Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

Дисциплина «Математическая логика и дискретная математика»

Лабораторная работа №2

«**Свойства отношений**»

Выполнил студент

группы ИВТАПбд-12

Матюнин И.С.

Ульяновск, 2022

Оглавление

[Цель Работы 3](#_Toc103942648)

[Описание метода работы 3](#_Toc103942649)

[Рефлексивность 7](#_Toc103942650)

[Симметричность 8](#_Toc103942651)

[Кососимметри́чность 9](#_Toc103942652)

[Транзитивность 10](#_Toc103942653)

[Вывод результата 11](#_Toc103942654)

[Вывод 11](#_Toc103942655)

[Код: 12](#_Toc103942656)

[Список использованной Литературы 16](#_Toc103942657)

# **Цель Работы**

Разработать алгоритм определений свойств отношений на языке JavaScript, определяющую свойства отношения. Формат ввода определяется заданием. Необходимо выполнить проверку корректности ввода отношения.

Определяемые свойства:

* рефлексивность
* симметричность
* кососимметричность
* транзитивность

# **Описание метода работы**

**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.

На странице лабораторной работы создаём форму для ввода данных.

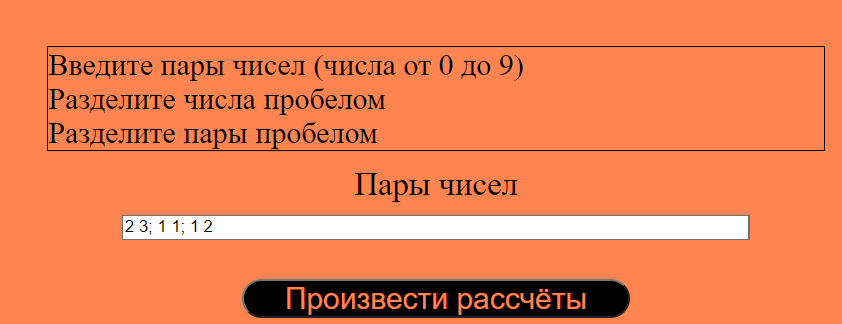
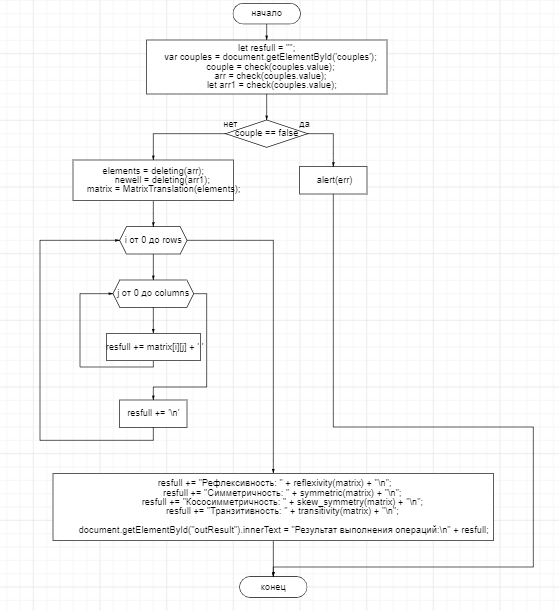


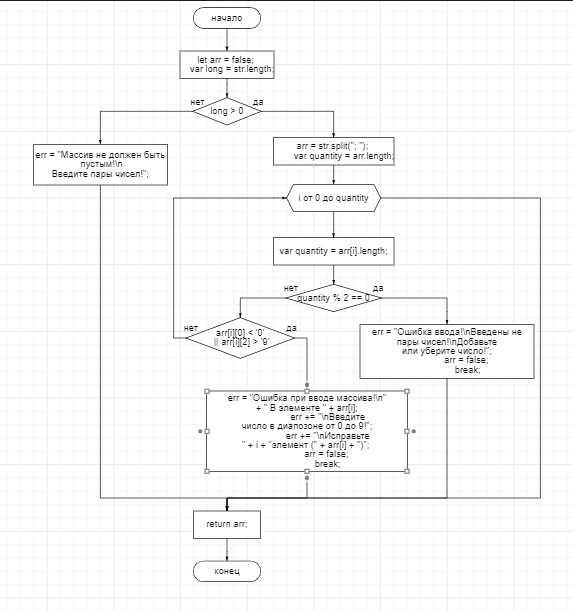
Рис. 1 – форма ввода пар чисел.

