МІНІСТЕРСТВО ОСВІТИ І НАУКИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра систем штучного інтелекту



Лабораторна робота №

З курсу “Дискретна Мвтематика ”

Виконав:  
ст.гр. КН-110

Андрусяк Нестор

Львів – 2018

Лабораторна робота № 3

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

**Варіант № 1**

**Завдання** **1.** Чи є вірною рівність (*A*× *B*) ∩ (*A*×*C*) = *A*×(*B* ∩*C*) ?

Нехай (x, y) ∈ (*A*× *B*) ∩ (*A*×*C*) ⇔

(x, y) ∈ (*A*× *B*) & (x, y) ∈ (*A*×*C*) ⇔

(x ∈ A & y ∈ B ) & (x ∈ A & y ∈ C ) ⇔

(x ∈ A) & (y ∈ B & y ∈ C) ⇔

(x ∈ A) & (y ∈ B ∩ C) ⇔ (x, y) ∈ *A*×(*B* ∩*C*);

Так, вірність є рівною.

**Завдання 2.**

Знайти матрицю відношення

*R* ⊂ *M* × 2*M* , де *M* ={1,2,3}:  
*R* = {(*x*, *y*) *x* ∈ *M* & *y* ⊂ *M* & |*y***|**= *x*}.

Згідно з означенням матриці відношення , розв’язок має вигляд:

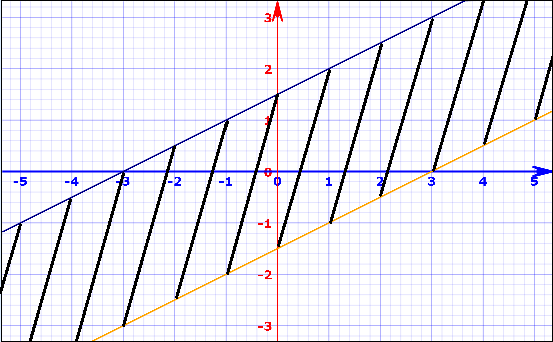
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ∅ | {1} | {2} | {3} | {1, 2} | {1, 3} | {2, 3} | {1, 2, 3} |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

**Завдання 3.**

Зобразити відношення графічно:  
α = {(*x*, *y*) (*x*, *y*) ∈ & |*x* − 2*y|* ≤ 3 }, де *R* - множина дійсних чисел

Запишемо систему нерівностей: . Відповідно .

Зобразимо це відношення графічно:



**Завдання 4.**

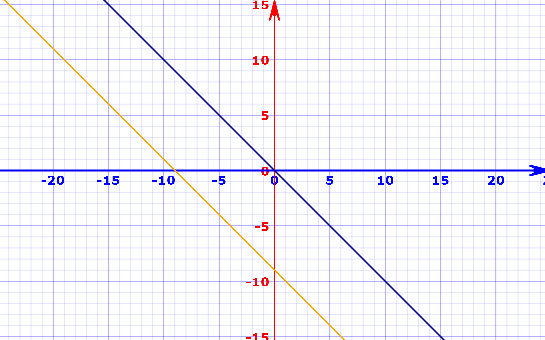
Навести приклад бінарного відношення *R* ⊂ *A*× *A*, де *A* ={*a*, *b, c, d, e*}, яке є рефлексивне,  
симетричне, нетранзитивне, та побудувати його матрицю.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | a | b | c | d | e |
| a | 1 | 0 | 0 | 0 | 1 |
| b | 0 | 1 | 0 | 0 | 1 |
| c | 0 | 0 | 1 | 1 | 0 |
| d | 0 | 0 | 1 | 1 | 0 |
| e | 1 | 1 | 0 | 0 | 1 |

**Завдання 5.** Визначити множину (якщо це можливо), на якій дане відношення є: а) функціональним; б)  
бієктивним:  
α = {(*x*, *y*) (*x*, *y*) ∈ *R*2 & = 9}

Розв’яжемо ; **|**x + y **| =** 9; ; .

