Министерство образования и науки Российской Федерации

ФГБОУ ВО Ульяновский государственный технический университет

Кафедра «Вычислительная техника»

Лабораторная работа №2

«Бинарные отношения»

Выполнил студент

группы ИВТАСбд-11

Зюзин Г.А

Проверил Игонин А.Г.

Ульяновск, 2022 г.

# Оглавление.

[Оглавление. 2](#_Toc1)

[Цель работы. 3](#_Toc2)

[Метод выполнения.](#_Toc3)

[Листинг кода. 8](#_Toc4)

[Выводы. 13](#_Toc5)

[Список литературы.](#_Toc6)

# Цель работы.

Написать программу на JavaScript, которая определяет свойства бинарных отношения. Формат ввода: пара элементов. Необходимо выполнить проверку корректности ввода отношения. Программа определяет следующие свойства свойства: рефлексивность, симметричность, кососимметричность, транзитивность.

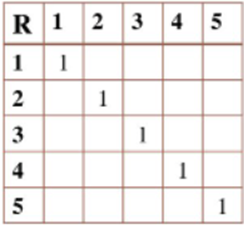
# Метод выполнения.

Данная программа определяет свойства элементов, имеющих формат ввода пара элементов. Программа определяет следующие свойства: рефлексивность, симметричность, кососимметричность, транзитивность.

Бинарным отношением между множествами A и В называется подмножество R прямого произведения A х В. В том случае, когда A = В, мы говорим просто об отношении R на А.

Рефлексивное отношение — бинарное отношение на множестве, при котором всякий элемент этого множества находится в отношении с самим собой.

Рис. 1

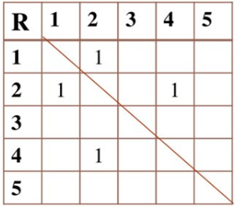


Формально, отношение рефлексивно, если свойство рефлексивности при заданных отношениях матрицей характеризуется тем, что все диагональные элементы матрицы равняются 1 (рис. 1).

После составления матрицы проверяется свойство рефлективности следующим образом: проверка диагонали на наличие нулей и единиц. Если находится ноль, то отношения не рефликсивны.

Симметричное отношение — бинарное отношение на множестве, при котором всякий элемент для каждой пары элементов множества (a, b) выполнение отношения aRb влечёт выполнение отношения bRa.

Рис. 2

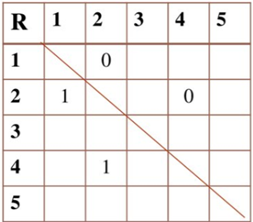


Формально, отношение симметрично, если единицы симметричны относительно главной диагонали. (симметрия матрицы относительно главной диагонали)(Рис. 2).

То есть программа проверяет симметричность. Если единицы не симметричны, то отношения не симметричны.

Кососимметричное (антисимметричное) отношение в математике — если {х Ry и у Rx => х = у) р,ля всех х и у из А (Рис. 3).

Рис. 3



Формально, отношение антисимметрично, если матрица не имеет ни одной симметричной единицы относительно главной диагонали. Проверка подобна симметрично (Рис. 4 – наглядная демонстрация).

Рис. 4

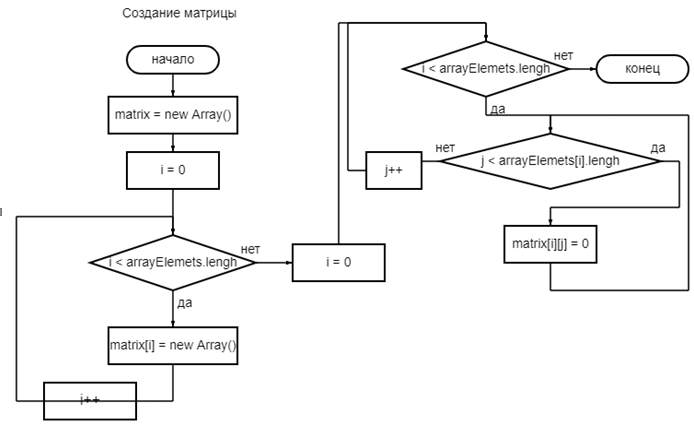


Транзитивное отношение в математике — если {х R y и у R z => х R z) для любой тройки элементов x^y^z Е А.