По нажатию на кнопку вызывается функция обработки введённых данных. В этой функции вызываются все остальные необходимые для работы функции. А именно, функция проверки ввода данных (валидации), функция построения бинарной матрицы на основании введённых пар, функции проверки свойств отношения уже построенной матрицы.



Блок-схема 1 – основная функция.

В первую очередь производится проверка корректности ввода пар чисел. Условие правильного ввода - это разделение чисел через пробел и отделение пар «;».

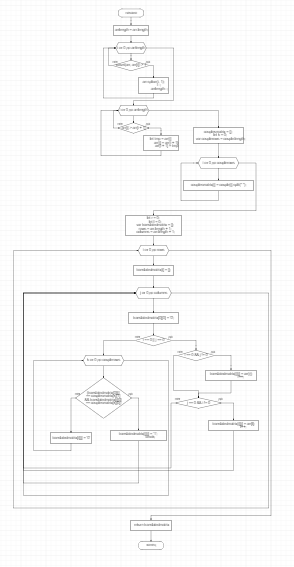


Блок-схема 2 – проверка ввода данных.

Если при воде была допущена ошибка, пользователю выводится сообщение о ней. Иначе строится бинарная матрица, и начинают проверяться свойства бинарного отношения.

**Построение бинарной матрицы**

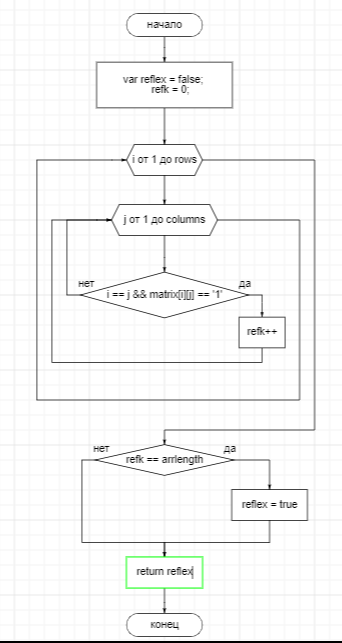
Чтобы проще было определить свойства пар чисел, я решил построить бинарную матрицу по следующему принципу. Сначала в матрицу вносятся по одному элементу пары. Предварительно все эти элементы были внесены в отдельный массив и удалены все повторяющиеся элементы так, что их осталось по одному. После, на пересечении элементов пары ставится «1» или «0» в зависимости от существования пары. Если пара была введена и существует, то ставится «1», иначе – «0». Получим квадратную бинарную матрицу.



Блок-схема 3 – построение бинарной матрицы.

## **Рефлексивность**

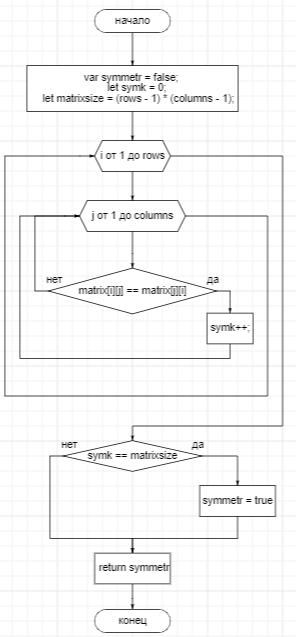
**Рефлексивное** **отношение** в математике - это такое отношение, что любой элемент всегда соотносится с самим собой.



Блок-схема 4 – функция проверки рефлексивности.

## **Симметричность**

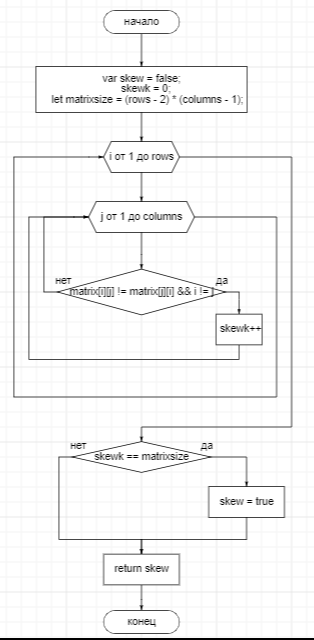
**Симметричное отношение** в математике - это бинарное отношение, не зависящее от порядка аргументов.



