МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА“

Кафедра систем штучного інтелекту

Лабораторна робота №3

З дисципліни

«Дискретна математика»

Виконала:

Студентка групи КН-115

Рокицька Анастасія

Викладач:

Мельникова Н.І.

Львів – 2019р.

Тема: Побудова матриці бінарного відношення.
Мета роботи: набуття практичних вмінь та навичок при побудові матриць бінарних відношень та визначені їх типів.

Короткі теоретичні відомості

Види бінарних відношень.
Нехай задано бінарне відношення R на множині A : R A A (a, b) a A, b A 2 .
1. Бінарне відношення R на множині A називається рефлексивним, якщо для будь якого a ∈ A виконується aRa , тобто (a,a)∈R . Головна діагональ матриці рефлексивного відношення складається з одиниць. Граф рефлексивного відношення обов'язково має петлі у кожній вершині.
2. Бінарне відношення R на множині A називається антирефлексивним, якщо для будь якого a ∈ A не виконується aRa , тобто (a,a)∉ R . Головна діагональ матриці антирефлексивного відношення складається з нулів. Граф антирефлексивного відношення не має петель.
3. Бінарне відношення R на множині A називається симетричним, якщо для будь яких a,b∈ A з aRb слідує bRa , тобто якщо (a,b)∈R то і (b,a)∈ R . Матриця симетричного відношення симетрична відносно головної діагоналі. Граф симетричного відношення не є орієнтованим.
4. Бінарне відношення R на множині A називається антисиметричним, якщо для будь яких a,b∈ A з aRb та bRa слідує що a = b . Тобто якщо (a,b)∈R і (b,a)∈ R , то a = b . Матриця антисиметричного відношення не має жодної пари одиниць, які знаходяться на симетричних місцях по відношенню до головної діагоналі. У графа антисиметричного відношення вершини з'єднуються тільки однією напрямною дугою.
5. Бінарне відношення R на множині A називається транзитивним, якщо для будь яких a, b, c∈ A з aRb та bRc слідує, що aRc . Тобто якщо (a,b)∈R і (b,c)∈ R, то (a,c)∈ R . Матриця транзитивного відношення характеризується тим, що якщо елемент матриці σ_{ij} = 1 та σ_{jm} =1, то обов'язково σ_{im} =1. Граф транзитивного відношення такий, що якщо з'єднані дугами, наприклад, перша-друга та друга-третя вершини, то обов'язково є дуга з першої в третю вершину. 6. Бінарне відношення R на множині A називається антитранзитивним, якщо для будь яких a, b, c∈ A з aRb та bRc слідує що не виконується aRc . Тобто якщо (a, b)∈R і (b, c)∈ R, то (a, c)∉ R . Матриця антитранзитивного відношення характеризується тим, що якщо елемент матриці σ_{ij} = 1 та σ_{jm} =1, то обов'язково σ_{im} =0. Граф транзитивного відношення такий, що якщо з'єднані дугами, наприклад, перша-друга та друга-третя вершини, то обов'язково немає дуги з першої в третю вершину.

Варіант 13

Додаток 1:

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

2. Знайти матрицю відношення $R \subset M \times 2^M$:

$R = \{(x,\ y)| x \in M \ \&\ x \in y \ \&\ |y| > x\}$, де $M = \{x| x \in Z \ \&\ |x| \le 1\}$, Z – множина цілих чисел.

3. Зобразити відношення графічно:

$\alpha = \{(x,\ y)|(x, y) \in R^2 \ \&\ (x-y)^2 = 9\}$, де R - множина дійсних чисел.

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

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

$$\alpha = \left\{(x,\ y)|(x, y) \in R^2 \ \&\ y = \left(\sqrt{x}\right)^4\right\}.$$

1. (A×B)∪(C×D)=(A∪C)×(B∪D)

Нехай (x,y) ∈ (A×B)∪(C×D) ⇔

(x,y) ∈ (A×B)||(x,y) ∈ (C×D) ⇔

(x ∈ A & y ∈ B)||(x ∈ C & y ∈ D) ⇔

(x ∈ A ||(x ∈ C)&(y ∈ B & y ∈ D) ⇔

(x ∈ A∪C)&(y ∈ B∪D) ⇔

(x,y) ∈ (A∪C)×(B∪D).

