Министерство образования и науки Российской Федерации

ФГБОУ ВО Ульяновский государственный технический университет

Кафедра «Вычислительная техника»

Лабораторная работа №2

«Свойства отношений»

Выполнил студент

группы ИВТАСбд-11

Абросимов К. С.

Ульяновск, 2022

Содержание

[1. Цель работы 3](#_Toc103792993)

[2. Описание метода выполнения 3](#_Toc103792994)

[2.1 Теория 3](#_Toc103792995)

[2.2 Файл «index» 4](#_Toc103792996)

[2.3 Файл «mldmLab2.php» 5](#_Toc103792997)

[2.4 Файл «scriptslab2.js» 6](#_Toc103792998)

[3. Выводы 14](#_Toc103792999)

[4. Список литературы 14](#_Toc103793000)

1. **Цель работы**

Необходимо написать программу, которая будет определять свойства отношения, введенного бинарной матрицей (состоящей из нулей и единиц), причём матрица должна быть размера n\*n. Определяться должны следующие отношения: рефлексивность, симметричность, кососимметричность (антисимметричность) и транзитивность. Пользователь должен иметь возможность вводить данные, а также должна присутствовать проверка на корректность введённых данных.

1. **Описание метода выполнения**

## **2.1 Теория**

Бинарным отношением между элементами множества A и B называется любое подмножество R множества AxB, то есть R ℂ A x B. 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.

Если же с матрицами, то

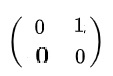
Рефлексивность:



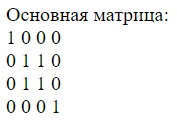
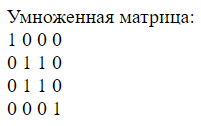
Симметричность:



Антисимметричность – идёт только в одну сторону относительно главной диагонали:



Транзитивность – матрица, умноженная на саму себя, всё равно равна изначальной.

Код проводимых операций будет написан на PHP. PHP - это язык программирования, специально разработанный для написания web-приложений (сценариев), исполняющихся на Web-сервере. PHP взаимодействует с html.

## **2.2 Файл «index»**

Первый файл «index», является html страницей, которая имеет ссылки (переходы) на другие страницы – лабораторные работы.

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Лабораторные работы</title> - основной заголовок  
 <script type="text/javascript" src="/scripts/scripts.js"></script> - ссылка на папку со скриптами  
</head>  
<body>  
 <h1> Мои лабораторные работы </h1>   
 <a href ="mldmLab1.php">Первая лабораторная работа по МЛиДМ</a><br>  
 <a href ="mldmLab2.php">Вторая лабораторная работа по МЛиДМ</a> - ссылки к другим лабораторным работам  
</body>  
</html>

## **2.3 Файл «mldmLab2.php»**

Для того, чтобы пользователь мог вводить данные необходимо написать небольшую html-структуру, состоящую поля для ввода (input), полей для вывода и кнопки, по которой произойдут операции над множествами. Также для полного вывода операций нужно создать значение для вывода.

Как только пользователь введёт бинарную матрицу и нажмёт кнопку «Сделать расчёт», то программа выполнит функцию «rasschetmatrix», ссылаясь на файл «scriptslab2.js», а затем выведет либо полученный результат в виде умноженной матрицы (Matrix) и полной характеристики бинарной матрицы (его свойств – outResult), либо экран с текстом ошибки (error\_text).

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Лабораторная работа 2</title> - заголовок  
 <script type="text/javascript" src="/scripts/scriptslab2.js"></script> - ссылка на другой файл с функциями  
</head>  
<body>  
<h1> Лабораторная работа №2 </h1>  
<form>  
 <table>  
 <tr>  
 <td> Введите бинарную матрицу n\*n </td>  
 <td> <textarea id="mass" value="" rows = "10" cols = "20" size=""></textarea></td> - поле для ввода  
 </tr>  
 <tr>  
 <td colspan="2"> <input type="button" value="Сделать расчёт" onclick="rasschetmatrix();"/></td> - кнопка для выполнения расчётов  
 </tr>  
 </table>  
</form>  
<div id ="Matrix"></div>  
<div id ="outResult"></div> - вывод результатов  
</body>  
</html>