Блок-схема 5 – функция проверки симметричности

## **Кососимметри́чность**

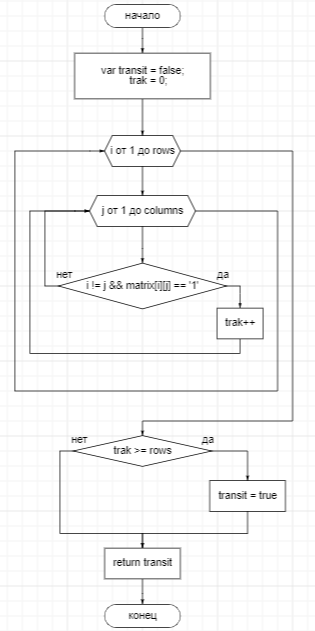
**Кососимметри́чность (или антисимметричность по паре данных аргументов)** — свойство математического объекта, являющегося функцией нескольких аргументов, менять знак (получать множитель −1) при перестановке каких-либо двух аргументов. Например, некоторые квадратные матрицы кососимметричны (антисимметричны) по отношению к перестановке индексов (то есть транспонированию: AT=−A, или Aij=−Aji).



Блок-схема 6 – функция проверки кососимметричности.

## **Транзитивность**

**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.



Блок-схема 7 – функция проверки транзитивности.

## **Вывод результата**

После всех операций на странице под кнопкой выводится результат. Если свойство выполняется, то пишется «true», иначе – «false».

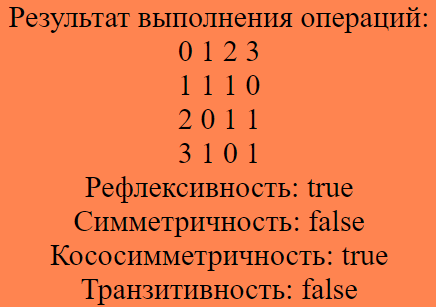


Рис. 2 – вывод результатов

## **Вывод**

В ходе данной лабораторной работы была разработана программа определения свойств бинарного отношения, изучены сами свойства бинарного отношения: транзитивность, симметричность, кососимметричность и рефлексивность.

# **Код:**

var err, matrix, couple, rows, columns, arr, couplesmatrix, arrlength, elements, newell;

//проверка ввода

function check(str) {

    let arr = false;

    var long = str.length;

    if (long > 0) {

        arr = str.split("; ");

        var quantity = arr.length;

        //Проверка на ввод

        for (let i = 0; i < quantity; i++) {

            var quantity = arr[i].length;

            if (quantity % 2 == 0) {

                err = "Ошибка ввода!\nВведены не пары чисел!\nДобавьте или уберите число!";

                arr = false;

                break;

            }

            //проверка, что введены числа

            if (arr[i][0] < '0' || arr[i][2] > '9') {

                err = "Ошибка при вводе массива!\n" + " В элементе " + arr[i];

                err += "\nВведите число в диапозоне от 0 до 9!";

                err += "\nИсправьте" + i + "элемент (" + arr[i] + ")";

                arr = false;

                break;

            }

        }

    } else {

        err = "Массив не должен быть пустым!\nВведите пары чисел!";

    }

    return arr;

}

//счётчик элементов

function count(arr, el) {

    let k = 0;

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

        if (arr[i] == el) {

            k++;

        }

    }

    return k;

}

//вывод всех чисел в одномерный массив

function deleting(arr) {

    var elemarr = [];

    let q = 0;

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

        arr[i] = arr[i].split(" ");

        for (let j = 0; j < 2; j++) {

            elemarr[q] = arr[i][j];

            q++

        }

    }

    return elemarr;

}

//выделение элементов массива по одному разу и постороение матрицы из нулей и единиц

function MatrixTranslation(arr) {

    arrlength = arr.length;

    for (let i = 0; i < arrlength; i++) {

        if (count(arr, arr[i]) > 1) {

            arr.splice(i, 1);

            i--;

            arrlength--;

        }

    }

    for (let i = 0; i < arrlength; i++) {

        if (arr[i] > arr[i + 1]) {

            let tmp = arr[i]

            arr[i] = arr[i + 1];

            arr[i + 1] = tmp;

