AI lab Test 2

Date 29/12/20

Expt. No.

Ajay Mitter
1BM18CS006

Page No.

Program 1 -> Unification

Code:
```
def unify (exp1, exp2):
    if exp1 == exp2:
        return []
    elif isConstant(exp1) and isConstant(exp2):
        ~~return~~ if exp1 != exp2 :
                    return []
    elif isConstant (exp1):
        return [(exp1, exp2)]
    elif isConstant (exp2):
        return [(exp2, exp1)]
    elif isVariable (exp1):
        return [] if checkOccurs(exp1, exp2]
                  else [(exp2, exp1)]
    elif isVariable (exp2):
        return [] if checkOccurs(exp2, exp1)
                  else [(exp1, exp2)]
    elif getPredicate (exp1) != getPredicate(exp2):
        return []
    elif len(getattributes(exp1)) != len(attributes(exp2))
        return []

    firstAttr1 = getFirstAttributes(exp1)
    firstAttr2 = getFirstAttributes (exp2)
    initialSubs = unify (firstAttr1, firstAttr2)
    if not initialSubs:
        return []
```

Teacher's Signature :

AI lab test 2

Expt. No. .... Date 29/12/20

Ajay Mittr
1BM18CS006

Page No. ....

```
    if len(getAttributes(exp1)) == 1:
        return initialSubst

    remainingAttr1 = getRemaining(exp1)
    remainingAttr2 = getRemaining(exp2)
    if initialSubs != []:
        remainingAttr1 = apply(remainingAttr1, initialSub)
        remainingAttr2 = apply(remainingAttr2, initialSub)
    remainingSubs = unify(remainingAttr1, remainingAttr2)
    if not remainingSubs:
        return []
    return replaceNestedSubst(initialSub + remainingSub)

def getAttributes(exp):
    exp = exp.split('(')[1:]
    exp = '('.join(exp)
    exp = exp.split(')')[:-1]
    attributes = exp.split(', ')
    return attributes

def substitute(exp, old, new):
    attr = getAttributes(exp)
    pred = getPredicate(exp)
    for i, val in enumerate(attr):
        if val == old:
            attr[i] = new
    return pred + '(' + ','.join(attr) + ')'
```

Teacher's Signature : ____________

AI lab Test 2

Ajay Mittur
1BM18CS006

Date 29/12/20

Expt. No.

Page No.

```
def getPredicate(exp):
    return exp.split('(')[0]

def apply(exp, subs):
    for sub in subs:
        exp = substitute(exp, sub[0], sub[1])
    return exp.

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

def getfirstAttribute(exp):
    return getAttributes(exp)[0]

def getRemaining(exp):
    return getPredicate(exp) + '(' + ','.join(getAttributes(exp)
                                                [1:] + ')')

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

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

1) Knows(John, F(x)) and Knows(y, F(h(y)))
Ans: {y/John, x/h(John)}

2) ~~Student(x) and Teacher(y)~~
Ans: Not possible. diff. predicates

Teacher's Signature :