

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

Кафедра систем штучного інтелекту

Лабораторна робота
з дисципліни
«Дискретна математика»

Виконав:
студент групи КН-109
Коваль Назар
Викладач:
Мельникова Н.І.

Львів – 2018 р.

Лабораторна робота № 2.

Тема: Побудова матриці бінарного відношення

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

Варіант № 5

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

Нехай $(x, y) \in (A \times B) \cap (C \times D) \Rightarrow (x, y) \in (A \times B) \& (x, y) \in (C \times D) \Rightarrow x \in A \& y \in B; x \in C \& y \in D; \Rightarrow x \in A \& x \in C; y \in B \& y \in D \Rightarrow x \in (A \cap C) \& y \in (B \cap D) \Rightarrow (x, y) \in (A \cap C) \times (B \cap D)$

Вираз $(A \cap C) \times (B \cap D) \Leftrightarrow (A \times D) \cap (C \times B)$, отже рівність є вірною.

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

$R = \{(x, y) \mid x \in M \& y \subset M \& |y| < x + 2\}$, де $M = \{x \mid x \in \mathbb{Z} \& |x| \leq 1\}$,

\mathbb{Z} - множина цілих чисел.

Оскільки $x \in \mathbb{Z}; |x| \leq 1 \Rightarrow$ множина $M = \{-1; 0; 1\}$

$x \backslash y$	$\{\emptyset\}$	$\{-1\}$	$\{0\}$	$\{1\}$	$\{-1, 0\}$	$\{-1, 1\}$	$\{0, 1\}$	$\{-1, 0, 1\}$
-1	0	0	0	0	0	0	0	0
0	1	1	1	1	0	0	0	0
1	1	1	1	1	1	1	1	0

Матриця відношення:
$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

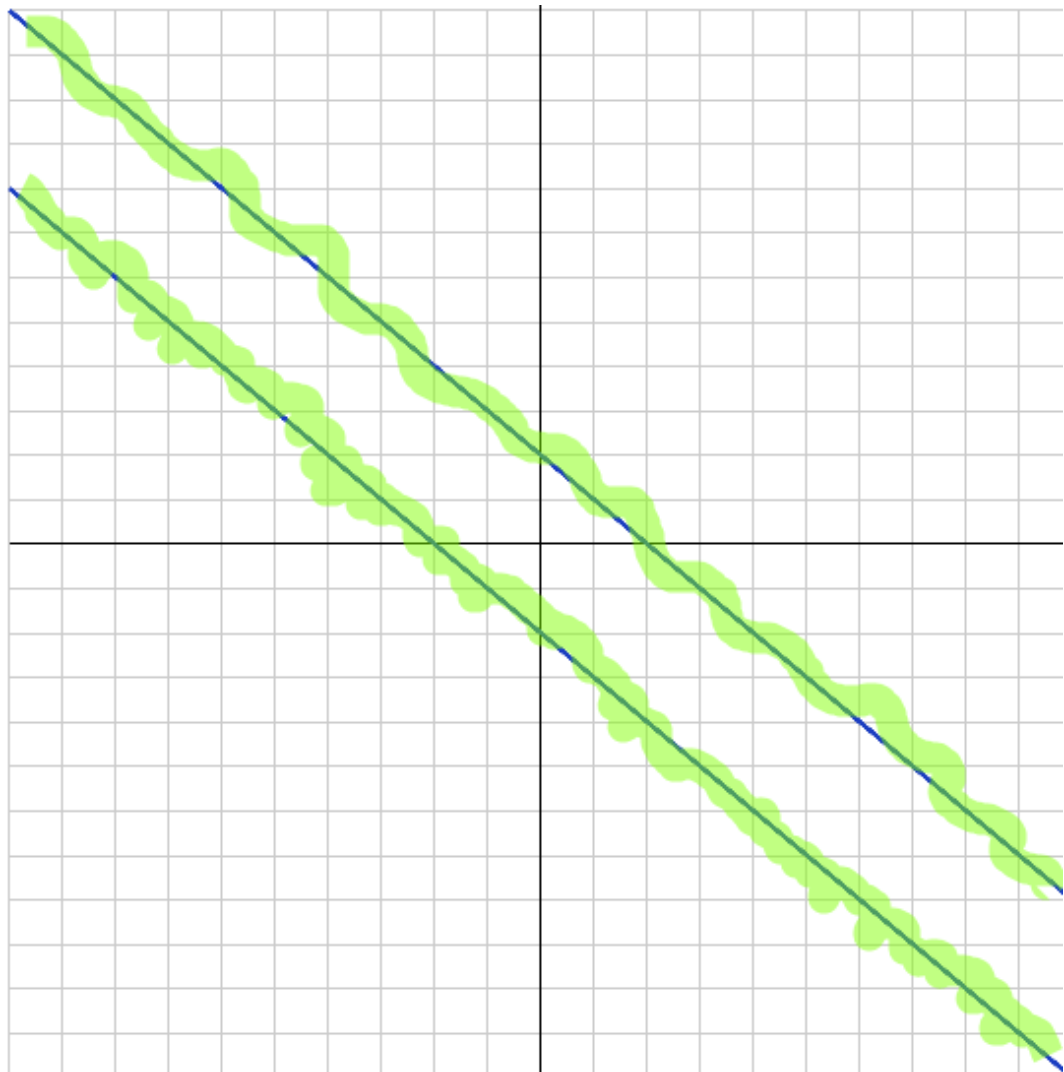
3. Зобразити відношення графічно:

$\alpha = \{(x, y) | (x, y) \in R^2 \ \& \ (x + y)^2 = 4\}$, де R - множина дійсних чисел.

Побудуємо графік функції $(x + y)^2 = 4$

$$|x+y|=2$$

$$\begin{cases} x + y = 2 \\ x + y = -2 \\ y = 2 - x \\ y = -2 - x \end{cases}$$



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

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

$$R = \{a, a\} \{a, c\} \{a, e\} \{b, b\} \{b, d\} \{c, a\} \{c, c\} \{c, e\} \{d, b\} \{d, d\} \{e, a\} \{e, c\} \{e, e\}$$

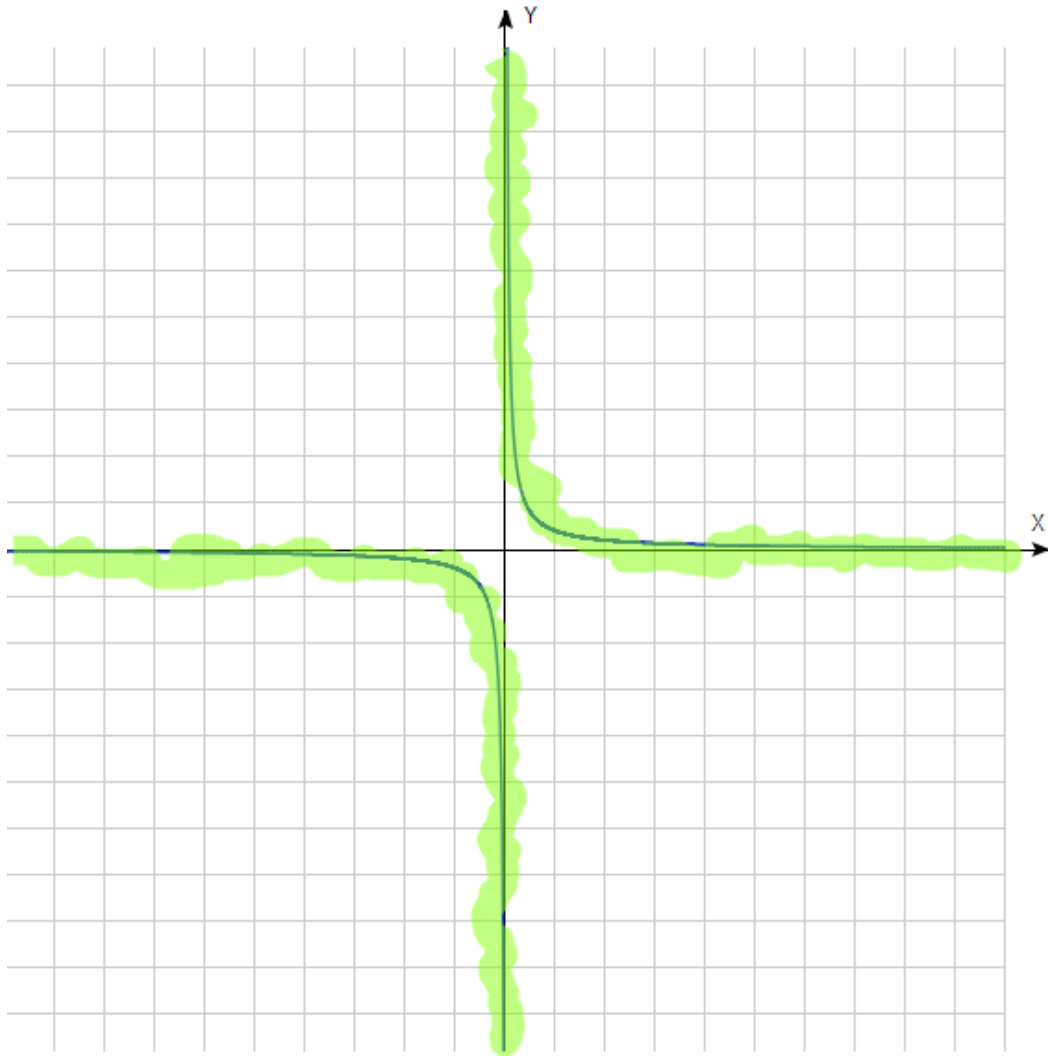
5. Визначити множину (якщо це можливо), на якій дане відношення є: а) функціональним; б) бієктивним:

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

Побудуємо графік функції $xy=2$

$$y = \frac{2}{x}$$

Графіком цієї функції буде гіпербола, зображена нижче.