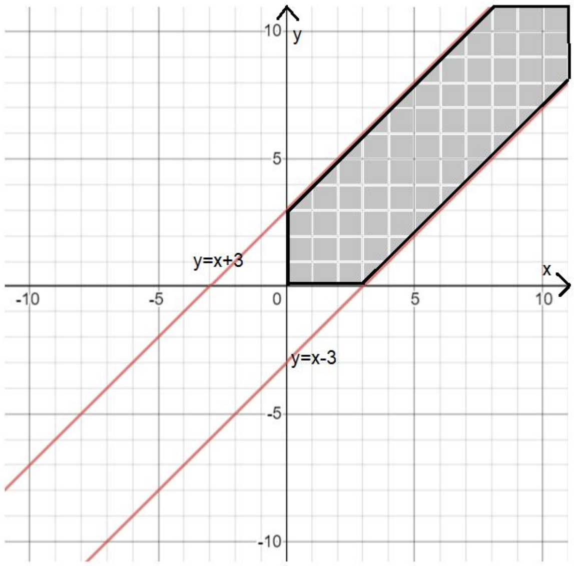
Рівність є правильною.

2.

	⊘	{-1}	{0}	{1}	{-1,0}	{-1,1}	{0,1}	{-1,0,1}
-1	1	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1
1	0	0	0	0	1	1	1	1

3.

$$\begin{cases} y=x-3 \\ y=x+3 \end{cases}$$

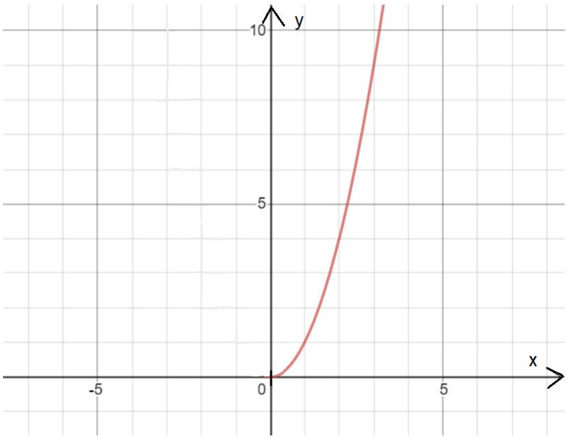


Сіра ділянка буде графічним відношенням так як (x,y) ∈ R^2

4. R ⊂ A×A, де A={a,b,c,d,e}
Нерефлексивне, коли хоча б один елемент, який стоїть на діагоналі (a_{ii}) дорівнює 1, але не всі.
Симетричне, коли $a_{ik} = a_{ki}$.
Транзитивне, коли $a_{ij} = a_{jk} = a_{ik}$.

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

5.



a={(x,y)|(x,y) ∈ R^2 & y= $(\sqrt{x})^4$ }

Область визначення: x ∈ [0;+∞)

Область значення: y ∈ [0;+∞)

