

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТУ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”**

Кафедра систем штучного інтелекту

Лабораторна робота №3
з дисципліни «Дискретна математика»

Виконав:
студент групи КН-115
Сирватка Максим
Викладач:
Мельникова Н.І.

Львів – 2019 р.

Тема: Побудова матриці бінарного відношення

Мета роботи: набуття практичних вмінь та навичок при побудові матриць бінарних відношень та визначені їх типів.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Декартів добуток множин A і B (позначається $A \times B$) – це множина всіх упорядкованих пар елементів (a, b) , де $a \in A, b \in B$. При цьому вважається, що $(a_1, b_1) = (a_2, b_2)$ тоді і тільки тоді, коли $a_1 = a_2, b_1 = b_2$.

Потужність декартового добутку дорівнює: $|A \times B| = |A| \times |B|$.

Бінарним відношенням R називається підмножина декартового добутку $A \times B : R \subset A \times B$.

Якщо пара (a, b) належить відношенню R, то пишуть: $(a, b) \in R$, або aRb .

Областю визначення бінарного відношення $R \subset X \times Y$ називається множина:

$$\delta_R = \{x \mid \exists y (x, y) \in R\}$$

Область значень - множина:

$$\rho_R = \{y \mid \exists x (x, y) \in R\}$$

Для скінчених множин бінарне відношення $R \subset A \times B$ зручно задавати за допомогою **матриці відношення**:

$$R_{m \times n} = (r_{ij}), \quad m = |A|, \quad n = |B|$$

Елементами матриці є значення:

$$r_{ij} = 1, \text{ якщо } ((a_i, b_j) \in R$$

$$r_{ij} = 0, \text{ якщо } ((a_i, b_j) \notin R$$

Властивості бінарних відношень

1. **Рефлексивність**, якщо для всіх $a \in M$ має місце aRa . Бінарне відношення R, визначене на деякій множині і відрізняється тим, що для будь-якого x цієї множини елемент x знаходиться у відношенні R до самого себе, тобто для будь-якого елемента x цієї множини має місце xRx .
2. **Антирефлексивність**, якщо для жодного $a \in M$ не виконується aRa . Відзначимо, що так само, як антисиметричність не збігається з несиметричною, антирефлексивність не збігається з нерефлексивністю. Двомісне відношення R, визначене на деякій множині M і відрізняється тим, що для будь-якого елемента x цієї множини не виконується, що воно знаходиться у відношенні R до самого себе (відсутнє xRx), тобто можливий випадок, що елемент множини не знаходиться у відношенні R до самого себе.
3. **Симетричність**, якщо для всіх $a, b \in M$ таких, що aRb виконується bRa . Бінарне відношення R, визначене на деякій множині і відрізняється тим, що для будь-яких

елементів x і y цієї множини з того, що x знаходиться до y відносно R (xRy), випливає, що і y знаходиться в тому ж відношенні до x (yRx).

4. **Асиметричність**, якщо для всіх $a, b \in M$ таких, що aRb не виконується bRa . Бінарне відношення R , визначене на деякій множині і відрізняється тим, що для будь-яких x та y з xRy слідує заперечення yRx .

5. **Антисиметричність**, якщо для всіх $a, b \in M$ таких, що aRb і bRa маємо $a = b$. Бінарне відношення R , визначене на деякій множині і відрізняється тим, що для будь-яких x та y з xRy і yRx — 1у слід $x = y$ (тобто R і R^{-1} виконуються одночасно лише для рівних між собою членів).

6. **Транзитивність**, якщо зі співвідношень aRb і bRc випливає aRc . Бінарне відношення R , визначене на деякій множині і відрізняється тим, що для будь-яких x, y, z цієї множини з xRy і yRz слід xRz ($xRy \& yRz \setminus \text{то } xRz$).

7. **Нетранзитивність** — бінарне відношення R , визначене на деякій множині і відрізняється тим, що для будь-яких x, y, z цієї множини з xRy і yRz не слідує xRz .

Функціональне відношення

Функцією з множини X на множину Y називається всюди визначена бінарна відповідність, при якому кожен елемент множини X зв'язаний з єдиним елементом множини Y .

Функція **записується наступним чином**: якщо $f \subseteq X \times Y$, то $f : X \rightarrow Y$.

