Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Ульяновский государственный технический университет»

Кафедра: «Вычислительная техника»

Дисциплина: «Математическая логика и дискретная математика»

Лабораторная работа №3

«Свойства отношений»

Выполнил студент

группы ИВТАСбд-12

Давыдов Д. Д.

Проверил

Игонин А. Г.

Ульяновск, 2022

**Цель работы**

Требуется написать программу, которая определяет – является ли отношение функцией. Отношения вводятся по парам элементов. А также вводятся все элементы двух множеств. Необходимо использовать JS.

**Описание метода выполнения**

Отношение рефлексивно, если для любых х из множества А существуют отношение xRx. Отношение симметрично, если для любых х из множества А существуют отношение xRy и yRx. Отношение симметрично, если для любых х и y из множества А существует xRy, из которого следует yRx. Отношение кососимметрично, если для любых х и у из множества А существует xRy и yRx, из которых следует, что x=y. Отношение транзитивно, если для любой тройки элементов x, y, z из А и таких бинарных отношений, как xRy и yRx следует xRz.

Для того чтобы пользователь мог вводить данные необходимо создать html страницу, состоящую из трёх форм и кнопок взаимодействия. Код алгоритма был написан на JS.

Пользователь вводит элементы первого множества, второго множества и пары элементов отношения через точку с запятой. После нажатия кнопки «Анализ», происходит удаление лишних пробелов, заполнение массивов и встроенная проверка.

let relations = relationInput.value.split(/;+\s\*/).filter( e => e.trim().length > 0);

let firstSet = new Set(firstSetInput.value.split(/(\s+)/).filter( e => e.trim().length > 0));

let secondSet = new Set(secondSetInput.value.split(/(\s+)/).filter( e => e.trim().length > 0));

let isOkay = true;

let realtionMatrixT = new Array(firstSet.size).fill().map(() => new Array(secondSet.size));

relations = relations.map(el=>{

let t = el.split(/(\s+)/).filter( e => e.trim().length > 0);

if (t.length != 2 || !(firstSet.has(t[0]) && secondSet.has(t[1]))) isOkay = false;

return t;

})

if(!isOkay){

console.log(relations)

alert("Была допущена ошибка при вводе, попробуйте ещё раз.");

return;

}

После этого происходит создание матрицы отношения.

let i = 0, j = 0;

firstSet.forEach(el => {

secondSet.forEach(el2 => {

relations.forEach(el3=>{

if (el3[0] == el && el3[1] == el2){

realtionMatrixT[i][j] = 1;

}

else if (realtionMatrixT[i][j] != 1){

realtionMatrixT[i][j] = 0;

}

})

j++;

})

i++;

j = 0;

});

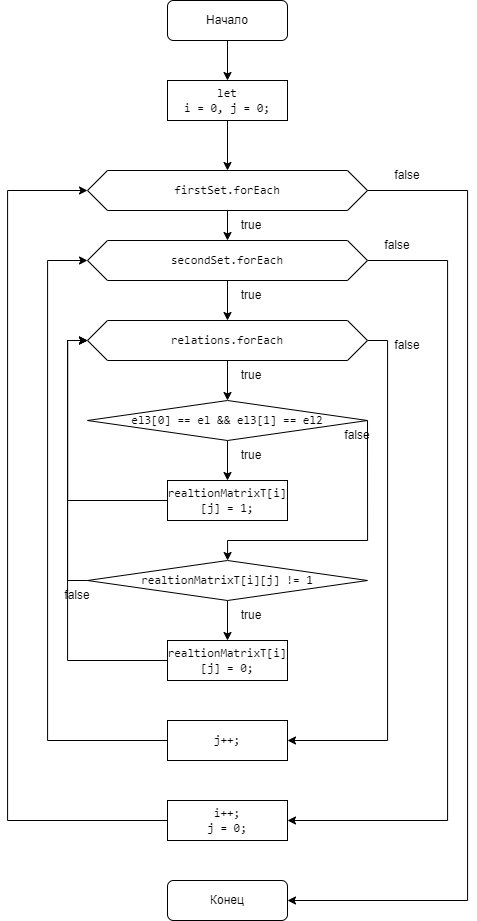


Рисунок 1. Блок-схема формирование матрицы отношения

В получившейся матрице по вертикале располагаются элементы первого множества, а по горизонтали – второго.

После того, как матрица отношений сформировалась, мы можем проверить её на функцию. Для этого необходимо сосчитать суммы элементов каждой строки отдельно. Если они все равны 1, то это отношение функция.

let isFunction = true;

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

let t = 0;

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

t += relationMatrix[i][j];

}

if (t != 1){

isFunction = false;

break;

}

}

workZone.style.display = "block";

isFunctionTxt.innerHTML = isFunction ? "является функцией." : "не является функцией";

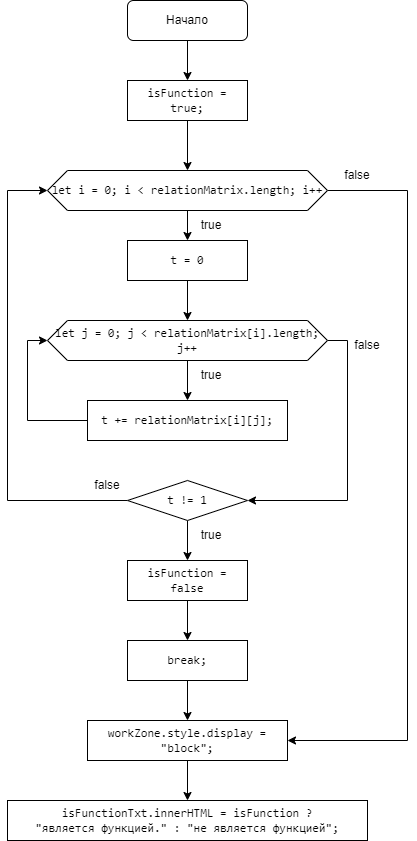


Рисунок 2. Блок-схема проверки на функцию

**Заключение**

Выполнив данную лабораторную работу, я создал сайт, который может проверять отношение на функцию и также проверять ввод на ошибки.