AI LAB TEST 2

Name: Arpana M Ramaswamy
USN:1BM18CS147
Date:29/12/2020

Q)Infer whether the given pair of statements can be unified or not. If unification is possible write the code for substitutions. Justify your answer for these cases also along with successful unification
1) Predicates are different 2) Mismatch in Number of arguments 3) If the arguments are constants

Program

```
# -*- coding: utf-8 -*-
"""
Created on Tue Dec 29 14:39:10 2020

@author: Arpana M R
"""

def Seperate(li):
    li = li.split("(")[1:]
    li = "(".join(li)
    li = li.split(")")[:-1]
    li = ")".join(li)
    x = li.split(',')
    return x


def PredicateCheck(s1,s2):
    return (s1.split("(")[0]==s2.split("(")[0])
```

```
def printSolution(subs):
    print("Unification is possible")
    print("Substitution:",end=" ")
    subans=[]
    for k, v in subs.items():
        subans.append(k)
        subans.append('/')
        subans.append(v)

    print('unifier θ= {', end=' ')
    for i in range(len(subans)):
        if(i%3==2 and i!=len(subans)-1):
            print(subans[i],end=',')
        else:
            print(subans[i],end='')
    print(' }')



def Unify(s1,s2,const,var):

    if(PredicateCheck(s1,s2)==False):
        print("Predicates dont match ")
        print("Unifcation not possible")
        return
```

```
a=Seperate(s1)
b=Seperate(s2)
len1=len(a)
len2=len(b)
if(len1!=len2):
    print("Mismatch in number of arguments")
    print("Unification not possible")
    return

subs=[]
flag=1
for i in range(len(a)):
    if(a[i] in const and b[i] in const):
        if (a[i]==b[i]):
            continue
        else:
            print("constant cannot be equated")
            print("Unification is not possible")
            flag=0
            break
    elif(a[i] in const and b[i] in var):
        subs.append([b[i],a[i]])
    elif(a[i] in var and b[i] in const):
        subs.append([a[i],b[i]])
    elif(a[i] in var and b[i] in var):
        subs.append([a[i],b[i]])
if(flag==1):
```

```
    d=dict()
    for ele in subs:
        if(ele[0] in d):
            val=d.get(ele[0])
            if(val!=ele[1]):
                print("Different subsitutions for same variable")
                print("Unification is not possible")
                return
        else:
            d[ele[0]]=ele[1]

    t = {val : key for key, val in d.items()}
    subs = {val : key for key, val in t.items()}

    printSolution(subs)




const=list(input("Enter the constants : ").split(','))
var=list(input("Enter the variables : ").split(','))

s1=input("Enter first statement : ")
s2=input("Enter second statement : ")

Unify(s1,s2,const,var)
```

Program Screenshots

```
# -*- coding: utf-8 -*-
"""
Created on Tue Dec 29 14:39:10 2020

@author: Arpana M R
"""

def Seperate(li):
    li = li.split("(")[1:]
    li = "(".join(li)
    li = li.split(")")[:-1]
    li = ")".join(li)
    x = li.split(',')
    return x


def PredicateCheck(s1,s2):
    return (s1.split("(")[0]==s2.split("(")[0])


def printSolution(subs):
    print("Unification is possible")
    print("Substitution:",end=" ")
    subans=[]
    for k, v in subs.items():
        subans.append(k)
        subans.append('/')
        subans.append(v)

    print('unifier δ= {', end=' ')
    for i in range(len(subans)):
        if(i%3==2 and i!=len(subans)-1):
            print(subans[i],end=',')
        else:
            print(subans[i],end='')
    print(' }')




def Unify(s1,s2,const,var):

    if(PredicateCheck(s1,s2)==False):
        print("Predicates dont match ")
        print("Unifcation not possible")
        return
```

```
47
48      a=Seperate(s1)
49      b=Seperate(s2)
50      len1=len(a)
51      len2=len(b)
52      if(len1!=len2):
53          print("Mismatch in number of arguments")
54          print("Unification not possible")
55          return
56
57      subs=[]
58      flag=1
59      for i in range(len(a)):
60          if(a[i] in const and b[i] in const):
61              if (a[i]==b[i]):
62                  continue
63              else:
64                  print("constant cannot be equated")
65                  print("Unification is not possible")
66                  flag=0
67                  break
68          elif(a[i] in const and b[i] in var):
69              subs.append([b[i],a[i]])
70          elif(a[i] in var and b[i] in const):
71              subs.append([a[i],b[i]])
72          elif(a[i] in var and b[i] in var):
73              subs.append([a[i],b[i]])
74      if(flag==1):
75          d=dict()
76          for ele in subs:
77              if(ele[0] in d):
78                  val=d.get(ele[0])
79                  if(val!=ele[1]):
80                      print("Different subsitutions for same variable")
81                      print("Unification is not possible")
82                      return
83              else:
84                 d[ele[0]]=ele[1]
85
86          t = {val : key for key, val in d.items()}
87          subs = {val : key for key, val in t.items()}
88
89          printSolution(subs)
```

```
89          printSolution(subs)
90
91
92  const=list(input("Enter the constants : ").split(','))
93  var=list(input("Enter the variables : ").split(','))
94
95  s1=input("Enter first statement : ")
96  s2=input("Enter second statement : ")
97
98  Unify(s1,s2,const,var)
99
```

