

LAB 7:

Program: Unification in First Order Logic

Code:

```
def unify(term1, term2, substitution=None):  
    if substitution is None:  
        substitution = {}  
    if term1 == term2:  
        return substitution  
    elif is_variable(term1):  
        return unify_variable(term1, term2, substitution)  
    elif is_variable(term2):  
        return unify_variable(term2, term1, substitution)  
    elif is_function(term1) and is_function(term2):  
        if term1[0] != term2[0]:  
            return None  
        else:  
            for arg1, arg2 in zip(term1[1:], term2[1:]):  
                substitution = unify(arg1, arg2, substitution)  
            if substitution is None:  
                return None  
            return substitution  
    else:  
        return None  
  
def unify_variable(variable, term, substitution):
```

```
if variable in substitution:
    return unify(substitution[variable], term, substitution)

if term == variable:
    return substitution

substitution[variable] = term

return substitution

def is_variable(term):
    return isinstance(term, str) and term.islower()

def is_function(term):
    return isinstance(term, tuple) and len(term) > 1

term1 = ("Loves", "x", "Mary")
term2 = ("Loves", "John", "Mary")

substitution = unify(term1, term2)

if substitution is not None:
    print("Unification successful, substitution:", substitution)
else:
    print("Unification failed")
```

20/1/20
Tuesday

Lab-7

(30)

Implement the unification algorithm in first order logic

Algorithm:

Function unify (x, y , Substitution):

if Substitution is None:

Substitution = {}

if $x == y$:

return Substitution

if x is a variable:

if x is in Substitution:

return unify (Substitution [x], y , Substitution)

if occur_check (x, y):

raise unificationError ("Occur check fails: " + x +

cannot be unified with " + y + ")

Substitution [x] = y

return Substitution

if y is a variable:

if y is in Substitution:

return unify (x , Substitution [y], Substitution)

if occur_check (y, x):

raise unificationError ("Occur check fails: "

+ y + " cannot be unified with " + x + ")

Substitution [y] = x

return Substitution

if x and y are function-terms:

if length (x) != length (y):

raise unificationError ("Cannot unify " + x +
with " + y + ": different arities")

for each pair of terms (t_1, t_2) in Zip (x, y)

Substitution = unify (t_1, t_2 , Substitution)

return Substitution

raise unificationError ("Cannot unify
" + x + " with " + y + ")

Output:

unification failed: cannot unify x with $f(x)$
 unification failed: cannot unify x with John