        }

    }

    couplesmatrix = [];

    let k = 0;

    var couplerows = couple.length;

    for (let i = 0; i < couplerows; i++) {

        couplesmatrix[i] = couple[i].split(" ");

    }

    let r = 0;

    let l = 0;

    var translatedmatrix = [];

    rows = arr.length + 1;

    columns = arr.length + 1;

    for (let i = 0; i < rows; i++) {

        translatedmatrix[i] = [];

        for (let j = 0; j < columns; j++) {

            translatedmatrix[0][0] = '0';

            if (i == 0 || j == 0) {

                if (i == 0 && j != 0) {

                    translatedmatrix[i][j] = arr[r];

                    r++;

                }

                if (j == 0 && i != 0) {

                    translatedmatrix[i][j] = arr[l];

                    l++;

                }

            } else {

                for (k = 0; k < couplerows; k++) {

                    if ((translatedmatrix[0][j] == couplesmatrix[k][1] && translatedmatrix[i][0] == couplesmatrix[k][0])) {

                        translatedmatrix[i][j] = '1';

                        break;

                    } else {

                        translatedmatrix[i][j] = '0';

                    }

                }

            }

        }

    }

    return translatedmatrix;

}

//рефлексивность

function reflexivity(matrix) {

    var reflex = false;

    refk = 0;

    for (let i = 1; i < rows; i++) {

        for (let j = 1; j < columns; j++) {

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

                refk++;

            }

        }

    }

    if (refk == arrlength) {

        reflex = true;

    }

    return reflex;

}

function symmetric(matrix) {

    var symmetr = false;

    let symk = 0;

    let matrixsize = (rows - 1) \* (columns - 1);

    for (let i = 1; i < rows; i++) {

        for (let j = 1; j < columns; j++) {

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

                symk++;

            }

        }

    }

    if (symk == matrixsize) {

        symmetr = true;

    }

    return symmetr;

}

var skewk;

var matrixsize;

function skew\_symmetry(matrix) {

    var skew = false;

    skewk = 0;

    let matrixsize = (rows - 2) \* (columns - 1);

    for (let i = 1; i < rows; i++) {

        for (let j = 1; j < columns; j++) {

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

                skewk++;

            }

        }

    }

    if (skewk == matrixsize) {

        skew = true;

    }

    return skew;

}

function transitivity(matrix) {

    var transit = false;

    trak = 0;

    for (let i = 1; i < rows; i++) {

        for (let j = 1; j < columns; j++) {

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

                trak++;

            }

        }

    }

    if (trak >= rows) {

        transit = true;

    }

    return transit;

}

//Основая функция

function res() {

    let resfull = "";

    var couples = document.getElementById('couples');

    couple = check(couples.value);

    arr = check(couples.value);

    let arr1 = check(couples.value);

    if (couple == false) {

        alert(err);

    } else {

        elements = deleting(arr);

        newell = deleting(arr1);

        matrix = MatrixTranslation(elements);

        for (let i = 0; i < rows; i++) {

            for (let j = 0; j < columns; j++) {

                resfull += matrix[i][j] + ' ';

            }

            resfull += '\n';

        }

        resfull += "Рефлексивность: " + reflexivity(matrix) + "\n";

        resfull += "Симметричность: " + symmetric(matrix) + "\n";

        resfull += "Кососимметричность: " + skew\_symmetry(matrix) + "\n";

        resfull += "Транзитивность: " + transitivity(matrix) + "\n";

        document.getElementById("outResult").innerText = "Результат выполнения операций:\n" + resfull;

    }

}

# **Список использованной Литературы**

* Яблонский С.В. Введение в дискретную математику. – М.: Высш. шк., 2003. – 384 с.
* Даккетт, Джон. HTML и CSS. Разработка и дизайн веб-сайтов/ Джон Дакетт; [пер. с англ. М.А.Райтмана]. - Москва: Эксмо, 2019. - 480 с. - (Мировой компьютерный бестселлер).
* Даккетт, Джон. JavaScript и jQuery. Интерактивная веб-разработка / Джон Дакетт; [пер. с англ. М.А.Райтмана]. - Москва: Эксмо, 2019. - 640 с.: ил. - (Мировой компьютерный бестселлер).