AI Lab test-2

Vishala K. Shetkar
1BM18CS118.
28/12/2020

```
import re
def getAttributes (expression):
    expression = expression. split ("(")[1:]
    expression = expression [:-1]
    expression = re.split ("(?<!\(.),(?!.\))", expression)
    return expression.

def getInitialPredicate (expression):
        return expression. split ("(")[0]

def isconstant (char):
        return char.isupper () and len (char) == 1

def isvariable (char):
        return char. is lower () 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, substitutions):
    for substitutions in substitutions:
        new, old = substitution
    exp = replaceAttributes (exp, old, new)
    return exp
```

AI Lab test-2

Trishala.r. Phetkar
1BM18CS118
28/12/2020

```
def chickOccurs(var, exp):
  if exp.find(var)==-1:
      return false
   return true.

 def getFirstPart(expression):



 def getRemainingPart(expression):
     predicate= get Initial Predicate(expression)
     attributes =getAttributes (expression)
    newExpression= predicate :r "(" + "," .join attributes
                                          [1: ])+ ")"

    return newExpression
```

AI Lab test-2

Trishala K. Shetty
1BM18CS118
28/12/2020

```
def unify (exp1, exp2)

    if exp1 == exp2.

    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 is variable (exp2):
        if checkOccurs (exp2, exp1):
            return False
    else:
        return False

    else:
    return [(exp1, exp2)]
```

AI Lab test - 2

Sushala
1BM18CS118
28/12/2020

```
if get_InitialPredicate(exp1) != get_Initial Predicate(exp2):
    print("Predicates do not match. No unification")
    return False

attributeCount1 = len(get Attributes(exp1))
attributes Count 2 = len(get Attributes(exp2))
    if attributeCount1 != attributeCount2:
        return False

head 1 = get FirstPart(exp1)
head 2 = get First Part(exp2)
initialSubstitution = unify(head 1, head 2)
    if not initialSubstitution:
        return False
    if attributed 1 == 1:
        return Initial Substitution


tail1 = get Remaining Part(exp1)
tail2 = get Remoining Part(exp2)
if initial Substitution != []:
    tail 1 = apply(tail1, initial Substitution)
    tail 2 = apply(tail 2, initial Substitution)
remaining Substitution = Unify(tail 1, tail 2)
    if not remaining Substitution:
        return False

initial Substitution = Unify(tail 1, tail 2)
    if not remaining Substitution:
        return False
```

4

Shweta

AI Lab test - 2

Trishala R Shah
1BM18CS118
28/12/2020

```
initial substitution.extend (remaining substitution)
return initial substitution.


exp1 = "likes (Ram, y)"
exp2 = "likes (x, Raj)"
substitutions = unify (exp1, exp2)
print ("Substitutions : ")
print (substitutions)
```

```
Substitutions :
[('Ram, x'), ('Raj, y')].
```