Транзитивное отношение в теории множеств - это такое отношение, при котором если один элемент упорядоченной пары соотносится с вторым, а второй элемент другой упорядоченной пары с третьим, то и первый элемент соотносится с третьим и образуют третью упорядоченную пару.

Формально, отношение рефлексивно, если некоторый элемент соотносится со вторым, а второй с третьим, то первый можно соотнести с третьим (и подобные случаи 2->3->4->5 => 2->5 или 3-> 5).

Рис. 5 – Создание матрицы



Для определения отношения на матрице используются булевы переменные (Листинг 1), от которых зависит и вывод.

Листинг 1

let **reflective** = true;

let **symmetric** = true;

let **antisymmetric** = true;

let **transitive** = true;

let multiplicationMatrix = **multiplyMatrix(matrix)**;

let outputMultiplication = "";

for(let i = 0; i < arrayElements.length; i++){...}

for (let i = 0; i < matrix.length; i++) {

for (let j = 0; j < matrix[i].length; j++) {

if (matrix[i][j] == 0 && i === j) {

reflective = false; //Рефлексивность

}

if (matrix[i][j] != matrix[j][i]) {... //Симметричность}

if (!(!((matrix[i][j] == 1) && (i != j)) || matrix[j][i] == 0)) {... //Кососимметричность}

if (matrix[i][j] == 0 && multiplicationMatrix[i][j] == 1) {... // Транзитивность}

}

}

Для определения транзитивности нужна умноженная матрица, поэтому нужна функция для умножения матрицы - function multiplyMatrix(matrix) (Листинг 2).

Листинг 2

function **multiplyMatrix(matrix)** {

let array = [];

for (let i = 0; i < matrix.length; i++) {

array[i] = [];

}

for (let k = 0; k < matrix.length; k++) {

for (let i = 0; i < matrix.length; i++) {

let p = 0;

for (let j = 0; j < matrix.length; j++) {

p += matrix[i][j] \* matrix[j][k];

}

array[i][k] = p;

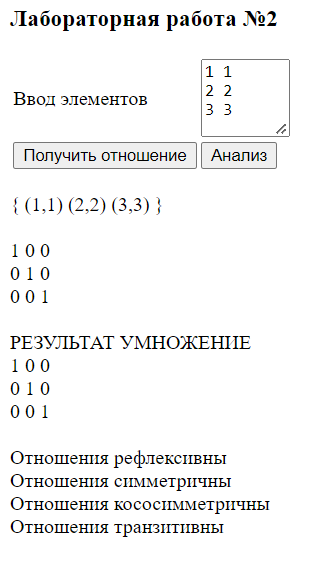
}

}

return array;

}

Интерфейс данной программы состоит из поля ввода и кнопок для вывода отношения и результата анализа отношений (Рис. 6). В поле элементы вводятся попарно.



# Листинг кода.

let error;

/\*\*

\* функция, которая выводит значение после проверки на корректность

\*/

function getValue(){

let outPut = "{";

let text = document.getElementById('array');

let array = writeArray(text);

if(array.length > 0) {

for(let i = 0; i < array.length; i++) {

if (array[i].length != 2) {

error = "В строке должно находиться ДВА элемента";

array = false;

alert(error);

}

}

}else{

alert(error);

}

for(let i = 0; i < array.length; i++){

outPut += " (" + array[i] + ") ";

}

outPut += "}";

document.getElementById("Value").innerText = outPut;

}

/\*\*

\* функция записи пар элементов в виде двумерного массива

\*/

function writeArray(text){

let array;

let str;

array = false;

if(text.value.length > 0) {

array = new Array();

str = text.value.split('\n');

for(let i = 0; i < str.length; i++){

if (str[i].length > 0) {

array[i] = str[i].split(' ');

} else{

array = false;

error = "Одна из строк пустая"

}

Number(array);

}

} else{

array = true;

error = "Введите хотя бы ОДНУ пару элементов";

}

return array;

}

/\*\*

\* Удаление повторяющихся элементов множеств

\*/

function DeleteRepeatElements(array)

{

array.sort();

for(let i = 0; i < array.length - 1; i++)

{

if(array[i] == array[i+1])

{

array.splice(i,1); // удаление элемента массива, а на место его массив со следующим индексом

i--;