Множину X називають **областю визначення**, а Y — **множиною значень** функції.

Областю значень функції називається підмножина Y , яка складається з образів всіх елементів $x \in X$. Вона **позначається символом $f(x)$** .

Оскільки для кожного $x \in X$ існує єдиним образом визначений $y \in Y$, такий що $(x, y) \in f$, то записують $y = f(x)$ та говорять, що **функція f відображує множину X на множину Y** , а $f(x)$ називають **образом x при відображенні f або значенням функції**, яка відповідає аргументу x .

Види функціональних відношень

1. Функція називається **ін'єктивною (ін'єкцією)**, якщо з умови $f(x_1) = f(x_2)$ слідує, що $x_1 = x_2$ для будь-яких $x_1, x_2 \in X$. Функція ін'єктивна тоді і тільки тоді, коли для будь-яких $x_1, x_2 \in X$ якщо $x_1 \neq x_2$, то $f(x_1) \neq f(x_2)$, тобто **для різних аргументів x функція f приймає різні значення**.

2. Функція називається **сюр'єктивною (сюр'єкцією)**, якщо для кожного $y^* \in Y$ знайдеться такий $x^* \in X$, що $y^* = f(x^*)$.

3. Функція називається **бієктивною (бієкцією)**, якщо вона ін'єктивна та сюр'єктивна одночасно. Таку функцію ще називають **взаємно-однозначним відображенням**.

ІНДИВІДУАЛЬНІ ЗАВДАННЯ

Варіант 15

Завдання №1. Використовуючи теоретичні відомості, розв'язати наступні задачі за своїм варіантом:

1. Чи є вірною рівність: $(A \times (B \cap C)) \cap ((A \cap B) \times C) = (A \times C) \cap (B \times B)$?

Розв'язання:

$$(A \times (B \cap C)) \cap ((A \cap B) \times C) = (A \times C) \cap (B \times B);$$

$$((A \times B) \cap (A \times C)) \cap ((A \times C) \cap (B \times C)) = (A \times C) \cap (B \times B);$$

$$(A \times B) \cap (A \times C) \cap (B \times C) = (A \times C) \cap (B \times B);$$

$$\text{Нехай } (x, y) \in (A \times B) \cap (A \times C) \cap (B \times C) \Leftrightarrow$$

$$(x, y) \in (A \times B) \ \& \ (x, y) \in (A \times C) \ \& \ (x, y) \in (B \times C) \Leftrightarrow$$

$$(x \in A \ \& \ y \in B) \ \& \ (x \in A \ \& \ y \in C) \ \& \ (x \in B \ \& \ y \in C) \Leftrightarrow$$

$$(x \in A \ \& \ x \in A) \ \& \ (y \in C \ \& \ y \in C) \ \& \ (x \in B \ \& \ y \in B) \Leftrightarrow$$

$$(x, y) \in (A \times C) \ \& \ (x, y) \in (B \times B) \Leftrightarrow$$

$$(x, y) \in ((A \times C) \cap (B \times B)).$$

Отже, дана рівність є **вірною**.

2. Знайти матрицю відношення $R \subset M \times 2^M$, де $M = \{1, 2, 3\}$:

$$R = \{(x, y) \mid x \in M \ \& \ y \subset M \ \& \ |y| \leq x\}$$

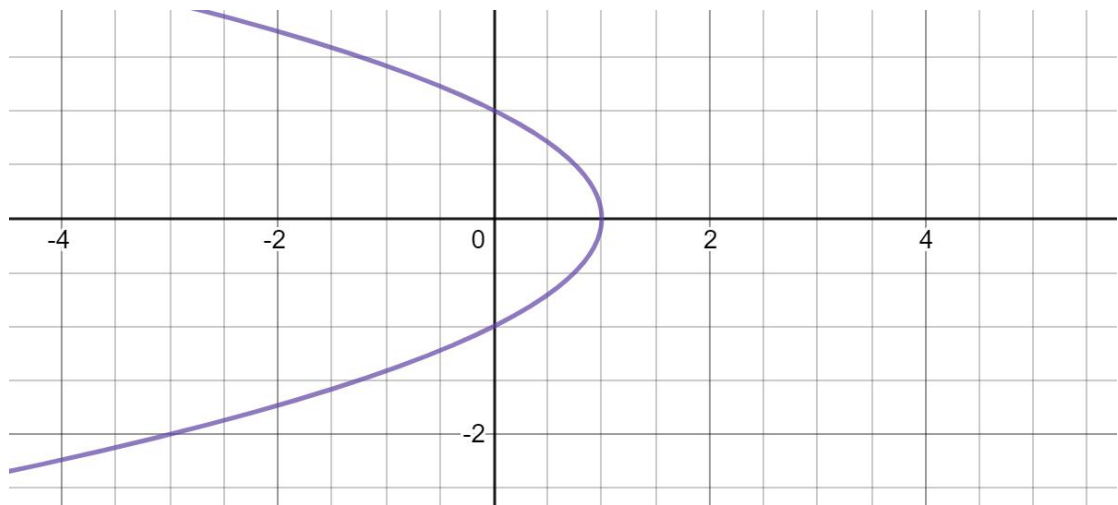
Розв'язання:

