

**Міністерство освіти і науки України
Національний університет
“Львівська політехніка”**

Кафедра систем штучного інтелекту

Лабораторна робота № 3

з дисципліни

«Дискретна математика»

Виконав:

студент групи КН-115
Поставка Маркіян

Викладач:

Мельникова Н. І.

Львів – 2019р.

Тема: “Побудова матриці бінарного відношення”

Мета роботи: Набуття практичних вмінь та навичок при побудові матриць бінарних відношень та визначенні їх типів.

Додаток 1:

Варіант 11.

Постановка задачі:

Варіант № 11

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

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

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

3. Зобразити відношення графічно:

$$\alpha = \{(x, y) \mid (x, y) \in \mathbb{R}^2 \text{ \& } |x+3| \geq |y|\}, \text{ де } \mathbb{R} - \text{множина дійсних чисел.}$$

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

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

$$\alpha = \{(x, y) \mid (x, y) \in \mathbb{R}^2 \text{ \& } x + \sqrt{y^2} = 1\}.$$

Рішення:

1.

$$(A \cup B) \times (C \cup D) = (A \times C) \cup (B \times D)$$

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

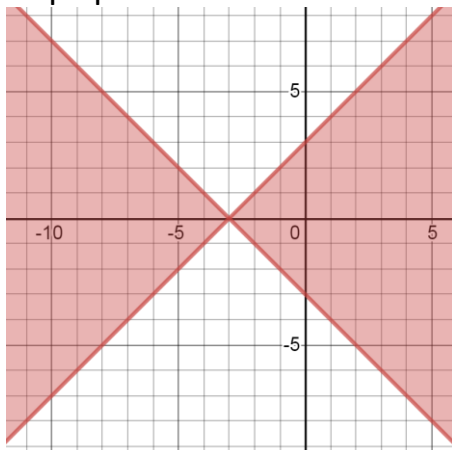
$$\Leftrightarrow (x \in A \text{ \& } y \in B) \text{ \& } (x \in C \text{ \& } y \in D) \Leftrightarrow (x \in A \text{ \& } x \in C) \text{ \& } (y \in B \text{ \& } y \in D) \Leftrightarrow$$

$$\Leftrightarrow (x \in (A \times C)) \text{ \& } (y \in (B \times D)) \Leftrightarrow (x, y) \in (A \times C) \cup (B \times D)$$

2.

$M \setminus 2^{ M }$	$\{\}$	$\{1\}$	$\{2\}$	$\{3\}$	$\{1, 2\}$	$\{1, 3\}$	$\{2, 3\}$	$\{1, 2, 3\}$
1	0	0	1	1	1	1	1	1
2	0	0	0	1	0	1	1	1
3	0	0	0	0	0	0	0	0

3. Графік:



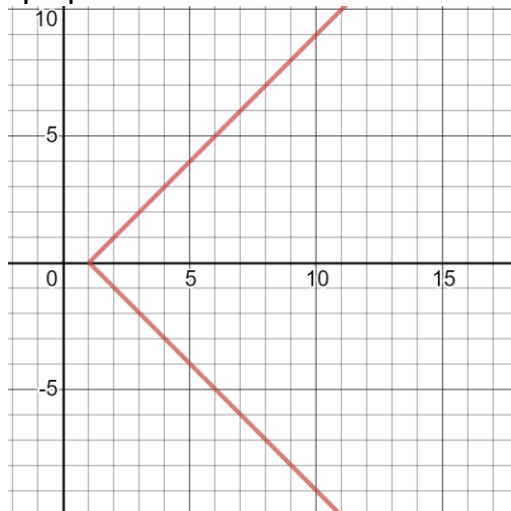
4.

	a	b	c	d	e
a	0	0	0	1	1
b	0	0	0	0	1
c	0	1	0	0	0
d	0	0	1	0	0
e	0	0	0	0	0

5.

$$x - 1 = |y|$$

Графік:



На $x, y \in \mathbb{R}$ – не функціональне.

На $x, y \in \mathbb{R}$ – не бієктивне.

Відношення буде функціональним та бієктивним якщо $y > 0$:

$$A = \{(x, y) \mid (x) \in \mathbb{R} \ \& \ y > 0 \ \& \ x + |y| = 1\}$$

Додаток 2:

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.