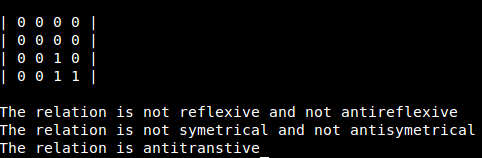
Побудуємо графік:

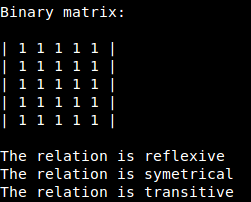
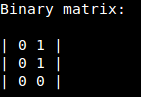


З графіку можна побачити що відношення не є функціональним і не є бієктивним.

**Завдання №2.** Написати програму, яка знаходить матрицю бінарного відношення ρ⊂ *A*× *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.

Результати:





Код:

#include <stdio.h>

int main(void)

{

int arr1[55], size1, arr2[55], size2, arr3[55][55];

printf("Enter the size of 1st array: ");

scanf("%d", &size1);

printf("Enter elements of 1st array:\n");

for(int i = 0; i < size1; i++)

{

scanf("%d", &arr1[i]);

}

printf("Enter the size of 2nd array: ");

scanf("%d", &size2);

printf("Enter elements of 2nd array:\n");

for(int i = 0; i < size2; i++)

{

scanf("%d", &arr2[i]);

}

printf("Binary matrix:\n\n");

for(int i = 0; i < size1; i++)

{

for(int j = 0; j < size2; j++)

{

if(arr1[i] > arr2[j])

{

arr3[i][j] = 1;

}

else

{

arr3[i][j] = 0;

}

}

}

for(int i = 0; i < size1; i++)

{

printf("| ");

for(int j = 0; j < size2; j++)

{

printf("%d ", arr3[i][j]);

}

printf("|\n");

}

int reflexive = 0;

int antireflexive = 0;

if(size1 == size2)

{

for(int i = 0, j = 0; i < size1; i++, j++)

{

if(arr3[i][j] == 1)

{

reflexive++;

}

else if(arr3[i][j] == 0)

{

antireflexive++;

}

}

}

printf("\n");

if(reflexive == size1){

printf("The relation is reflexive\n");

}

if(antireflexive == size1)

{

printf("The relation is antireflexive\n");

}

if((reflexive != size1) && (antireflexive != size2))

{

printf("The relation is not reflexive and not antireflexive\n");

}

int symetrical = 0;

int antisymetrical = 0;

if(size1 == size2)

{

int transposed[size1][size2];

for(int i = 0; i < size1; i++)

{

for(int j = 0; j < size2; j ++)

{

transposed[i][j] = arr3[j][i];

}

}

for(int i = 0; i < size1; i++)

{

for(int j = 0; j < size2; j++)

{

if(transposed[i][j] == arr3[i][j])

{

symetrical++;

}

else if(transposed[i][j] != arr3[i][j])

{

antisymetrical++;

}

}

}

}

if(symetrical == size1 \* size2)

{

printf("The relation is symetrical\n");

}

if(antisymetrical == size1 \* size2)

{

printf("The relation is antisymetrical\n");

}

if((symetrical != size1 \* size2) && (antisymetrical != size1 \* size2))

{

printf("The relation is not symetrical and not antisymetrical\n");

}

int transitive = 1;

int antitransitive = 1;

for(int i = 0; i < size1; i++)

{

for(int j = 0; j < size1; j++)

{

for(int k = 0; k < size1; k++)

{

if(arr3[i][j]==1 && arr3[j][k]==1 && arr3[i][k]==1)

{

antitransitive = 0;

}

else if (arr3[i][j]==1 && arr3[j][k]==1 && arr3[i][k]==0)

{

transitive = 0;

}

else if(arr3[i][j] != 1)

{

transitive = 0;

antitransitive =0;

}

}

}

}

if ( transitive == 1)

{

printf("The relation is transitive\n");

}

else if( antitransitive == 1)

{

printf("The relation is antitransitive\n");

}

else if( (transitive==0)&&(antitransitive==0))

{

printf("The relation is antitranstive\n");

}

}