

Міністерство освіти та науки України
Львівський національний університет імені Івана Франка

Факультет електроніки та
Комп'ютерних технологій

Звіт

Про виконання лабораторної роботи №6
“Оптимальні альтернативи та бінарні відношення”

Виконав:
Студент групи ФеІ-44
Сапанюк М.І.
Перевірила:
Мостова М. Р.

Львів 2022

Мета:

Навчитися знаходити відношення байдужості, домінування, непорівняльності, а також максимуми, мінімуми, мажоранти та міноранти.

Хід роботи:

1. Титульний лист, тему, мету, завдання
2. Письмові відповіді на контрольні запитання:
 - а. Дати означення мажоранти, міноранти, максимуму, мінімуму.
 - б. Дати означення відношення байдужості, домінування, непорівняльності.
 - в. Чим відрізняються поняття мажоранти та міноранти від понять мінімуму та максимуму.
4. Написати програму, яка для заданого бінарного відношення будує відношення байдужості, домінування, непорівняльності.

1 0 0 1 1

0 1 1 0 1

1 0 1 0 0

0 1 0 1 1

1 1 0 0 1

5. Написати програму, яка для заданого бінарного відношення шукає максимуми, мінімуми, мажоранти, міноранти

1 0 0 1 1

1 1 1 1 1

0 0 0 1 0

1 0 1 1 0

1 1 0 1 0

6. Виконати завдання 4,5 письмово у зошиті. Результат виконання надіслати на корпоративну адресу викладача.

7. У висновках описати алгоритми Ваших програми та зобразити блок схему.

Виконання завдання:

Контрантні запитання:

1. Маторант (мінорант) підмножини M M' впорядкованої множини M - це такий елемент $M_i(M_j)$ що належить M , якщо M_i це належить M виконуватиметься $m_i = m_j$ ($m_i \leq m_j$). Якщо маторант (мінорант) належить підмножині M , тоді вона наз. максимум (мінімум) елементами підмножини M' впорядкованої множини M .
2. Байдутність наз. таке відношення R , яке водночас водночас нерівності та симетричності.

Домінують наз. таке відношення, яке водночас є антисиметричності та антисиметричності.

Непорівнянності наз. пара відношень a та b , якщо вони не є байдутними і жоден з них не є менше ніж іншим.

3. Максимум та мінімум визначаються від маторантних та мінорантних тим, що максимум та мінімум шукатимуть для заданої матриці, а маторант та мінорант - для матриці доповнення.

Матрица

$$R = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix} \quad R^{-1} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Визначення дотичності:

$$R_s = R \cap R^{-1}$$

$$R_s = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Визначення ганкельованості:

$$R_D = R \setminus R_s$$

$$R_D = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad R_D^{-1} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

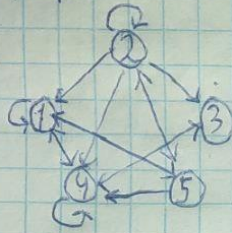
Визначення некорівності:

$$R^N = R_D \cup R_s \cup R_D^{-1}$$

$$R_D \cup R_s = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix}; \quad R_D \cup R_s \cup R_D^{-1} = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \end{pmatrix}$$

$$R = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Граф:



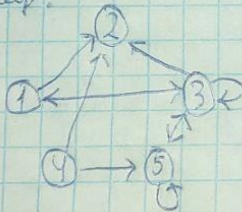
	b_i	b_{ix}
1	4	3
2	2	5
3	2	1
4	5	3
5	2	3

$$A^+(R) = \{x_2\}$$

$$A^-(R) = \{x_5\}$$

$$\bar{R} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Граф:



	b_i	b_{ix}
1	1	2
2	3	0
3	3	4
4	0	2
5	5	2

Мажоранта - $\{x_2, x_3, x_5\}$

Міноранта - $\{x_3\}$

Завдання №4

Вхідна матриця:

```
1 0 0 1 1
0 1 1 0 1
1 0 1 0 0
0 1 0 1 1
1 1 0 0 1
```

Байдужості:

```
1 0 0 0 1
0 1 0 0 1
0 0 1 0 0
0 0 0 1 0
1 1 0 0 1
```

Домінування:

```
0 0 0 1 0
0 0 1 0 0
1 0 0 0 0
0 1 0 0 1
0 0 0 0 0
```

Непорівняності:

```
1 0 1 1 1
0 1 1 1 1
1 1 1 0 0
1 1 0 1 1
1 1 0 1 1
```

Завдання №5

Вхідна матриця:

```
1 0 0 1 1
1 1 1 1 1
0 0 0 1 0
1 0 1 1 0
1 1 0 1 0
```

Мажоранти:

$x_2, x_3, x_5,$

Максимум:

$x_2,$

Міноранти:

$x_3,$

Мінімум:

$x_4,$

Висновок:

На цій лабораторній роботі я навчився знаходити відношення байдужості, домінування, непорівняльності, а також максимуми, мінімуми, мажоранти та міноранти.