Відношення обрати згідно варіанту:

$$11. \quad \rho = \{(a, b) \mid a \in A \ \& \ b \in B \ \& \ 2a > 3b\};$$

Постановка задачі:

Код програми:

```
#include "stdafx.h"
#include <iostream>
#include <conio.h>
#include <math.h>
```

```

using namespace std;

//const int n = 5;

struct set
{
    void inp_set(int *a, int n) ///////////
    {
        char ck[5];
        cout << "Enter set with different values" << endl;
        for (int i = 0; i < n; i++)
        {
            p1:
                cin >> ck;
                if (atoi(ck) || ck[0] == '0') a[i] = atoi(ck);
                else
                {
                    cout << "Enter again" << endl;
                    goto p1;
                }
        }
    }

    void out_set(int *a, int n) ///////////
    {
        for (int i = 0; i < n; i++)
        {
            cout << a[i] << "\t";
        }
        cout << endl;
    }

    void bin_rel(int *a, int an, int *b, int bn, bool **b_r)
    {
        for (int i = 0; i < an; i++)
        {
            for (int j = 0; j < bn; j++)
            {
                if ((2 * a[i]) > (3 * b[j])) b_r[i][j] = 1;
                else b_r[i][j] = 0;
            }
        }

        //b_r[0][0] = 1; b_r[0][1] = 0; b_r[0][2] = 1; b_r[0][3] = 1; //Preset 1
        //b_r[1][0] = 0; b_r[1][1] = 1; b_r[1][2] = 0; b_r[1][3] = 0;
        //b_r[2][0] = 0; b_r[2][1] = 0; b_r[2][2] = 1; b_r[2][3] = 0;
        //b_r[3][0] = 1; b_r[3][1] = 0; b_r[3][2] = 0; b_r[3][3] = 1;

        //b_r[0][0] = 1; b_r[0][1] = 0; b_r[0][2] = 1; b_r[0][3] = 0; //Preset 2
        //b_r[1][0] = 0; b_r[1][1] = 1; b_r[1][2] = 0; b_r[1][3] = 1;
        //b_r[2][0] = 1; b_r[2][1] = 0; b_r[2][2] = 1; b_r[2][3] = 0;
        //b_r[3][0] = 0; b_r[3][1] = 1; b_r[3][2] = 0; b_r[3][3] = 1;

        cout << "AxB" << "\t";
        for (int j = 0; j < bn; j++)
        {
            cout << b[j] << "\t";
        }
        cout << endl;

        for (int i = 0; i < an; i++)
        {
            cout << a[i] << "\t";
            for (int j = 0; j < bn; j++)
            {
                cout << b_r[i][j] << "\t";
            }
            cout << endl;
        }
    }
}

struct binary
{
    int reflex(bool **b_r, int an, int bn, set s)
    {

```

```

        //bool flag = 1;
        int k = 0;
        for (int i = 0; i < an; i++)
        {
            if (b_r[i][i] == 1) k++;
        }
        cout << "Reflex: ";
        if (k == an)
        {
            cout << "True" << endl;
            return 1;
        }
        else
        {
            cout << "False" << endl;
            return 0;
        }
    }

void anti_reflex(bool **b_r, int an, int bn, set s)
{
    int k = 0;
    for (int i = 0; i < an; i++)
    {
        if (b_r[i][i] == 0) k++;
    }
    cout << "Anti reflex: ";
    if (k == an) cout << "True" << endl;
    else cout << "False" << endl;
}

int simetr(bool **b_r, int an, int bn, set s)
{
    bool flag = 1;
    int i = 0;
    do {
        for (int j = 0; j < an; j++)
        {
            if (i != j)
            {
                if ((b_r[i][j] == 1 && b_r[j][i] == 0) || (b_r[i][j] == 0 &&
b_r[j][i] == 1)) flag = 0;
            }
        }
        i++;
    } while (i < an && flag != 1);
    cout << "Simetr: ";
    if (flag)
    {
        cout << "True" << endl;
        return 1;
    }
    else
    {
        cout << "False" << endl;
        return 0;
    }
}

int a_simetr(bool **b_r, int an, int bn, set s, int sim)
{
    if (sim == 1)
    {
        cout << "A_simetr: ";
        cout << "False" << endl;
        return 0;
    }
    bool flag = 1;
    int i = 0;
    int tf = 0;
    int ff = 0;
    do {
        for (int j = 0; j < bn; j++)
        {
            if (i != j)
            {

