Міністерство освіти та науки України

Львівський національний університет імені Івана Франка

Факультет електроніки та

Комп’ютерних технологій

**Звіт**

Про виконання лабораторної роботи №4

“ Композиція нечітких бінарних відношень”

Виконав:

Студент групи ФеІ-44

Сапанюк М.І.

Перевірила:

Притула М.

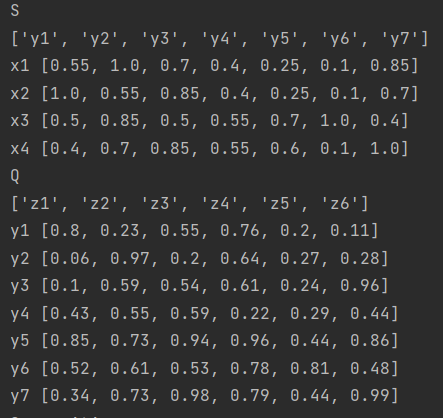
Львів 2022

**Мета:**

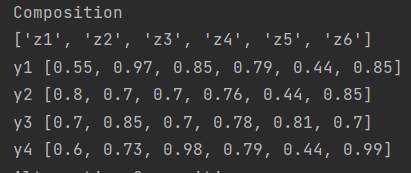
ознайомитися з основними операціями над нечіткими відношеннями, застосувати операцію композиції бінарних нечітких відношень у практичних задачах.

**Хід роботи:**

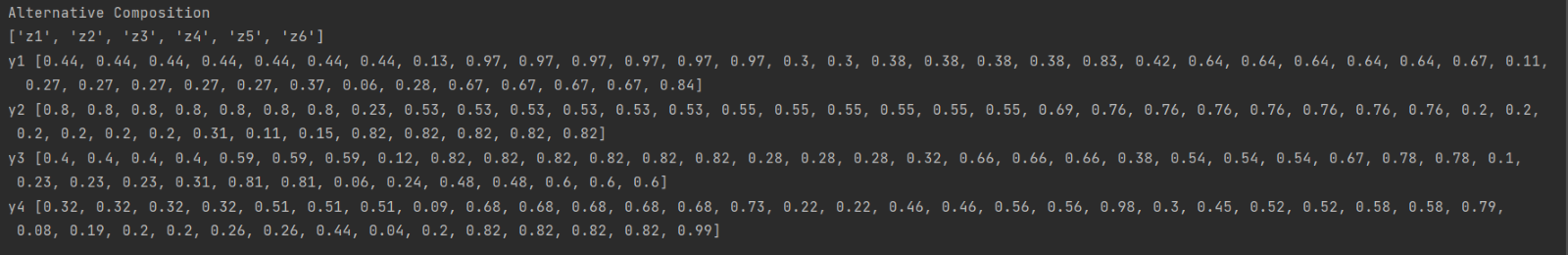
1. Застосувати операцію композиції бінарних нечітких відношень у задачі, пов’язаній з консалтингом в області вибору професії, щодо навчання та отримання відповідної спеціальності. Для цього на основі психо-фізіологічного профілювання спеціальностей, яке визначається нечітким бінарним відношенням S = {, µS()}, заданим на декартовому добутку Х × Y, а також психо-фізіологічних характеристик претендентів на ці спеціальності, яке визначаються нечітким бінарним відношенням Q = {, µQ()}, заданим на декартовому добутку Y × Z, визначити оптимальні рекомендації для вибору професій. Х – множина спеціальностей: х1 – менеджер; х2 – програміст; х3 – водій; х4 – секретар. Y – множина характеристик: у1 – гнучкість мислення; у2 – уміння швидко приймати рішення; у3 – концентрація уваги; у4 – зорова пам’ять; у5 – витривалість; у6 – швидкість реакції рухів; у7 – відповідальність. Z – множина претендентів: z1 – Андрієнко; z2 – Василенко; z3 – Іваненко; z4 – Дмитренко; z5 – Петренко; z6 – Романенко.   
2. На основі інформації експертів задати нечіткі відношення S = {, µS()} і Q = {, µQ()} у вигляді матриці профілювання спеціальностей МS, розміром 4×7, і матриці профілювання претендентів МQ, розміром 7×6.



3. Запрограмувати алгоритм, який за допомогою операції композиції нечітких відношень S ÄQ знаходить переваги і недоліки претендентів на кожну зі спеціальностей



4. Визначити переваги і недоліки претендентів на кожну зі спеціальностей за допомогою альтернативної операції композиції бінарних нечітких відношень. Порівняти результати.



