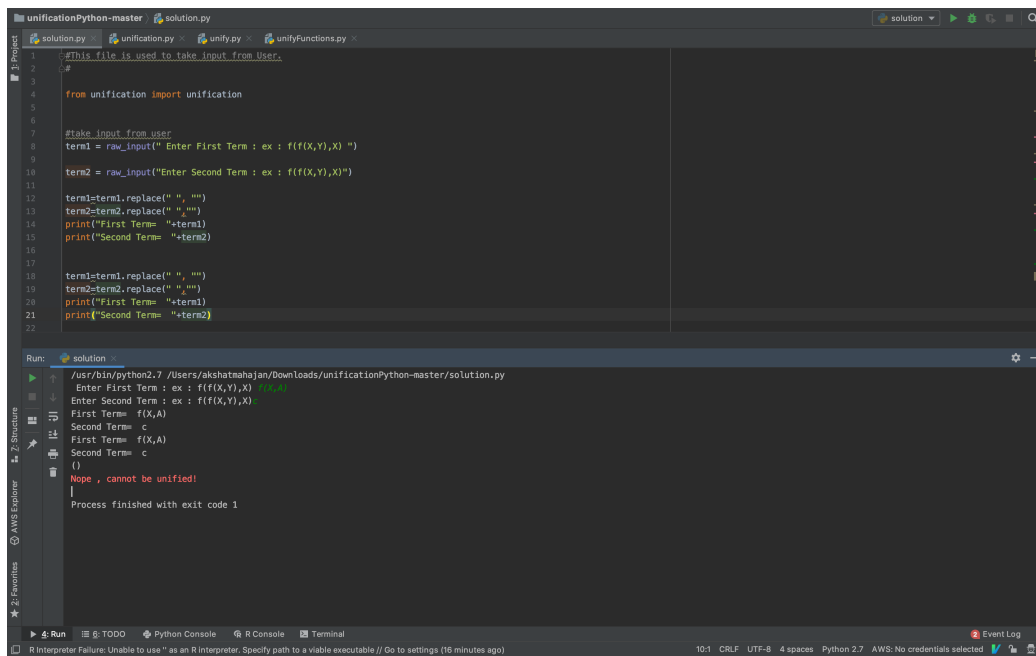
3 Screenshots

```
1 # This file is used to take input from user.
2 #
3
4 from unification import unification
5
6
7 #take input from user
8 term1 = raw_input("Enter First Term : ex : f(f(X,Y),X) ")
9
10 term2 = raw_input("Enter Second Term : ex : f(f(X,Y),X)")
11
12 term1=term1.replace(" ", "")
13 term2=term2.replace(" ", "")
14 print("First Term= "+term1)
15 print("Second Term= "+term2)
16
17
18 term1=term1.replace(" ", "")
19 term2=term2.replace(" ", "")
20 print("First Term= "+term1)
21 print("Second Term= "+term2)
```

Run: solution

```
/usr/bin/python2.7 /Users/akshatnaha/Downloads/unificationPython-master/solution.py
Enter First Term : ex : f(f(X,Y),X)
Enter Second Term : ex : f(f(X,Y),X)
First Term= X
Second Term= h(a,Y)
First Term= X
Second Term= h(a,Y)
()
Yes ,Unification Is Possible
'X'='h( 'a', 'Y')"
Process finished with exit code 0
```

Figure 1: Unification is Possible



```
1 #This file is used to take input from user.
2 #
3
4 from unification import unification
5
6
7 #take input from user
8 term1 = raw_input("Enter First Term : ex : f(f(X,Y),X) ")
9
10 term2 = raw_input("Enter Second Term : ex : f(f(X,Y),X)")
11
12 term1=term1.replace(" ", "")
13 term2=term2.replace(" ", "")
14 print("First Term= "+term1)
15 print("Second Term= "+term2)
16
17
18 term1=term1.replace(" ", "")
19 term2=term2.replace(" ", "")
20 print("First Term= "+term1)
21 print("Second Term= "+term2)
22
```

Run: solution

```
/usr/bin/python2.7 /Users/akshatnaha/Downloads/unificationPython-master/solution.py
Enter First Term : ex : f(f(X,Y),X)
Enter Second Term : ex : f(f(X,Y),X)
First Term= f(X,A)
Second Term= c
First Term= f(X,A)
Second Term= c
()
Nope , cannot be unified!
Process finished with exit code 1
```

R Interpreter Failure: Unable to use " as an R interpreter. Specify path to a viable executable // Go to settings (16 minutes ago)

Figure 2: Unification Not possible