28/12/20.

AI Lab Test-2.

Shaan Ram D.B
1BM18CS097.

```
import re.
def getAttributes(expression):
expression=expression.split("(")
expression="(".join(expression)
expression = expression[:-1]
expression =re.split("(?<=\)),(?!\))", expression)
return expression
def getInitialPredicate(expression):
return expression.split("(")
def isConstant(char):
return char.isupper() and len(char)==1.
def isVariable(char):
return char.islower() and len(char)==1
def replaceAttributes(exp,old,new):
attributes=getAttributes(exp)
for index, val, in enumerate(attributes):
if val==old:
attributes[index]=new
predicate=getInitialPredicate(exp)
return predicate+"("+",".join(attributes)+")"
def apply(exp, substitution):
for substitutions in substitutions:
new,old=substitutions
exp=replaceAttributes(exp,old,new)
return exp.
def checkOccurs(var,exp):
if exp.find(var)==-1:
return False.
return True.
```


Shaan Ram D.B

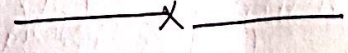
Shavan Ram D.B.
1BM18CS097.

```
def getFirstPart(expression):
attributes = getAttributes(expression)
return attributes[0]
def getRemainingPart(expression):
predicate = getInitialPredicate(expression)
attributes = getAttributes(expression)
newExpression = predicate + "(" + ",".join(attributes[1:]+")"
return newExpression.
def unify(exp1, exp2:
if exp1 == exp2:
return []
if isConstant(exp1) and isConstant(exp2):
if exp1 != exp2:
return False
if isConstant(exp1):
return [(exp1, exp2)]
if isConstant(exp2):
return [(exp2, exp1)]
if isVariable(exp1):
if checkOccurs(exp1, exp2):
return False
else:
return [(exp2, exp1)]
if isVariable(exp2):
if checkOccurs(exp2, exp1):
return False.
else:
return [(exp1, exp2)]
if getInitialPredicate(exp1)
!= getInitialPredicate(exp2):
```


Shavan Ram D.B.

28/12/20. AI-LabTest-2. Shaurav Ram D.B 1BM18CS097

```
Point("Predicates do not match. Cannot be unified")
return false.
attribute count 1 = len(getAttributes(exp1))
attribute count 2 = len(getAttributes(exp2))
if attribute count 1 != attribute count 2:
return False.
head 1 = getFirstPart(exp1)
head 2 = getFirstPart(exp2)
initial Substitutions = unify(head 1, head 2).
if not initial Substitution :
    return False.
exp1 = "knows(A, x)"
exp2 = "knows(y, y)"
Substitutions = unify(exp1, exp2)
print("Substitutions:")
print(substitutions)
exp1 = "knows(A, x)"
exp2 = "knows(y, mother(y))"
substitutions = unify(exp1, exp2)
print("substitutions:)"
print(Substitutions)
```



Shaurav Ram D.B.