Додаток:

```
import os
```

```
script_dir = os.path.dirname(__file__) # <-- absolute dir the script is in
rel_path_1 = "matrix1.txt"
rel_path_2 = "matrix2.txt"
path_to_file_1 = os.path.join(script_dir, rel_path_1)
path_to_file_2 = os.path.join(script_dir, rel_path_2)
```

```
def transpose_marix(matr1):
    length1 = len(matr1)
    matr = [[0 for k in range(len(matr1[0]))] for n in range(length1)]
    for n in range(length1):
        for k in range(len(matr1[0])):
            matr[k][n] = matr1[n][k]
    return matr
```

```
def find_reverse(matr):
    length = len(matr)
    RES = [[0 for k in range(len(matr[0]))] for n in range(length)]
    for n in range(length):
        for k in range(len(matr[0])):
            RES[n][k] = True if matr[n][k] == False else False
    return RES
```

```
def find_max(matr):
    majorants = []
    for e in matr:
        majorants.append(sum(e))
    mx = max(majorants)
    return majorants, mx
```

```
def find_min(matr):
    matr = transpose_marix(matr)
    majorants = []
    for e in matr:
        majorants.append(sum(e))
    mx = max(majorants)
    return majorants, mx
```

```
def find_majorants(matr):
    matr = transpose_marix(find_reverse(matr))
    majorants = []
    for e in matr:
        majorants.append(sum(e))
    return majorants, max(majorants)
```

```
def find_minorants(matr):
    matr = find_reverse(matr)
    minorants = []
    for e in matr:
        minorants.append(sum(e))
```

```

mx = max(minorants)
return minorants, mx

def print_out_matrix(matr):
    for e in matr:
        for e2 in e:
            print(e2 * 1, end=" ")
        print()

def interseccion(A, B):
    lenght = len(A)
    RES = [[0 for k in range(len(A[0]))] for n in range(lenght)]
    for n in range(lenght):
        for k in range(len(A[0])):
            RES[n][k] = A[n][k] & B[n][k]
    return RES

def union(A, B):
    lenght = len(A)
    RES = [[0 for k in range(len(A[0]))] for n in range(lenght)]
    for n in range(lenght):
        for k in range(len(A[0])):
            RES[n][k] = B[n][k] or A[n][k]
    return RES

def tolerance(matrix):
    return interseccion(matrix, transpose_marix(matrix))

def incomparability(matrD, matrB):
    temp1 = union(matrD, matrB)
    temp2 = union(temp1, transpose_marix(matrD))
    return temp2

def difference(A, B):
    length = len(A)
    RES = [[0 for k in range(len(A[0]))] for n in range(length)]
    for n in range(length):
        for k in range(len(A[0])):
            RES[n][k] = A[n][k] & (1 if B[n][k] == 0 else 0)
    return RES

# 1
print("Завдання №4")
with open(path_to_file_1, 'r') as file:
    matr = [list(map(lambda x: True if x == '1' else False, line.replace("\n", "").split())) for line in
              file.readlines()]
print("Вхідна матриця:")
print_out_matrix(matr)
print("\nБайдужості:")
tol = tolerance(matr)
print_out_matrix(tol)
print("\nДомінування:")
diff = difference(matr, tol)
print_out_matrix(diff)
print("\nНепорівняності:")
inc = incomparability(diff, tol)
print_out_matrix(inc)

```

```
# 2
print("\nЗавдання №5")
with open(path_to_file_2, 'r') as file:
    matr = [list(map(lambda x: True if x == '1' else False, line.replace("\n", "").split())) for line in
              file.readlines()]
print("Вхідна матриця:")
print_out_matrix(matr)
print()
majorants, mx = find_majorants(matr)
print("Мажоранти:")
for i in range(len(majorants)):
    if majorants[i] == mx:
        print("x" + str(i + 1), end=" ")
print("\n")
minorants, mx2 = find_minorants(matr)
print("Міноранти:")
for i in range(len(minorants)):
    if (minorants[i] == mx2):
        print("x" + str(i + 1), end=" ")
print("\n")
print("Максимум: ")
majorants, mx = find_max(matr)
for i in range(len(minorants)):
    if (majorants[i] == mx):
        print("x" + str(i + 1), end=" ")
print("\n")
print("Мінімум: ")
minorants, mx2 = find_min(matr)
for i in range(len(minorants)):
    if (minorants[i] == mx2):
        print("x" + str(i + 1), end=" ")
print("\n")
```