

Міністерство освіти і науки, молоді та спорту України

Національний університет «Львівська політехніка»

Інститут комп'ютерних наук та інформаційних технологій

Кафедра систем штучного інтелекту

Лабораторна робота №3

з дисципліни

“Дискретна математика”

Виконав:

Студент КН-114

Пилипів Андрій

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.

Варіант 6

ТЕОРЕТИЧНІ ВІДОМОСТІ ТА ПРИКЛАДИ РОЗВ'ЯЗАННЯ ЗАДАЧ

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 декартова добутку дорівнює $|A \times B| = |A| \times |B|$.

Приклад. Довести тотожність $(A \times B) \cap (C \times D) = (A \cap C) \times (B \cap D)$.

Розв'язання.

Нехай $(x, y) \in (A \times B) \cap (C \times D) \Leftrightarrow$

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

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

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

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

Бінарним відношенням R називається підмножина декартового добутку $A \times B$ (тобто $R \subset A \times B$).

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 $\delta_R = \{x \mid \exists y (x, y) \in R\}$, а *областю значень* – множина $\rho_R = \{y \mid \exists x (x, y) \in R\}$ (\exists – існує).

Для скінчених множин бінарне відношення $R \subset A \times B$ зручно задавати за допомогою *матриці* відношення $R_{m \times n} = (r_{ij})$, де $m = |A|$, а $n = |B|$.

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

Нехай $(x, y) \in (A \times B) \cap (C \times D) =$

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

$x \in A \text{ \& } y \in B \text{ \& } x \in C \text{ \& } y \in D =$

$x \in A \text{ \& } x \in C \text{ \& } y \in B \text{ \& } y \in D \neq (A \times C) \cap (B \times D)$.

Отже рівність не є вірною.

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

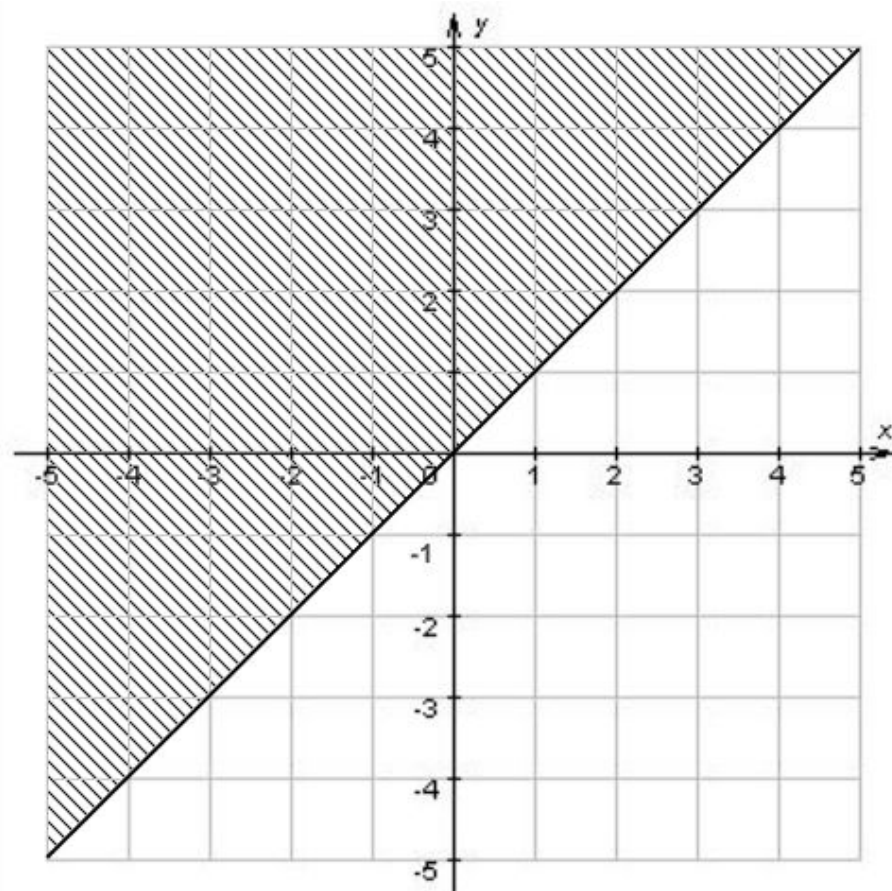
$R = \left\{ (x, y) \mid x \subset B \text{ \& } y \in A \text{ \& } |x| = \frac{y}{2} \right\}$, де $B = \{1, 2\}$, $A = \{y \mid y \in Z \text{ \& } 1 \leq y \leq 4\}$, Z - множина цілих чисел.

$R^{m \times n}$, де $m = |2^B| = 4$, $n = |A| = 4$

	$\{1\}$	$\{2\}$	$\{3\}$	$\{4\}$
$\{\}$	0	0	0	0
$\{1\}$	0	1	0	0
$\{2\}$	0	1	0	0
$\{3\}$	0	0	0	1

3. Зобразити відношення графічно:

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



4. Маємо бінарне відношення $R \subset A \times A$, де $A = \{a, b, c, d, e\}$, яке задане своєю матрицею:

$$A(R) = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \text{ Перевірити чи є дане відношення}$$

рефлексивним, симетричним, транзитивним, антисиметричним?

