AI Lab Test

Snehita. J
1BM18CS109
26/12/20

```
import re
def getAttributes(exp)
    exp = exp.split('(')[1:]
    exp = '('.join(exp)
    exp = exp.split(')')[:-1]
    exp = ')'.join(exp)
    att = exp.split(',')
    return att

def getInitialPredicate(exp):
    return exp.split('(',)[0]

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

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

def replaceAtt(exp, old, new):
    att = getAttributes(exp)
    pred = getInitialPredicate(exp)
    for i, val in enumerate(att):
        if val == old:
            att[i] = new
    return pred+'('+','.join(att)+')'

def apply(exp, subs):
    for sub in subs:
        new, old = sub
        exp = replaceAtt(exp, old, new)
    return exp
```


Snehita

Snehita. J
1BM18CS109
28/12/20

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

def getFirstPart(exp):
    att = getAttributes(exp)
    return att[0]

def getRemainingPart(exp):
    pred = getInitialPredicate(exp)
    att = getAttributes(exp)
    newExp = pred + "(" + ",".join(att[1:]) + ")"
    return newExp

def unify(exp1, exp2):
    if exp1 == exp2:
        return []
    if isConstant(exp1) and isConstant(exp2):
        if exp1 != exp2
            print(f"{exp1} and {exp2} are constants. 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 []
```

Snehita. J
1BM18CS109
28/12/20
AI- Lab- Test

```
attc1 = len(getAttributes(exp1))
attc2 = len(getAttributes(exp2))
if attc1 != attc2:
    print(f"length do not match. Cannot be unified")
    return []
h1 = getFirstPart(exp1)
h2 = getFirstPart(exp2)
initialSubstitution = unify(h1, h2)
if not initialSubstitution:
        return []
if attc1 == 1:
        return initialSubstition
t1 = getRemainingPart(exp1)
t2 = getRemainingPart(exp2)
if initialSubstition != []
        t1 = apply(t1, initialSubstitution)
        t2 = apply(t2, initialSubstition)
remainingSub = unify(t1, t2)
if not remainingSub:
        return []
return initalSubstition + remainingSub

def main()
    printf("Enter first exp")
    e1 = input()
```


Snehita. J

```
print("Enter second exp")
e2 = input()
substitutions = unify(e1, e2)
print("The substitutions are:")
print('\n'.join(substitution) for substitution in substitutions))

main()
```

Snehita.J
IBM18CS109
28/12/20
AI-lab-Test