## **2.4 Файл «scriptslab2.js»**

Как и было сказано, выполняется функция “rasschetmatrix”. Сначала эта функция создаёт переменные для считывания информации, введённое пользователем. Но они не являются массивом, а значением. После преобразования значения в двумерный массив с помощью функции «needmass» (об этом позже), проверяем массив на правильность ввода (по условиям задачи и с помощью функции validate). После этого проводится функция с умножением матрицы на саму себя (multiplyMatrix), затем вывод основной и умноженной матрицы в формате String (путём создания новых двумерных массивов, копирования цикличным методом) и последующее определение свойств матрицы. После всех этих действий, переменная типа String добавляет нужный текст (свойства матрицы) по условиям и переносит значение в предыдущий файл для вывода в страницу.

function rasschetmatrix()  
{  
 var a = ***document***.getElementById('mass'); - получение значения из предыдущего файла  
 var massmat = [];  
 massmat = needmass( a.value ); - функция преобразования значения в матрицу  
 if(validate(massmat) == false) – проверка ввода  
 {  
 alert(***error\_text***); - вывод ошибки при условии при определённых условиях  
 }  
 else {  
 let doublemat = multiplyMatrix(massmat); - новая матрица, умноженная на саму себя  
 let matrixes = ""; - будущий вывод матриц  
 for (let i = 0; i < massmat.length; i++) {  
 for (let j = 0; j < massmat[i].length; j++) {  
 matrixes += massmat[i][j] + " "; - создаём матрицу формата String для вывода в сайт, циклично добавляя туда элементы   
 }  
 matrixes += "\n"; - следующая строка  
 }  
 let doublematrixes = "";  
 for (let i = 0; i < doublemat.length; i++) {  
 for (let j = 0; j < doublemat[i].length; j++) {  
 doublematrixes += doublemat[i][j] + " "; - тоже самое, что и с предыдущей матрицей  
 }  
 doublematrixes += "\n";  
 }  
 ***document***.getElementById('Matrix').innerText = "Основная матрица:\n" + matrixes +  
 "\n" + "Умноженная матрица:\n" + doublematrixes + "\n"; - добавление полученных матриц к общему значению «Matrix»

Для определения свойств отношения я использовал два цикла, один из которых проверяет условия каждого из свойств. Если условие было истинным, то особым переменным типа Boolean присваивается значение «false», которое впоследствии и определит вывод сообщения на странице.  
 let antiSym = true;  
 let refl = true;  
 let sym = true;  
 let tranz = true; - особые переменные, определяющие вывод сообщения  
 for(let i = 0; i < massmat.length; i++) { - основная работа над свойствами матрицы именно здесь:  
 for(let j = 0; j < massmat.length; j++) {  
 if (!(!((massmat[i][j] == 1) && (i != j)) || massmat[j][i] == 0)) {  
 antiSym = false; - если относительно главной диагонали пара элементов матрицы равны или равны отношению 1 -> 0 (считая, что идём от нижнего левого к правому верхнему краю), то матрица не антисимметрична  
 }  
 if (i == j) {  
 if (massmat[i][j] == 0) { - если хоть один элемент главной диагонали матрицы равна нулю, то матрица не рефлективна  
 refl = false;  
 }  
 }else {  
 if(massmat[i][j] != massmat[j][i]) { - если элементы относительно главной диагонали не равны, то матрица не симметрична  
 sym = false;  
 }  
 }  
 if ((massmat[i][j] == 0 && doublemat[i][j] == 1) || (massmat[i][j] == 1 && doublemat[i][j] == 0)) { - если элемент основной матрицы не совпадает с элементов умноженной матрицы, то матрица не транзитивна  
 tranz = false;  
 }  
 }  
 }