}

}

}

/\*\*

\* функция, умножающая матрицу на саму себя

\*/

function multiplyMatrix(matrix) {

let array = [];

for (let i = 0; i < matrix.length; i++) {

array[i] = [];

}

for (let k = 0; k < matrix.length; k++) {

for (let i = 0; i < matrix.length; i++) {

let p = 0;

for (let j = 0; j < matrix.length; j++) {

p += matrix[i][j] \* matrix[j][k];

}

array[i][k] = p;

}

}

return array;

}

/\*\*

\* функция, которая анализирует отношения и составляет матрицу отношений

\*/

function analysis() {

let text = document.getElementById('array');

let array = writeArray(text);

getValue();

let arrayElements = new Array();

let c = 0;

for(let i = 0; i < array.length; i++){

for(let j = 0; j < array[i].length; j++){

arrayElements[i+j+c] = array[i][j];

}

c++;

}

DeleteRepeatElements(arrayElements);

let matrix = new Array();

for(let i = 0; i < arrayElements.length; i++) {

matrix[i] = new Array();

}

for(let i = 0; i < arrayElements.length; i++) {

for(let j = 0; j < arrayElements.length; j++){

matrix[i][j] = 0;

}

}

let arrayKey = {};

for(let i = 0; i < arrayElements.length; i++){

arrayKey[i] = {num : arrayElements[i]};

}

/\*\*

\* выставление единиц на места

\*/

for(let i = 0; i < array.length; i++){

for(let a = 0; a < arrayElements.length; a++){

for(let b = 0; b < arrayElements.length; b++){

if (array[i][0] === arrayKey[a].num && array[i][1] === arrayKey[b].num){

matrix[a][b] = 1;

}

}

}

}

let output = "";

for(let i = 0; i < arrayElements.length; i++){

for(let j = 0; j < arrayElements.length;j++){

output += matrix[i][j] + " ";

}

output += "\n";

}

let reflective = true;

let symmetric = true;

let antisymmetric = true;

let transitive = true;

let multiplicationMatrix = multiplyMatrix(matrix);

let outputMultiplication = "";

for(let i = 0; i < arrayElements.length; i++){

for(let j = 0; j < arrayElements.length;j++){

outputMultiplication += multiplicationMatrix[i][j] + " ";

}

outputMultiplication += "\n";

}

for (let i = 0; i < matrix.length; i++) {

for (let j = 0; j < matrix[i].length; j++) {

if (matrix[i][j] == 0 && i === j) {

reflective = false;

}

if (matrix[i][j] != matrix[j][i]) {

symmetric = false;

}

if (!(!((matrix[i][j] == 1) && (i != j)) || matrix[j][i] == 0)) {

antisymmetric = false;

}

if (matrix[i][j] == 0 && multiplicationMatrix[i][j] == 1) {

transitive = false;

}

}

}

let reflectiveMessage = "";

let symmetricMessage = "";

let antisymmetricMessage = "";

let transitiveMessage = "";

if (reflective == true){

reflectiveMessage = "Отношения рефлексивны";

} else{

reflectiveMessage = "Отношения не рефлексивны";

}

if (symmetric == true){

symmetricMessage = "Отношения симметричны";

} else{

symmetricMessage = "Отношения не симметричны";

}

if (antisymmetric == true){

antisymmetricMessage = "Отношения кососимметричны";

} else{

antisymmetricMessage = "Отношения не кососимметричны";

}

if (transitive == true){

transitiveMessage = "Отношения транзитивны";

} else{

transitiveMessage = "Отношения не транзитивын";

}

document.getElementById("Analysis").innerText = output + "\nРЕЗУЛЬТАТ\ " +

"УМНОЖЕНИЕ\n" + outputMultiplication + "\n" + reflectiveMessage + "\n" + symmetricMessage + "\n"

+ antisymmetricMessage + "\n" + transitiveMessage;

}

# Выводы.

Во время выполнения лабораторной работы, были изучены свойства бинарных отношений, разработана программа для их определения на матрице. Для разработки был использован язык JS.

# Список литературы.

1. Элементы математической логики. Отношения - <https://ppt-online.org/606090>.
2. Рефлексивные отношения - <https://hmong.ru/wiki/Reflexive_relation>.
3. Транзитивные отношения - <https://studfile.net/preview/4229244/page:3/>
4. Справочник по JavaScript - https://javascript.ru/manual