Відношення:

рефлексивне(головна діагональ -1)

симетричне(відносно головної діагоналі)

не транзитивне

не антисиметричне

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

$$\alpha = \{(x, y) | (x, y) \in R^2 \ \& \ (x + y)^3 = 5\}$$

а) на множині дійсних чисел

б) на множині дійсних чисел

Додаток 2

$$\rho = \{(a, b) | a \in A \& b \in B \& 2a < b\}$$

```
#include <iostream>
#include <sstream>
#include <locale>;
#include "windows.h";

using namespace std;

int size1;
int size2;
int* a;
int* b;
int** binaryMatr;

void inputSize() {
    cout << "Потужність першої множини = ";
    cin >> size1;
    cout << "потужність другої множини = ";
    cin >> size2;
    cout << endl;
    a = new int[size1];
    b = new int[size2];

    binaryMatr = new int* [size1];
    for (int i = 0; i < size1; i++) {
        binaryMatr[i] = new int[size2];
    }
}

void inputData(string title, int* arr, int size) {
    int d;
    string line;
    int factSize = 0;
```

```
    cout << title << "[" << factSize << "]" = ";
    while (getline(cin, line))
    {
        stringstream ss(line);
        if (ss >> d)
        {
            if (ss.eof())
            {
                bool present = 0;
                for (int i = 0; i < factSize; i++) {
                    if (arr[i] == d) present = 1;
                }
                if (!present) {
                    arr[factSize] = d;
                    factSize++;
                    if (factSize == size) break;
                    cout << title << "[" << factSize << "]" = ";
                }
            }
        }
    }

void sortArray(int* arr, int size) {
    int temp;
    for (int j = 0; j < size - 1; j++) {
        for (int i = 0; i < size - 1; i++) {
            if (arr[i] > arr[i + 1]) {
                temp = arr[i];
                arr[i] = arr[i + 1];
                arr[i + 1] = temp;
            }
        }
    }
```

```
    }
}

void displayArray(string title, int* arr, int size) {
    cout << title;
    for (int i = 0; i < size; i++) {
        cout << arr[i] << " ";
    }
    cout << endl << endl;
}

void displayMatr(string title, int** matr, int rows, int columns) {
    cout << title << endl;
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < columns; j++) {
            cout << matr[i][j] << " ";
        }
        cout << endl;
    }
}

void fillBinary() {
    for (int row = 0; row < size1; row++) {
        for (int column = 0; column < size2; column++) {
            if (2 * a[row] < b[column]) {
                binaryMatr[row][column] = 1;
            }
            else {
                binaryMatr[row][column] = 0;
            }
        }
    }
}
```

```
bool checkRefleks(int** matr, int rows, int columns) {
    for (int i = 0; i < rows && i < columns; i++) {
        if (matr[i][i] == 0) return false;
    }
    return true;
}

bool checkAntirefleks(int** matr, int rows, int columns) {
    for (int i = 0; i < rows && i < columns; i++) {
        if (matr[i][i] == 1) return false;
    }
    return true;
}

bool checkSymmetry(int** matr, int rows, int columns) {
    if (rows != columns) return false;
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < rows; j++) {
            if (matr[i][j] != matr[j][i]) return false;
        }
    }
    return true;
}

bool checkAntisymmetry(int** matr, int rows, int columns) {
    if (rows != columns) return false;
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < rows; j++) {
            if (i != j && matr[i][j] == matr[j][i]) return false;
        }
    }
}
```

```

    return true;
}

bool checkTrans(int** matr, int rows, int columns) {
    if (rows != columns) return false;
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < rows; j++) {
            for (int m = 0; m < rows; m++) {
                if (matr[i][j] == 1 && matr[j][m] == 1 && matr[i][m] != 1) return false;
            }
        }
    }
    return true;
}

bool checkAntitrans(int** matr, int rows, int columns) {
    if (rows != columns) return false;
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < rows; j++) {
            for (int m = 0; m < rows; m++) {
                if (matr[i][j] == 1 && matr[j][m] == 1 && matr[i][m] != 0) return false;
            }
        }
    }
    return true;
}

int main() {
    //setlocale (LC_CTYPE, "ukr");
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

```

```

inputSize();
inputData("Множина 1", a, size1);
sortArray(a, size1);
displayArray("Множина 1 = ", a, size1);
inputData("Множина 2", b, size2);
sortArray(b, size2);
displayArray("Множина 2 = ", b, size2);
fillBinary();
displayMatr("Бінарна матриця: ", binaryMatr, size1, size2);

if (checkRefleks(binaryMatr, size1, size2)) cout << "рефлексивне" << endl;
if (checkAntirefleks(binaryMatr, size1, size2)) cout << "антирефлексивне" << endl;
if (checkSymmetry(binaryMatr, size1, size2)) cout << "симетричне" << endl;
if (checkAntisymmetry(binaryMatr, size1, size2)) cout << "антисиметричне" << endl;
if (checkTrans(binaryMatr, size1, size2)) cout << "транзитивне" << endl;
if (checkAntitrans(binaryMatr, size1, size2)) cout << "антитранзитивне" << endl;

return 0;

```

```
Потужність першої множини = 3  
потужність другої множини = 5
```

```
Множина 1[0] = 2  
Множина 1[1] = 3  
Множина 1[2] = 4  
Множина 1 = 2 3 4
```

```
Множина 2[0] = 4  
Множина 2[1] = 5  
Множина 2[2] = 6  
Множина 2[3] = 7  
Множина 2[4] = 8  
Множина 2 = 4 5 6 7 8
```

```
Бінарна матриця:
```

```
0 1 1 1 1  
0 0 0 1 1  
0 0 0 0 0
```

```
антирефлексивне  
антитранзитивне
```