МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІСЬКА ПОЛІТЕХНІКА»

Кафедра ІСМ



**Звіт**

До лабораторної роботи №3

З дисципліни:

«Теорія прийняття рішень»

На тему:

**«Дослідження операцій з метризованими бінарними**

**відношеннями та їх властивостей»**

Виконав:

Студент групи КН-38

Макара С.В.

Прийняла:

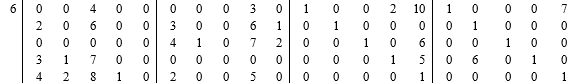
Заяць М.М.

**Львів-2017**

**Мета:** Вивчення та практичне ознайомлення з основними алгебрами метризованих бінарних відношень та з властивостями і типами метризованих бінарних відношень.

**Завдання:**

За допомогою програми визначити властивості та тип відношень *P*та*Q*, та здійснити обчислення основних операцій (об‘єднання, перетин, різниця, композиція) з допомогою відповідної алгебри Значення відношення *P*та*Q* згідно варіанту. При реалізації операцій з метризованими відношеннями необхідно враховувати вид відношення (адитивне, мультиплікативне) та реалізовувати відповідну алгебру. Повторити дії відношення *P1*та*Q1***.**

**Варіант №6:**

**Текст програми:**

**import java.util.Arrays;**

**import java.util.Scanner;**

**import java.util.StringTokenizer;**

**public class thff {**

**public static void outputing(double [][] A)**

**{**

**for(int i = 0;i<A.length;i++)**

**{**

**for(int j = 0;j<A.length;j++)**

**System.out.print(A[i][j] + " ");**

**System.out.println();**

**}**

**}**

**static double[][] peretyn(double[][] A, double[][] B)**

**{**

**double [][] R = new double[A.length][A.length];**

**for (int i = 0; i < A.length; i++)**

**for (int j = 0; j < A[i].length; j++)**

**if (A[i][j] == 0 || (B[i][j] == 0)) R[i][j] = 0;**

**else R[i][j] = ((A[i][j]) + (B[i][j])) / 2;**

**return R;**

**}**

**static double[][] plus(double[][] A, double[][] B)**

**{**

**double [][] R = new double[A.length][A.length];**

**for (int i = 0; i < A.length; i++)**

**for (int j = 0; j < A[i].length; j++)**

**if (A[i][j] == 0 || (B[i][j] == 0))**

**R[i][j] = (A[i][j]) + (B[i][j]);**

**else R[i][j] = ((A[i][j]) + (B[i][j])) / 2;**

**return R;**

**}**

**static double[][] minus(double[][] A, double[][] B)**

**{**

**double [][] R = new double[A.length][A.length];**

**for (int i = 0; i < A.length; i++)**

**for (int j = 0; j < A.length; j++)**

**if (A[i][j] != 0 && (B[i][j] == 0))**

**R[i][j] = (A[i][j]) + (B[i][j]);**

**else R[i][j] = 0;**

**return R;**

**}**

**static double[][] compose(double[][] A, double[][] B)**

**{**

**double [][] R = new double[A.length][A.length];**

**for (int i = 0; i < A.length; i++) {**

**for (int j = 0; j < A[i].length; j++) {**

**int cout = 0;**

**for (int k = 0; k < A.length; k++) {**

**if (A[i][k] != 0 && (B[k][j] != 0)) {**

**cout++;**

**R[i][j] += (A[i][k] + B[k][j]);**

**}**

**if (k == (A.length - 1) && (R[i][j] != 0)) R[i][j] /= cout;**

**}**

**}**

**}**

**return R;**

**}**

**static double[][] peretynM(double[][] A, double[][] B)**

**{**

**double [][] R = new double[A.length][A.length];**

**for (int i = 0; i < A.length; i++)**

**for (int j = 0; j < A[i].length; j++)**

**if (A[i][j] == 0 || (B[i][j] == 0)) R[i][j] = 0;**

**else R[i][j] = Math.sqrt((A[i][j]) \* (B[i][j]));**

**return R;**

**}**

**static double[][] plusM(double[][] A, double[][] B)**

**{**

**double [][] R = new double[A.length][A.length];**

**for (int i = 0; i < A.length; i++)**

**for (int j = 0; j < A[i].length; j++)**

**if (A[i][j] == 