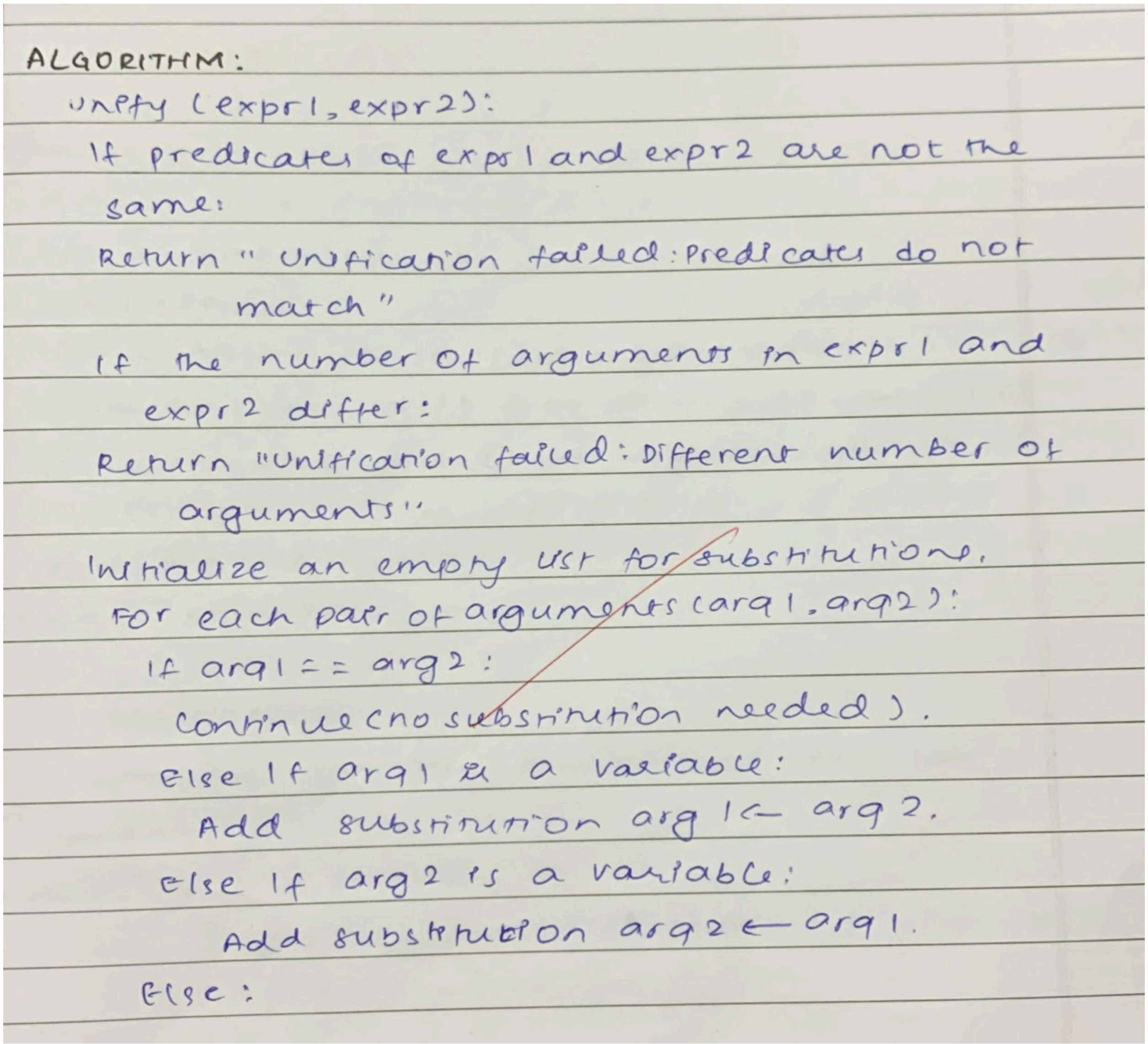
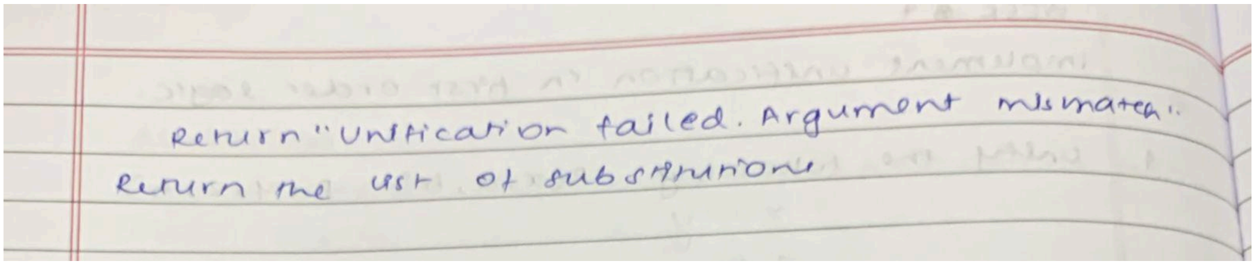
03-12-2024 WEEK 7

UNIFICATION IN FIRST ORDER LOGIC

ALGORITHM-



```
ALGORITHM:
  unify (expr1, expr2):
   If predicates of expr1 and expr2 are not the
    same:
   Return " Unification failed: Predicates do not
          match"
   If the number of arguments in expr1 and
     expr2 differ:
   Return "Unification failed: Different number of
      arguments"
   Initialize an empty list for substitutions.
    For each pair of arguments (arg1, arg2):
      If arg1 == arg2:
        Continue (no substitution needed).
      Else If arg1 is a variable:
        Add substitution arg1 ← arg2.
      Else If arg2 is a variable:
         Add substitution arg2 ← arg1.
     Else:
```

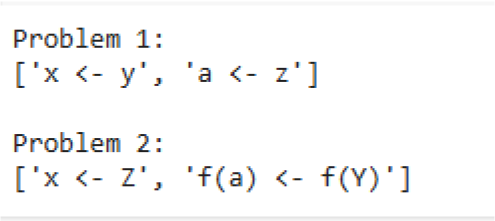


```
      Return "Unification failed. Argument mismatch".
   Return the list of substitutions
```

CODE-

```
def unify(expr1, expr2):
    substitutions = []
    if expr1[0] != expr2[0]:
        return "Unification failed: Predicates do not match."
    args1 = expr1[1:]
    args2 = expr2[1:]

    if len(args1) != len(args2):
        return "Unification failed: Different number of arguments."

    for i, (arg1, arg2) in enumerate(zip(args1, args2)):
        if arg1 == arg2:
        elif isinstance(arg1, str) and arg1.islower():  # arg1 is a variable
            substitutions.append(f"{arg1} <- {arg2}")
        elif isinstance(arg2, str) and arg2.islower():  # arg2 is a variable
            substitutions.append(f"{arg2} <- {arg1}")
        else:
            return f"Unification failed: Argument mismatch at position {i + 1}."

    return substitutions or "No substitutions needed."

expr1_1 = ("P", "x", "a", "b")
expr1_2 = ("P", "y", "z", "b")
expr2_1 = ("P", "x", "f(Y)")
expr2_2 = ("P", "Z", "f(a)")
print("Problem 1:")
result1 = unify(expr1_1, expr1_2)
print(result1)
print("\nProblem 2:")
result2 = unify(expr2_1, expr2_2)
print(result2)
```

OUTPUT-



```
Problem 1:
['x <- y', 'a <- z']

Problem 2:
['x <- Z', 'f(a) <- f(Y)']
```