Додаток:

```cpp
#include <iostream>
#include <math.h>
using namespace std;
void Output(int arr[], int size);
int main() {
    int size;
    int A[100];
    int B[100];
    //створення масиву
    cout << "Enter a size of sets (n*n) : ";
    cin >> size;
    //перевірка
    do {
        if (size > 100 || size < 0) {
            cout << " Error, try again :\t";
            cin >> size;
        }
    } while (size > 100 || size < 0);
    cout << "Fill A: " << endl;
    for (int i = 0; i < size; i++) {
        cout << "\t";
        cin >> A[i];
    }
    cout << " .................................................." << endl;
    cout << "Fill B : " << endl;
    for (int i = 0; i < size; i++) {
        cout << "\t";
        cin >> B[i];
    }
    //виведення
    cout << "\t\t YOUR_SETS_ARE : " << endl << endl;
    cout << "\t\tA = ";
    Output(A, size);
    cout << "\t\tB = ";
    Output(B, size);
    //створення відношень
    cout << endl << "\tCartesian product of A and B ( A * B ) is :" << endl << endl;
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++)
        {
```

```cpp
        {
            cout << "\t";
            cout << "(" << A[i] << ", " << B[j] << ")";
        }
        cout << endl;
    }
    cout << endl;
    cout << "Couples which satisfy the condition :" << endl;
        int counter = 0;
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++)
        {
            if ((A[i]*2-B[j] ) < 3 ) {
                cout << "(" << A[i] << ", " << B[j] << ")" << " ";
            }
            else
            {
                counter++;
            }
        }
    }
    if (counter == size * size) {
        cout << " Couples are not found...";
    }
    cout << endl << endl;
    //відношення
    int prod[100][100];
    for (int i = 0; i < size; i++)
    {
        for (int j = 0; j < size; j++)
        {
            if ((A[i] + 2) > 3 * B[j]) {
                prod[i][j] = 1;
            }
            else {
                prod[i][j] = 0;
            }
        }
    }
    for (int i = 0; i < size; i++)
    {
```

```
	}
	for (int i = 0; i < size; i++)
	{
		cout << "\t";
		for (int j = 0; j < size; j++)
		{
			cout << prod[i][j] << "\t";
		}
		cout << endl;
	}
	cout << endl << endl;
	//типи відношень
	cout << "This relation is :" << endl;
	if (counter == size * size) {
		cout << "\tEmpty";
	}
	else {
		//рефлексивне
		int ref = 0;
		for (int i = 0; i < size; i++)
		{
			if (prod[i][i] == 1) {
				ref++;
			}
		}
		if (ref == size) {
			cout << "Reflexive " << endl;
		}
		//антирефлексивне
		int aref = 0;
		for (int i = 0; i < size; i++)
		{
			if (prod[i][i] == 0) {
				aref++;
			}
		}
		if (aref == size) {
			cout << "Antireflexive " << endl;
		}
		//симетричне
		int sym = 0;
```

```
int sym = 0;
for (int i = 0; i < size; i++)
{
    for (int j = 0; j < size; j++)
    {
        if (prod[i][j] == prod[j][i] && i!=j) {
            sym++;
        }
    }
}
if (sym == (pow(size, 2)) - size) {
    cout << "Symmetrical" << endl;
}
//антисиметричне
int asym = 0;
for (int i = 0; i < size; i++)
{
    for (int j = 0; j < size; j++)
    {
        if (prod[i][j] == 1 && prod[j][i] == 1) {
            asym++;
        }
    }
}
if (asym == 0) {
    cout << "Antisymmetrical" << endl;
}
//транзитивна
int trans = 0;
for (int i = 0; i < size; i++) {
    for (int j = 0; j < size; j++)
    {
        for (int k = 0; k < size; k++)
        {
            if ((prod[i][j] == 1) && (prod[j][k] == 1)) {
                if (prod[i][k] == 1) {
                    trans = 1;
                    break;
                }
            }
```

```
                {
                    if ((prod[i][j] == 1) && (prod[j][k] == 1)) {
                        if (prod[i][k] == 1) {
                            trans = 1;
                            break;
                        }
                    }
                }
            }
        }
        if (trans == 1) {
            cout << "Transitive" << endl;
        }
        if (trans == 0) {
            cout << "Not transitive" << endl;
        }
    }
    cout << endl << endl;
    system("pause");
    return 0;
}
void Output(int arr[], int size) {
    cout << " { ";
    for (int i = 0; i < size; i++) {
        if (i == (size - 1)) {
            cout << arr[i] << " ";
        }
        else {
            cout << arr[i] << ", ";
        }
    }
    cout << "} " << endl;
}
```

Результат програми:

```
Enter a size of sets (n*n) : 5
Fill A:
        1
        4
        5
        6
        3
Fill B :
        5
        4
        5
        6
        3
                 YOUR_SETS_ARE :

                A =  { 1, 4, 5, 6, 3 }
                B =  { 5, 4, 5, 6, 3 }

        Cartesian product of A and B ( A * B ) is :

        (1, 5)  (1, 4)  (1, 5)  (1, 6)  (1, 3)
        (4, 5)  (4, 4)  (4, 5)  (4, 6)  (4, 3)
        (5, 5)  (5, 4)  (5, 5)  (5, 6)  (5, 3)
        (6, 5)  (6, 4)  (6, 5)  (6, 6)  (6, 3)
        (3, 5)  (3, 4)  (3, 5)  (3, 6)  (3, 3)
```

```
Couples which satisfy the condition :
(1, 5) (1, 4) (1, 5) (1, 6) (1, 3) (4, 6) (3, 5) (3, 4) (3, 5) (3, 6)

        0       0       0       0       0
        0       0       0       0       0
        0       0       0       0       0
        0       0       0       0       0
        0       0       0       0       0


This relation is :
Antireflexive
Symmetrical
Antisymmetrical
Not transitive
```