	\emptyset	$\{1\}$	$\{2\}$	$\{3\}$	$\{1, 2\}$	$\{2, 3\}$	$\{1, 3\}$	$\{1, 2, 3\}$
1	1	1	1	1	0	0	0	0
2	1	1	1	1	1	1	1	0
3	1	1	1	1	1	1	1	1

3. Зобразити відношення графічно:

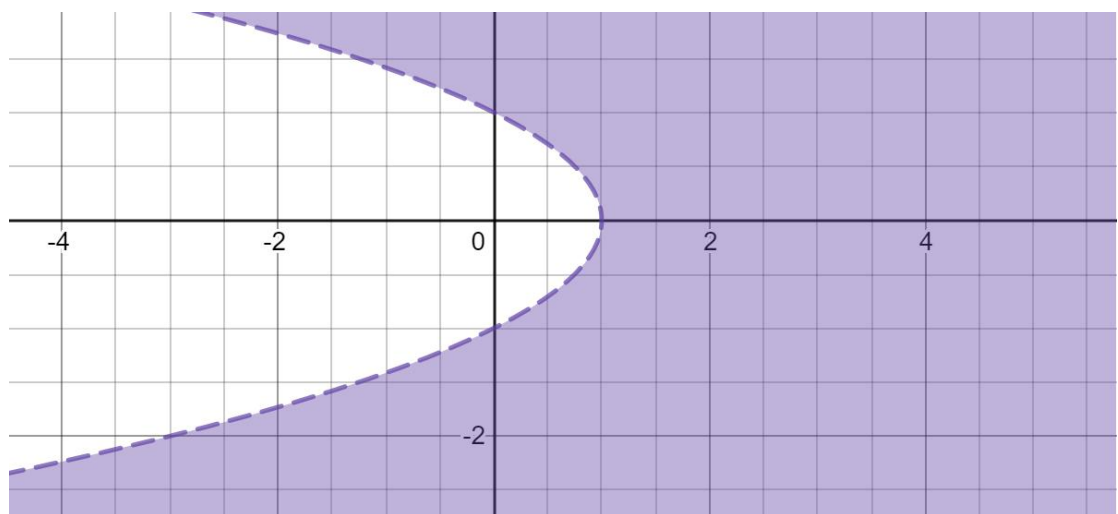
$$\alpha = \{(x, y) \mid (x, y) \in R^2 \ \& \ x + y^2 - 1 > 0\}, \text{ де } R - \text{множина дійсних чисел.}$$

Розв'язання: побудуємо графік функції $x + y^2 - 1 = 0$; $y^2 = 1 - x$.



Тепер зобразимо на координатній площині $x + y^2 - 1 > 0$. Даному відношенню відповідає частина площини, зображена на рисунку нижче.

Область визначення : $\delta = R$; область значень : $\rho = R$.



4. Навести приклад бінарного відношення $R \subset A \times A$, де $A = \{a, b, c, d, e\}$, яке є антирефлексивне, несиметричне, транзитивне, та побудувати його матрицю.

Розв'язання: бінарне відношення: $R = (a,b), (a,c), (a,d), (a,e), (c,b), (d,b), (d,c), (d,e), (e,b), (e,c)$.

Матриця бінарного відношення:

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

Дане відношення є **антирефлексивне, несиметричне та транзитивне**.