**Висновок:**

Виконавши лабораторну роботу, ознайомився з основними операціями над нечіткими відношеннями, застосував операцію композиції бінарних нечітких відношень у практичних задачах.

**Додаток:**

from random import random as rand  
# X  
# х1 – менеджер;   
# х2 – програміст;   
# х3 – водій;   
# х4 – секретар.   
# Y  
# у1 – гнучкість мислення;   
# у2 – уміння швидко приймати рішення;   
# у3 – концентрація уваги;   
# у4 – зорова пам’ять;   
# у5 – витривалість;   
# у6 – швидкість реакції рухів;   
# у7 – відповідальність. -   
# Z  
# z1 – Андрієнко;   
# z2 – Василенко;   
# z3 – Іваненко;   
# z4 – Дмитренко;   
# z5 – Петренко;   
# z6 – Романенко.   
from typing import List  
  
  
def printMarrix(matrix: List[List[float]], column\_label: List[str], row\_label:  
List[str]) -> None:  
 print(column\_label)  
 for i in range(0, len(matrix)):  
 print(row\_label[i], matrix[i])  
  
  
def alternative\_composition(matrix1: List[List[float]], matrix2:  
List[List[float]]):  
 answer: List[List[float]] = []  
 for i in range(0, len(matrix1)):  
 answer.append([])  
 for j in range(0, len(matrix2[0])):  
 temp: List[float] = []  
 for k in range(0, len(matrix1[0])):  
 temp.append(matrix1[i][k] \* matrix2[k][j])  
 # answer[i].append(max(temp))  
 answer[i].append(float(format(max(temp), '.2f')))  
 return answer  
  
  
def composition(matrix1: List[List[float]], matrix2: List[List[float]]):  
 answer: List[List[float]] = []  
 for i in range(0, len(matrix1)):  
 answer.append([])  
 for j in range(0, len(matrix2[0])):  
 temp: List[float] = []  
 for k in range(0, len(matrix1[0])):  
 temp.append(min(matrix1[i][k], matrix2[k][j]))  
 # answer[i].append(max(temp))  
 answer[i].append(float(format(max(temp), '.2f')))  
 return answer  
  
  
def main() -> int:  
 x\_labels: List[str] = ['x1', 'x2', 'x3', 'x4']  
 y\_labels: List[str] = ['y1', 'y2', 'y3', 'y4', 'y5', 'y6', 'y7']  
 z\_labels: List[str] = ['z1', 'z2', 'z3', 'z4', 'z5', 'z6']  
 # y1 y2 y3 y4 y5 y6 y7  
 S: List[List[float]] = [[0.55, 1.00, 0.70, 0.40, 0.25, 0.10, 0.85], # х1  
 [1.00, 0.55, 0.85, 0.40, 0.25, 0.10, 0.70], # х2  
 [0.50, 0.85, 0.50, 0.55, 0.70, 1.00, 0.40], # x3  
 [0.40, 0.70, 0.85, 0.55, 0.60, 0.10, 1.00]] # x4  
 # # z1 z2 z3 z4 z5 z6  
 # Q: list[list[float]] = [[0.00, 0.00, 0.00, 0.00, 0.00, 0.00], #y1  
 # [0.00, 0.00, 0.00, 0.00, 0.00, 0.00], #y2  
 # [0.00, 0.00, 0.00, 0.00, 0.00, 0.00], #y3  
 # [0.00, 0.00, 0.00, 0.00, 0.00, 0.00], #y4  
 # [0.00, 0.00, 0.00, 0.00, 0.00, 0.00], #y5  
 # [0.00, 0.00, 0.00, 0.00, 0.00, 0.00], #y6  
 # [0.00, 0.00, 0.00, 0.00, 0.00, 0.00]] #y7  
 Q: List[List[float]] = [[float(format(rand(), '.2f')) for i in range(0, 6)]  
 for j in range(0, 7)]  
 print("S")  
 printMarrix(S, y\_labels, x\_labels)  
 print("Q")  
 printMarrix(Q, z\_labels, y\_labels)  
 print("Composition")  
 printMarrix(composition(S, Q), z\_labels, y\_labels)  
 print("Alternative Composition")  
 printMarrix(alternative\_composition(S, Q), z\_labels, y\_labels)  
 composition(S, Q)  
 # printMarrix(S, z\_labels, x\_labels)  
 return 0  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()