Дане відношення є функціональним на множині усіх дійсних чисел, окрім нуля, тобто $\mathbf{R} \setminus \{0\}$

Дане відношення є неін'єктивним, оскільки деякі (y) та (x) мають більше ніж 1 пару. Оскільки воно неін'єктивне, воно також не є бієкційним, або можна сказати що воно є бієкційним на порожній множині \emptyset

Завдання №2. Написати програму, яка знаходить матрицю бінарного відношення $\rho \subseteq A \times B$, 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:

6. $\rho = \{(a, b) | a \in A \ \& \ b \in B \ \& \ (a + 2) > 3b\};$

Код програми:

```
#include<stdio.h>

#include<stdlib.h>

int main ()
```

```
{
int a[5];
printf("Put the elements of a array\n");
    for(int i=0; i<5; i++)
    {
        printf("Put element[%d]: ",i);
        scanf("%d",&a[i]);
    }
printf("\n");
printf("A array:");
    for(int i=0; i<5; i++)
    {
        printf("%d",a[i]);
        printf("|");
    }
printf("\n\n");
int b[5];
printf("Put the elements of b array\n");
    for(int i=0; i<5; i++)
    {
        printf("Put element[%d]: ",i);
        scanf("%d",&b[i]);
    }
printf("\n");
printf("B array:");
    for(int i=0; i<5; i++)
    {
        printf("%d",b[i]);
        printf("|");
    }
printf("\n");

int c[5][5];
    for(int i=0; i<5; i++)
```

```

{
    for(int j=0; j<5; j++)
    {
        if(2+a[i]>3*b[j])
        {
            c[i][j]=1;
        }
        else
        {
            c[i][j]=0;
        }
    }
}

printf("\n");

for(int i=0; i<5; i++)
{
    for(int j=0; j<5; j++)
    {
        printf("%d",c[i][j]);
        printf("|");
    }
    printf("\n");

}

for(int i=0; i<5-1; i++)
{
    if(c[i][i]==1 && c[1+i][1+i]==1)
    {
        if(i==5-2)
        {
            printf("Reflecsive matrix");
        }
    }
}

```

```

if(c[i][i]==0 && c[1+i][1+i]==0)
{
    if(i==5-2)
    {
        printf("Unreflecsive matrix");
    }
}
if((c[i][i]==1 && c[1+i][1+i]==0) || (c[i][i]==0 && c[1+i][1+i]==1))
{
    if(i==5-2)
    {
        printf("Antireflecsive matrix");
    }
}
}
printf("\n");
for(int i=0; i<5-1; i++)
{
    for(int j=0; j<5-1; j++)
    {
        if( c[i][j]==c[j][i]==1 || c[i][j]==c[j][i]==0)
        {
            if(i==5-2 && j==5-2)
            {
                printf("Symetric matrix");
                i+=10;
                j+=10;
            }
        }
        if((c[i][j]==1 && c[j][i]==0) || (c[i][j]==0 && c[j][i]==1))
        {
            if(i==5-2 && j==5-2)
            {
                printf("Antisymetric matrix");
            }
        }
    }
}

```



```

        i+=10;
        j+=10;
    }
}
else
{
    printf("Unsymetric matrix");
    i+=10;
    j+=10;
}
}
}
printf("\n");
for(int i=0; i<5-1; i++)
{
    for(int j=0; j<5-1; j++)
    {
        for(int k=0; k<5-1; k++)
        {
            if(c[i][j]==c[j][k]==1)
            {
                if(c[k][i]==1)
                {
                    if(c[i][j]==c[j][k]==0)
                    {
                        if(c[k][i]==0)
                        {
                            printf("Transitive matrix");
                            i+=10;
                            j+=10;
                            k+=10;
                        }
                    }
                }
            }
        }
    }
}

```

```

    }
    if(c[i][j]==c[j][k]==1)
    {
        if(c[k][i]==0)
        {
            if(c[i][j]==c[j][k]==0)
            {
                if(c[k][i]==1)
                {
                    printf("Antitransitive matrix");
                    i+=10;
                    j+=10;
                    k+=10;
                }
            }
        }
    }
    else
    {
        printf("Untransitive matrix");
        i+=10;
        j+=10;
        k+=10;
    }
}

printf("\n");

return 0;
}

```