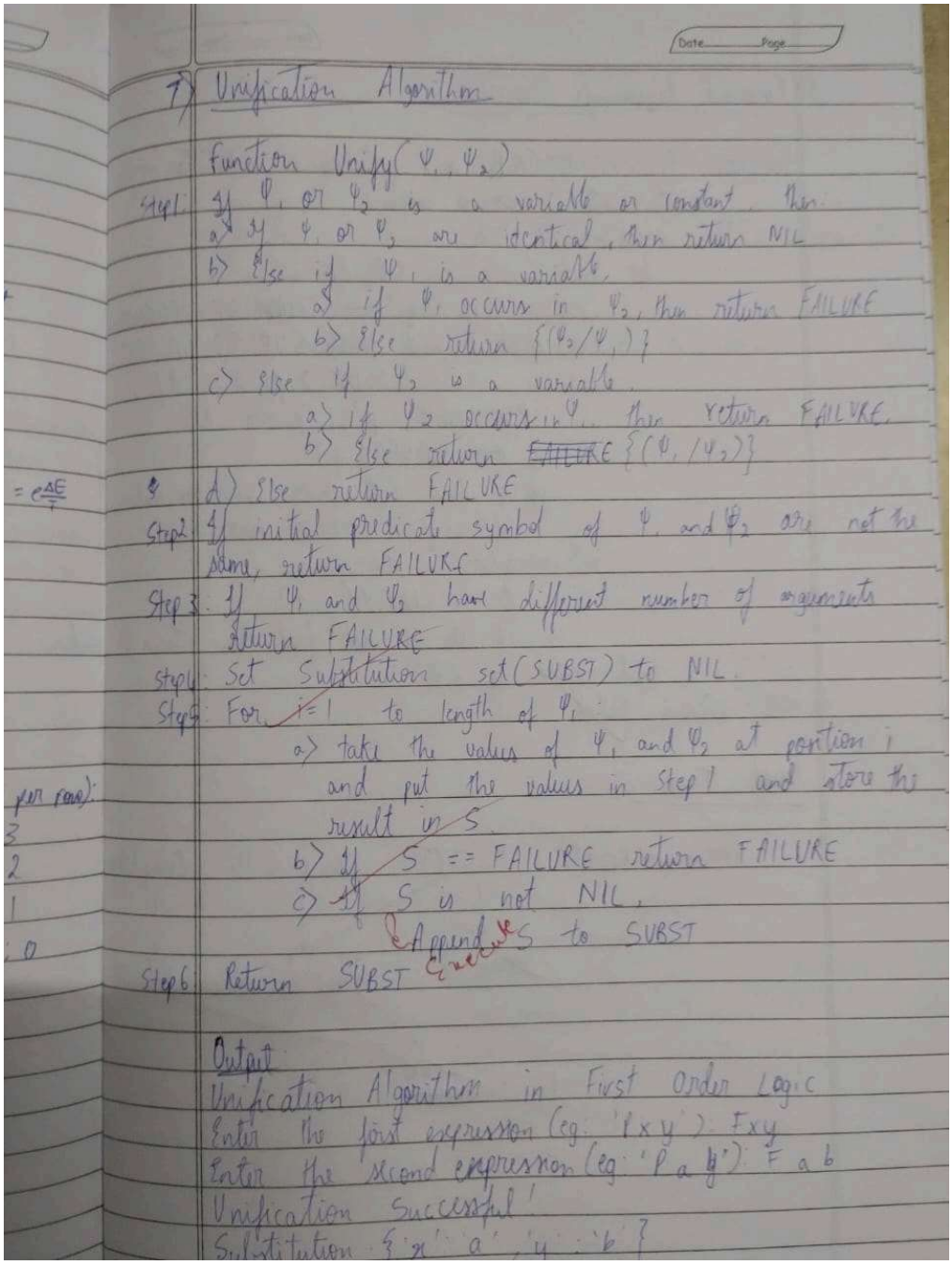
Unification Algorithm:



1) Unification Algorithm

Function Unify(ψ_1, ψ_2)

Step1: If ψ_1 or ψ_2 is a variable or constant, then:
a) If ψ_1 or ψ_2 are identical, then return NIL
b) Else if ψ_1 is a variable,
   a) if ψ_1 occurs in ψ_2, then return FAILURE
   b) Else return $\{(\psi_2/\psi_1)\}$
c) Else if ψ_2 is a variable,
   a) if ψ_2 occurs in ψ_1 then return FAILURE.
   b) Else return $\{(\psi_1/\psi_2)\}$
d) Else return FAILURE

Step2: If initial predicate symbol of ψ_1 and ψ_2 are not the same, return FAILURE

Step3: If ψ_1 and ψ_2 have different number of arguments return FAILURE

Step4: Set Substitution set(SUBST) to NIL.

Step5: For i=1 to length of ψ_1:
   a) take the values of ψ_1 and ψ_2 at position i and put the values in Step1 and store the result in S.
   b) If S == FAILURE return FAILURE
   c) If S is not NIL.
      Append S to SUBST

Step6: Return SUBST

Output:
Unification Algorithm in First Order Logic
Enter the first expression (eg: 'P x y'): P x y
Enter the second expression (eg: 'P a b'): P a b
Unification Successful!
Substitution: {'x': 'a', 'y': 'b'}

Code:

```
def unify(s1, s2, theta={}):

    if theta is None:
        return None

    if s1 == s2:
        return theta

    if isinstance(s1, str) and s1.islower():
        return unify_var(s1, s2, theta)

    if isinstance(s2, str) and s2.islower():
        return unify_var(s2, s1, theta)

    if isinstance(s1, tuple) and isinstance(s2, tuple) and len(s1) == len(s2):
        return unify(s1[1:], s2[1:], unify(s1[0], s2[0], theta))

    return None

def unify_var(var, x, theta):
    if var in theta:
        return unify(theta[var], x, theta)
    elif x in theta:
        return unify(var, theta[x], theta)
    elif occurs_check(var, x, theta):
        return None
    else:
        theta[var] = x
        return theta

def occurs_check(var, x, theta):
    if var == x:
        return True
    elif isinstance(x, str) and x.islower() and x in theta:
        return occurs_check(var, theta[x], theta)
    elif isinstance(x, tuple):
        for arg in x:
            if occurs_check(var, arg, theta):
                return True
    return False


s1 = ('p', 'x', ('f', 'x'), ('y'))
s2 = ('p', 'a', 'y', ('f', 'x'))

substitution = unify(s1, s2)

if substitution:
```

```
    print("Unification successful:")
    print(f"Substitution: {substitution}")
else:
    print("Unification failed.")
```

Output:
Unification successful:

Substitution: {'x': 'a', 'y': ('f', 'x')}