5. Визначити множину (якщо це можливо), на якій дане відношення є: а) функціональним; б) бієктивним:

$$\alpha = \{(x, y) \mid (x, y) \in R^2 \ \& \ y = x + |x|\}.$$

Розв'язання:

Множина:

$$R = \{-2, -1, 0, 1, 2\}$$

На даній множині відношення є **функціональним, але не може бути бієктивним**.

Завдання №2. Написати програму, яка знаходить матрицю бінарного відношення $\rho \subset A \times B$, заданого на двох числових множинах. Реалізувати введення цих множин, та виведення на екран матриці відношення. Перевірити програмно якого типу є задане відношення. Навести різні варіанти тестових прикладів.

Текст програми

```
#include <iostream>
#include "BW1.h"

using namespace std;

void Output(int **A, int *B, int *C, int p, int q);
int Symetria(int **A, int p, int q);
int Reflexive(int **A, int p, int q);
int Anrireflexive(int **A, int p, int q);
int Transitive(int **A, int p, int q);
int Antitransitive(int **A, int p, int q);

int main()
{
    int n, m;
    cout << "Input size of A: ";
    cin >> n;
    cout << "Input size of B: ";
    cin >> m;
    if (m != n)
    {
        cout << "The size of A and B must be equal integer figures!" << endl;
    }
    else
    {
        cout << endl;
        int *A = new int[n];
        int *B = new int[m];
        int **Ro = new int *[n];
        for (int i = 0; i < n; i++)
```

```
        for (int i = 0; i < n; i++)
        {
            Ro[i] = new int[m];
        }
        for (int i = 0; i < n; i++)
        {
            cout << "Input the element number " << i + 1 << " of A: ";
            cin >> A[i];
        }
        for (int j = 0; j < m; j++)
        {
            cout << "Input the element number " << j + 1 << " of B: ";
            cin >> B[j];
        }
        Output(Ro, A, B, n, m);
        Symetria(Ro, n, m);
        Reflexive(Ro, n, m);
        Antireflexive(Ro, n, m);
        Transitive(Ro, n, m);
        Antitransitive(Ro, n, m);
    }
    return 0;
}
```

Текст заголовкового файлу "BW.h"

```
#include <iostream>

using namespace std;

void Output(int **A, int *B, int *C, int p, int q)
{
    for (int i = 0; i < p; i++)
    {
        for (int j = 0; j < q; j++)
        {
            A[i][j] = 0;
        }
    }
    for (int i = 0; i < p; i++)
    {
        for (int j = 0; j < q; j++)
        {
            if ((B[i] + C[j] + 1) > 3)
            {
                A[i][j] = 1;
            }
            else
            {
                A[i][j] = 0;
            }
        }
    }
    for (int i = 0; i < p; i++)
    {
        for (int j = 0; j < q; j++)
        {
            cout << A[i][j] << " ";
        }
        cout << endl;
    }
}

int Symetria(int **A, int p, int q)
{
    int a = 1;
    for (int i = 0; i < p; i++)
    {
        for (int j = 0; j < q; j++)
        {
            if ((A[i][j] == 1) && (A[j][i] == 1))
                a *= 1;
            else
                a *= 0;
        }
    }
    if (a == 1)
    {
        cout << "Matrix is symmetric." << endl;
    }
    if (a == 0)
    {
        cout << "Matrix is asymmetric." << endl;
    }
    return a;
}
```



```

int Reflexive(int **A, int p, int q)
{
    int a = 0;
    for (int i = 0; i < p; i++)
    {
        if (A[i][i] == 0)
        {
            cout << "Matrix isn't reflexive." << endl;
            a = 1;
            break;
        }
    }
    if (a == 0)
    {
        cout << "Matrix is reflexive." << endl;
    }
    return a;
}

int Antireflexive(int **A, int p, int q)
{
    int a = 1;
    for (int i = 0; i < p; i++)
    {
        for (int j = 0; j < q; j++)
        {
            if ((A[i][j] != 0) && (i == j))
            {
                a *= 0;
            }
        }
    }
    if (a == 1)
    {
        cout << "Matrix is antireflexive." << endl;
    }
    else
    {
        cout << "Matrix isn't antireflexive." << endl;
    }
    return a;
}

int Transitive(int **A, int p, int q)
{
    int a = 0;
    for (int i = 0; i < p; i++)
    {
        for (int j = 0; j < q; j++)
        {
            for (int h = 0; h < q; h++)
            {
                if (A[i][j] == 1 && A[j][h] == 1)
                {
                    if (A[i][h] != 1)
                    {
                        a = 1;
                        break;
                    }
                }
            }
        }
    }
}