OUTPUT SCENARIOS

A possible unification is->

```
In [1]: runfile('D:/CSE/5thsem/AI Lab/
Labtest2/1BM18CS147-prog.py', wdir='D:/
CSE/5thsem/AI Lab/Labtest2')

Enter the constants : Ram,Kishan

Enter the variables : x,y

Enter first statement : Brotherof(x,Ram)

Enter second statement :
Brotherof(Kishan,y)
Unification is possible
Substitution: unifier θ= { x/Kishan,y/Ram
}
```

1) Predicates are different
   If predicates (the functions) are different, then the statements cannot be unified.
   OUTPUT->

```
In [2]: runfile('D:/CSE/5thsem/AI Lab/
Labtest2/1BM18CS147-prog.py', wdir='D:/
CSE/5thsem/AI Lab/Labtest2')

Enter the constants : Dolly,Rani

Enter the variables : x,y

Enter first statement :
Sisterof(Dolly,Rani)

Enter second statement : Motherof(x,y)
Predicates dont match
Unifcation not possible
```

If predicates match and all other conditions are satisfied, unification is possible. OUTPUT->

```
In [8]: runfile('D:/CSE/5thsem/AI Lab/
Labtest2/1BM18CS147-prog.py', wdir='D:/CSE/5thsem/AI
Lab/Labtest2')

Enter the constants : Dolly,Rani

Enter the variables : x,y

Enter first statement : Motherof(Dolly,Rani)

Enter second statement : Motherof(x,y)
Unification is possible
Substitution: unifier θ= { x/Dolly,y/Rani }
```

2) Mismatch in Number of arguments
If number of arguments don't match, Unification is not possible

```
In [3]: runfile('D:/CSE/5thsem/AI Lab/
Labtest2/1BM18CS147-prog.py', wdir='D:/CSE/5thsem/AI
Lab/Labtest2')

Enter the constants : Ram,Raj

Enter the variables : x,y,z

Enter first statement : Friend(Ram,x)

Enter second statement : Friend(y,z,Raj)
Mismatch in number of arguments
Unification not possible
```

If equal number of arguments are given and all other conditions are satisfied, unification is possible. OUTPUT->

```
In [9]: runfile('D:/CSE/5thsem/AI Lab/
Labtest2/1BM18CS147-prog.py', wdir='D:/CSE/5thsem/AI
Lab/Labtest2')

Enter the constants : Ram,Raj

Enter the variables : x,y,z

Enter first statement : Friend(Ram,x)

Enter second statement : Friend(y,Raj)
Unification is possible
Substitution: unifier θ= { y/Ram,x/Raj }
```

3) If the arguments are constants

Constants cannot be equated if they are different.
Unification is not possible. OUTPUT->

```
In [5]: runfile('D:/CSE/5thsem/AI Lab/
Labtest2/1BM18CS147-prog.py', wdir='D:/CSE/5thsem/AI
Lab/Labtest2')

Enter the constants : Tony,Steve

Enter the variables : x,y

Enter first statement : Enemy(Tony,x)

Enter second statement : Enemy(Steve,x)
constant cannot be equated
Unification is not possible
```

Same variable cannot have different substitution of constants. Unification is not possible. OUTPUT->

```
In [10]: runfile('D:/CSE/5thsem/AI Lab/
Labtest2/1BM18CS147-prog.py', wdir='D:/CSE/5thsem/AI
Lab/Labtest2')

Enter the constants : Tony,Steve

Enter the variables : x,y

Enter first statement : Enemy(Tony,x)

Enter second statement : Enemy(x,Steve)
Different subsitutions for same variable
Unification is not possible
```

If variables don’t have different constants substitution, constants are not getting equated and all other conditions are satisfied, Unification is possible. OUTPUT->

```
In [6]: runfile('D:/CSE/5thsem/AI Lab/
Labtest2/1BM18CS147-prog.py', wdir='D:/CSE/5thsem/AI
Lab/Labtest2')

Enter the constants : Tony,Steve

Enter the variables : x,y

Enter first statement : Enemy(Tony,x)

Enter second statement : Enemy(Tony,Steve)
Unification is possible
Substitution: unifier θ= { x/Steve }

In [7]:
```