```

```

        b_r[j][i] == 1))

        if ((b_r[i][j] == 1 && b_r[j][i] == 0) || (b_r[i][j] == 0 &&
        {
            flag = 0;
            ff++;
        }
        if (b_r[i][j] == b_r[j][i]) tf++;
    }
    }
    i++;
} while (i < an && flag != 1);
cout << "A_simetr: ";
if (!flag && ff!=0 &&tf !=0) {
    cout << "True" << endl;
    return 1;
}
else
{
    cout << "False" << endl;
    return 0;
}
}

void anti_simetr(bool **b_r, int an, int bn, set s, int sim, int a_sim)
{
    if (sim == 1 || a_sim ==1)
    {
        cout << "Anti_simetr: ";
        cout << "False" << endl;
        return;
    }

    bool flag = 1;

    int i = 0;
    do {
        for (int j = 0; j < an; j++)
        {
            if (i != j)
            {
                if (b_r[i][j] == 1 && b_r[j][i] == 1) flag = 0;
            }
        }
        i++;
    } while (i < an && flag != 1);
    cout << "Anti_simetr: ";
    if (flag) cout << "True" << endl;
    else cout << "False" << endl;
}

int trans(bool **b_r, int an, int bn, set s)
{
    bool flag = 1;

    for (int i = 0; i < an; i++)
    {
        for (int j = 0; j < an; j++)
        {
            for (int k = 0; k < an; k++)
            {
                if (b_r[i][k] && b_r[k][j] && !b_r[i][j]) flag = 0;
            }
        }
    }
    cout << "Trans: ";
    if (flag)
    {
        cout << "True" << endl;
        return 1;
    }
    else
    {
        cout << "False" << endl;
        return 0;
    }
}

```

```

    }

    void anti_trans(bool **b_r, int an, int bn, set s)
    {
        bool flag = 1;

        for (int i = 0; i < an; i++)
        {
            for (int j = 0; j < bn; j++)
            {
                for (int k = 0; k < an; k++)
                {
                    if (b_r[i][k] && b_r[k][j] && b_r[i][j]) flag = 0;
                }
            }
        }
        cout << "Anti_trans: ";
        if (flag) cout << "True" << endl;
        else cout << "False" << endl;
    }

}bin;

void type_b_r(bool **b_r, int an, int bn, set s)
{
    int refl = bin.reflex(b_r, an, bn, s);
    bin.anti_reflex(b_r, an, bn, s);
    int sim = bin.simetr(b_r, an, bn, s);
    int a_sim = bin.a_simetr(b_r, an, bn, s, sim);
    bin.anti_simetr(b_r, an, bn, s, sim, a_sim);
    int tran = bin.trans(b_r, an, bn, s);
    bin.anti_trans(b_r, an, bn, s);

    cout << "Full: ";
    if (refl == 1 && sim == 1 && tran == 1)
    {
        cout << "True" << endl;
    }
    else cout << "False" << endl;
}

};

int main()
{
    set s;

    int an, bn, b_rn;
    //cout << "Enter size of set a, b" << endl;
    //cin >> an >> bn;
    an = 4; bn = 4;
    int *a = new int[an];
    int *b = new int[bn];

    bool **b_r = new bool*[an];
    for (int i = 0; i < an; i++)
        b_r[i] = new bool[bn];

    s.inp_set(a, an);
    s.inp_set(b, bn);
    cout << endl;

    cout << "Set a:" << endl;
    s.out_set(a, an);
    cout << endl;
    cout << "Set b:" << endl;
    s.out_set(b, bn);
    cout << endl;

    s.bin_rel(a, an, b, bn, b_r);
    cout << endl;

    s.type_b_r(b_r, an, bn, s);

    for (int i = 0; i < an; i++)
        delete[] b_r[i];
    delete[] b_r;
}

```

```

delete[] a;
delete[] b;
_getch();
return 0;
}

```

Результат роботи програми:

The first screenshot shows the program's execution in a Windows command prompt. The user is prompted to enter two sets of values. The first set 'a' contains the values 5, 7, and 6. The second set 'b' contains the values 4, 3, and 8. The program then displays a matrix AxB, which is a 3x4 matrix of binary values (0s and 1s). Below the matrix, the program lists several properties of the relation: Reflex: False, Anti reflex: False, Simetr: False, A_simetr: True, Anti_simetr: False, Trans: False, Anti_trans: False, and Full: False.

```

C:\Windows\system32\cmd.exe
Enter set with different values
1
5
7
6
Enter set with different values
2
4
3
8

Set a:
1      5      7      6

Set b:
2      4      3      8

AxB
2      4      3      8
1      0      0      0      0
5      1      0      1      0
7      1      1      1      0
6      1      0      1      0

Reflex: False
Anti reflex: False
Simetr: False
A_simetr: True
Anti_simetr: False
Trans: False
Anti_trans: False
Full: False

```

The second screenshot shows the same command prompt window displaying the resulting matrix AxB and the properties of the relation. The matrix AxB is a 4x4 matrix of binary values (0s and 1s). Below the matrix, the program lists several properties of the relation: Reflex: True, Anti reflex: False, Simetr: True, A_simetr: False, Anti_simetr: False, Trans: True, Anti_trans: False, and Full: True.

```

C:\Windows\system32\cmd.exe
1      0      1      0
0      1      0      1
1      0      1      0
0      1      0      1

Reflex: True
Anti reflex: False
Simetr: True
A_simetr: False
Anti_simetr: False
Trans: True
Anti_trans: False
Full: True

```

Висновок: Я набув практичних вмінь та навичок при побудові матриць бінарних відношень та визначенні їх типів.