```

```

    }
}
if (a == 0)
{
    cout << "Matrix is transitive." << endl;
}
else if(a == 1)
{
    cout << "Matrix isn't transitive." << endl;
}
return a;
}

int Antitransitive(int **A, int p, int q)
{
    int a = 0;
    for (int i = 0; i < p; i++)
    {
        for (int j = 0; j < q; j++)
        {
            for (int h = 0; h < q; h++)
            {
                if (A[i][j] == 1 && A[j][h] == 1)
                {
                    if (A[i][h] != 0)
                    {
                        a = 1;
                        break;
                    }
                }
            }
        }
    }
    if (a == 0)
    {
        cout << "Matrix is antitransitive." << endl;
    }
    else if (a == 1)
    {
        cout << "Matrix isn't antitransitive." << endl;
    }
    return a;
}
}

```

Результати програми

```
Input size of A: 3
Input size of B: 3

Input the element number 1 of A: 4
Input the element number 2 of A: -2
Input the element number 3 of A: 5
Input the element number 1 of B: 1
Input the element number 2 of B: -3
Input the element number 3 of B: 7
0 0 1
1 0 1
0 0 1
1 0 1
Matrix is asymmetric.
Matrix isn't reflexive.
Matrix isn't antireflexive.
Matrix isn't transitive.
Matrix isn't antitransitive.
```

```
Input size of A: 4
Input size of B: 4

Input the element number 1 of A: -2
Input the element number 2 of A: -4
Input the element number 3 of A: 1
Input the element number 4 of A: 6
Input the element number 1 of B: -7
Input the element number 2 of B: 2
Input the element number 3 of B: 3
Input the element number 4 of B: 5
0 0 0 1
0 0 0 0
0 1 1 1
0 1 1 1
Matrix is asymmetric.
Matrix isn't reflexive.
Matrix isn't antireflexive.
Matrix isn't transitive.
Matrix isn't antitransitive.
```

```
Input size of A: 3
Input size of B: 3

Input the element number 1 of A: -5
Input the element number 2 of A: -2
Input the element number 3 of A: 1
Input the element number 1 of B: 5
Input the element number 2 of B:
4
Input the element number 3 of B: -1
0 0 0
1 0 0
1 1 0
Matrix is asymmetric.
Matrix isn't reflexive.
Matrix is antireflexive.
Matrix is transitive.
Matrix isn't antitransitive.
```

```
Input size of A: 3
Input size of B: 3

Input the element number 1 of A: 3
Input the element number 2 of A: 2
Input the element number 3 of A: -4
Input the element number 1 of B: 1
Input the element number 2 of B: 4
Input the element number 3 of B: -2
1 1 0
1 1 0
0 0 0
Matrix is asymmetric.
Matrix isn't reflexive.
Matrix isn't antireflexive.
Matrix is transitive.
Matrix isn't antitransitive.
```

```
Input size of A: 3
Input size of B: 3

Input the element number 1 of A: -2
Input the element number 2 of A: 3
Input the element number 3 of A: -1
Input the element number 1 of B: 5
Input the element number 2 of B: -3
Input the element number 3 of B: 2
1 0 0
1 0 1
1 0 0
Matrix is asymmetric.
Matrix isn't reflexive.
Matrix isn't antireflexive.
Matrix is transitive.
Matrix isn't antitransitive.
```

```
Input size of A: 4
Input size of B: 4

Input the element number 1 of A: 9
Input the element number 2 of A: 2
Input the element number 3 of A: -4
Input the element number 4 of A: 1
Input the element number 1 of B: -2
Input the element number 2 of B: 3
Input the element number 3 of B: 7
Input the element number 4 of B: -6
1 1 1 1
0 1 1 0
0 0 1 0
0 1 1 0
Matrix is asymmetric.
Matrix isn't reflexive.
Matrix isn't antireflexive.
Matrix is transitive.
Matrix isn't antitransitive.
```

Висновок: на цій лабораторній роботі я набув практичних вмінь та навичок при побудові матриць бінарних відношень та визначені їх типів.