Впоследствии к значению типа String добавляются сообщения, которые выводятся соответственно с условием особых переменных типа Boolean.   
 let full\_result = ""; - будущий вывод свойств матрицы  
 if (refl == true) {  
 full\_result = "Данная матрица рефлексивна\n";  
 }  
 else {  
 full\_result = "Данная матрица не рефлексивна\n";  
 }  
 if (sym == true) {  
 full\_result += "Данная матрица симметрична\n";  
 }  
 else {  
 full\_result += "Данная матрица не симметрична\n";  
 }  
 if (antiSym == true) {  
 full\_result += "Данная матрица антисимметрична\n";  
 }  
 else {  
 full\_result += "Данная матрица не антисимметрична\n";  
 }  
 if (tranz == true) {  
 full\_result += "Данная матрица транзитивна\n";  
 }  
 else {  
 full\_result += "Данная матрица не транзитивна\n";  
 }  
 ***document***.getElementById('outResult').innerText = "Свойства матрицы:\n" + full\_result; - вывод в страницу  
  
 }  
}

Функция needmass создаёт переменную (будущий массив, который будем возвращать), присваиваем ему значение с «rasschetmatrix», с помощью функции «str.split» по пробелам и переносам «\n» делим значение на нормальный двумерный массив. Однако пользователь мог написать при вводе данных дополнительные пробелы и переносы, которые тоже будут считаться как отдельный элемент (пустой). Для этого, пользуясь циклом, выполняю функцию «splice» - удаление элемента, если найден пустой элемент, сперва для переносов, затем для пробелов.

function needmass( str )  
{  
 var massiv = [];  
 massiv = str.split("\n").map(function (x) { return x.split(" "); }); - делим значение по элементам  
  
 for (let i = 0; i < massiv.length; i++) {  
 if (massiv[i] == "" && massiv[i].length == 1) { - удаление лишних строк  
 massiv.splice(i, 1);  
 i--;  
 }  
 }  
 for (let i = 0; i < massiv.length; i++) {  
 for (let j = 0; j <= massiv[i].length; j++) { - удаление лишних пробелов  
 if (massiv[i][j] == "") {  
 massiv[i].splice(j, 1);  
 }  
 }  
 }  
 return massiv; - возвращаем полученный массив  
}

Функция validate проверяет полученный массив на 3 условия:

* Что он не пустой
* Что его формат n\*n
* Что его значения только 0 и 1

Сперва функция создаёт переменную, в которую мы и присвоим весь полученный массив из «rasschetmatrix», и переменную типа Boolean «flag», которая будет отвечать за то, будет ли выполняться определение свойств матрицы или же выведется окно ошибки. Затем последовательно выполняется проверка вышесказанных условия. Если какая-то ошибка в вводе произошла, то сразу цикл прерывается и выводится окно с текстом ошибки.

function validate(str)

{

let mass = str;

let flag = true;

if(str.length>0) { - первое условие

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

if (mass.length != mass[i].length) { - второе условие, при котором сравниваются длины столбца и строки

error\_text = "Матрица неверного формата. Формат n\*n"; - добавление в значение типа String текста ошибки

flag = false;

break;

}

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

if (mass[i][j] != 1 && mass[i][j] != 0) { - третье условие

error\_text = "Матрица должна содержать только элементы 1 или 0";

flag = false;

break;

}

}

}

}

else {

error\_text = "Массив не может быть пустым";

flag = false;

}

return flag;

