Sakshi Srivastava
1BM18CS090.
B3 - Batch.

Program1

```
1) import re
def getAttributes(expression):
    expression = expression.split("(")[1:]
    u = "(".join(expression)
    u = expression.split(")")[:-1]
    '' = ")".join(expression)
    attributes = expression.split(',')
    return attributes

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

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

def isVariable(char):
    return char.islower() and len(char)==1
```

Scanned with CamScanner

```
def replaceAttributes(exp, old, new):
    attributes = getAttributes(exp)
    predicate = getInitialPredicate(exp)
    for index, val in enumerate(attributes):
        if val == old:
            attributes[index] = new
    return predicate + "(" + ", ".join(attributes) + ")"

def apply(exp, substitutions):
    for substitution in substitutions:
        new, old = substitution
        exp = replaceAttributes(exp, old, new)
    return exp

def checkOccurs(var, exp):
    if (exp.find(var) == -1):
        return False
    return True

def getFirstPart(expression):
    attributes = getAttributes(expression)
    return attributes[0]
```


Scanned with CamScanner

```
def getRenamingPart(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
    print(f"{exp1} and {exp2} are constant.
                Cannot be unified)
    return []

if isConstant(exp1)
    return ([exp1, exp2])
if isConstant(exp2)
    return ([exp2, exp1])
if isVariable(exp1)
    return [(exp2, exp1)] if not checkOccurs
                    (exp1, exp2) else []
```

Scanned with CamScanner

if is variable(exp2):
    return [(exp1, exp2) if not checkOccurs
            (exp2, exp1) else []

attribute count1 = len(getAttribute(exp1))
attribute count2 = len(getAttribute(exp2))

if attributeCount1 != attributeCount2:
    print("Length of attributes")
    return []

head1 = getFirstpart(exp1)
head2 = getFirstpart(exp2)

initial substitution = unify(head1, head2)

if not initial substitution:
    return []

if attribute count1 == 1:
    return initial substitution

tail1 = getRemainingPart(exp1)
tail2 = getRemainingPart(exp2)

if initial substitution != []:
    tail1 = apply()
    tail2 = apply()

remaining substitution = unify(tail1, tail2)

if not remaining substitution
    return [].

Scanned with CamScanner