0 || (B[i][j] == 0)) R[i][j] = A[i][j] + B[i][j];**

**else R[i][j] = Math.sqrt(A[i][j] \* B[i][j]);**

**return R;**

**}**

**static double[][] minusM(double[][] A, double[][] B)**

**{**

**double [][] R = new double[A.length][A.length];**

**for (int i = 0; i < A.length; i++)**

**for (int j = 0; j < A.length; j++)**

**if (A[i][j] != 0 && (B[i][j] == 0))**

**R[i][j] = (A[i][j]) + (B[i][j]);**

**else R[i][j] = 0;**

**return R;**

**}**

**static double[][] composeM(double[][] A, double[][] B)**

**{**

**double [][] R = new double[A.length][A.length];**

**int cout = 0;**

**for (int i = 0; i < A.length; i++) {**

**for (int j = 0; j < A[i].length; j++) {**

**cout = 0;**

**for (int k = 0; k < A.length; k++) {**

**if (A[i][k] != 0 && (B[k][j] != 0)) {**

**cout++;**

**if (R[i][j] == 0)**

**R[i][j] = A[i][k] \* B[k][j];**

**else**

**R[i][j] \*= A[i][k] \* B[k][j];**

**}**

**if ((k == (A.length - 1)) && (R[i][j] != 0) && (cout >= 2))**

**R[i][j] = Math.pow(R[i][j], 1.0 / cout);**

**}**

**}**

**}**

**return R;**

**}**

**public static void main(String[] args) {**

**double[][] A = { { 0, 0, 4, 0, 0 }, { 2, 0, 6, 0, 0 }, { 0, 0, 0, 0, 0 }, { 3, 1, 7, 0, 0 },**

**{ 4, 2, 8, 1, 0 } };**

**double[][] B = { { 0, 0, 0, 3, 0 }, { 3, 0, 0, 6, 1 }, { 4, 1, 0, 7, 2 }, { 0, 0, 0, 0, 0 },**

**{ 2, 0, 0, 5, 0 } };**

**double[][] P = { { 1, 0, 0, 2, 10 }, { 0, 1, 0, 0, 0 }, { 0, 0, 1, 0, 6 }, { 0, 0, 0, 1, 5 },**

**{ 0, 0, 0, 0, 1 } };**

**double[][] Q = { { 1, 0, 0, 0, 7 }, { 0, 1, 0, 0, 0 }, { 0, 0, 1, 0, 0 }, { 0, 6, 0, 1, 0 },**

**{ 0, 0, 0, 0, 1 } };**

**System.out.println("Відношення:\nР:"); outputing(A);**

**System.out.println("Q:"); outputing(B);**

**System.out.println("P1:"); outputing(P);**

**System.out.println("Q1:"); outputing(Q);**

**boolean b = true;**

**Scanner in = new Scanner(System.in);**

**int scn = 0, scn2 = 0;**

**while (b) {**

**do{**

**System.out.println("Nип відношення:\n1 - Адитивні відношення\n2 - Мультиплікативні");**

**scn = in.nextInt();**

**}while(scn<1||scn>2);**

**do{**

**System.out.println("Виберіть оперцію, результат якої хочете отримати;");**

**System.out.println("1 - Перетин множин\n2 - Об'єднання множин");**

**System.out.println("3 - Різниця множин\n4 - Композиція");**

**scn2 = in.nextInt();**

**}while(scn2<1||scn2>4);**

**if (scn == 1)**

**switch (scn2) {**

**case 1: outputing(peretyn(A, B)); break;**

**case 2: outputing(plus(A, B)); break;**

**case 3: outputing(minus(A, B)); break;**

**case 4: outputing(compose(A, B)); break;**

**} else if (scn == 2)**

**switch (scn2) {**

**case 1:outputing(peretynM(P, Q)); break;**

**case 2: outputing(plusM(P, Q)); break;**

**case 3: outputing(minus(P, Q)); break;**

**case 4: outputing(composeM(P, Q)); break;**

**}**

**System.out.println("Бажаєте отримати результат ще якоїсь операції?\n1 - Так\n2 - Ні");**

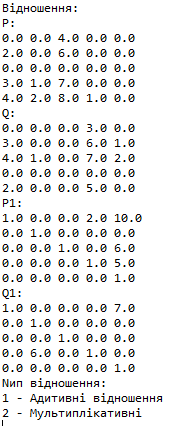
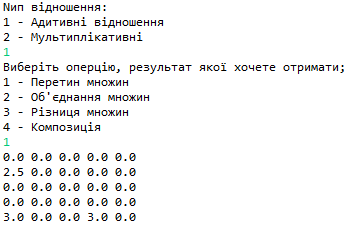
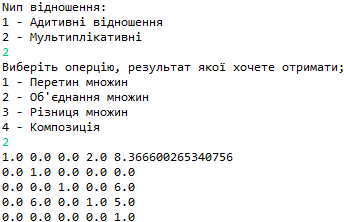
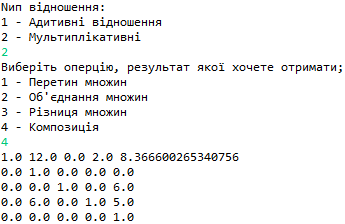
**int v = in.nextInt();**

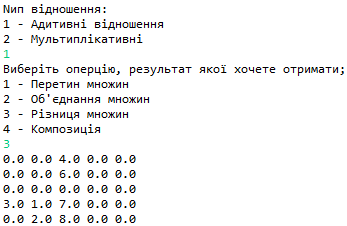
**if (v == 2) b = false;**

**}**

**}**

**}**

**Приклади роботи програми**

****

*Рис.1. Результати тестових прикладів*

**Висновки:** На даній лабораторній роботі мною було вивчено основи алгебри метризованих бінарних відношень, властивості та типи метризованих бінарних відношень.