}

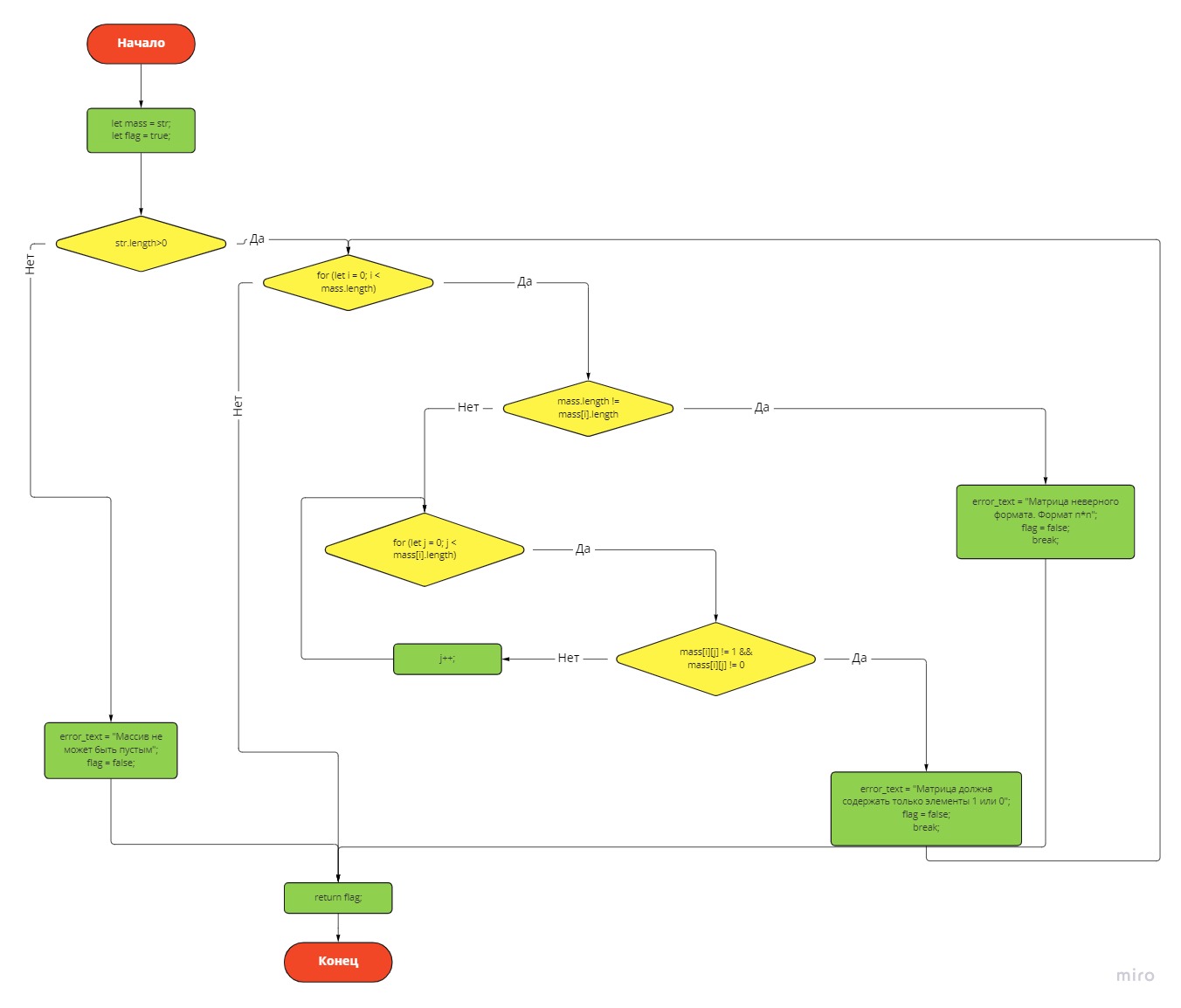


Рис. 1 - функция validate

Функция multiplyMatrix создаёт возвращаемый массив result, который циклом преобразуем в двумерный массив. Затем с помощью циклов и сбрасываемой переменной «elem» находится сумма умножений каждого элемента строки на соответствующий элемент столбца. Если конечная сумма больше или равна единице, то окончательный элемент результата присваивается единице, в противном случае нулю.

function multiplyMatrix(n)  
{  
 let result = [];  
 for (let i = 0; i < n.length; i++) {  
 result[i] = []; - создание двумерного массива  
 }  
 for (let i = 0; i < n.length; i++) {  
 for (let j = 0; j < n.length; j++) {  
 let elem = 0;  
 for (let k = 0; k < n.length; k++) {  
 elem += n[j][k] \* n[k][i]; - складываем общий результат из умножений элементов  
 }  
 if (elem >= 1) {  
 result[i][j] = 1;  
 } else {  
 result[i][j] = 0;  
 }  
 }  
 }  
 return result; - возвращаем конечную, умноженную матрицу  
}

1. **Выводы**

В результате проведения работы были изучены основы работы с двумерными массивами в JavaScript, изучено понятие отношения множеств и его свойства (рефлексивность, симметричность, антисимметричность, транзитивность).

# **Список литературы**

(б.д.).

*Metanit*. (б.д.). Получено из JavaScript | Введение в массивы: https://metanit.com/web/javascript/2.5.php

*ppt-online*. (б.д.). Получено из Теория множеств: https://ppt-online.org/5744