МИНОБРНАУКИ РОССИИ

ФГБОУ ВО Череповецкий государственный университет

Институт информационных технологий

Кафедра: Математическое и программное обеспечение ЭВМ

Дисциплина: Теория автоматов и формальных языков

ЛАБОРАТОРНАЯ РАБОТА № 3-4

Тема: Регулярные множества и регулярные выражения

Выполнил:

студент гр. 1ИСб-00-1оп-21 Березин В.А.

Проверил: Ганичева О.Г.

Череповец, 2022 г.



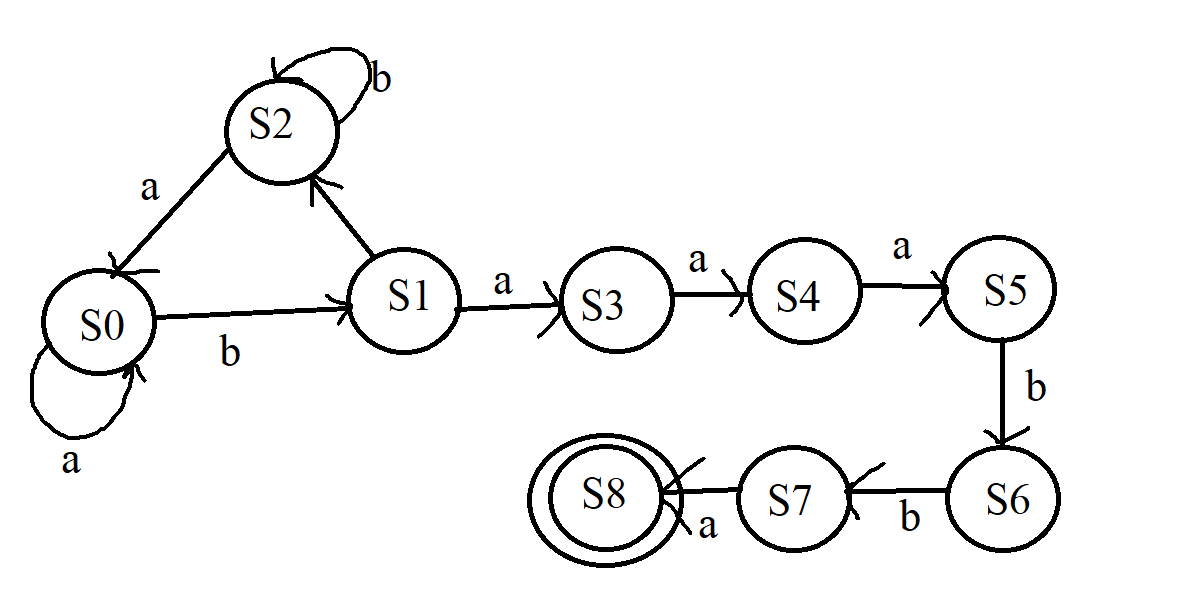


Е) X – {a, b};

S – {S0, S1,S2,S3,S4,S5,S6,S7,S8};

S0 – начальное состояние;

(a/b)\*aabba



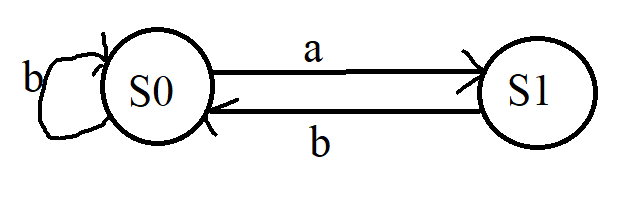
Ж) Все цепочки, включающие строчку aabba.

X – {a, b};

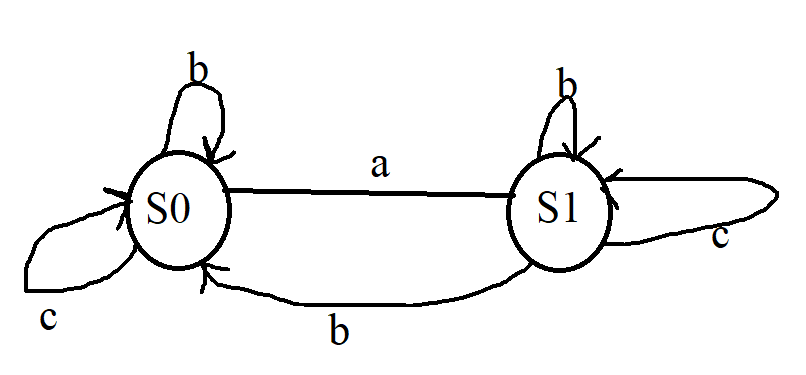
S – {S0, S1};

S0 – начальное состояние;

(b/ab)\* aabba+(b/ab)\*



**Задание 2.**   
Построить конечный автомат с входным алфавитом V – {a, b, c}, распознающий:  
а) Все цепочки в которых за каждым а когда-нибудь в будущем следует b:   
X – {a, b, c};  
S – {S0, S1, S2};  
S0 – начальное состояние;  
(c/b)\*(ac\*b(b/c)\*)\*



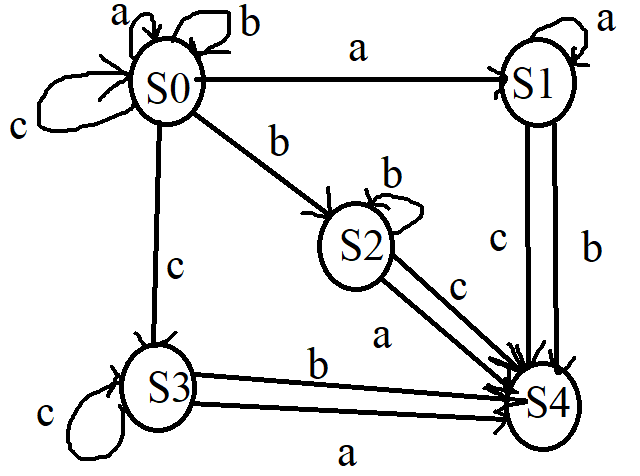
б) все цепочки в которых две последние буквы не совпадают:

X – {a, b, c};

S – {S0, S1, S2, S3, S4};

S0 – начальное состояние;

(a/b/c)\*(a(b/c)/c(a/b)/b(c/a))\*

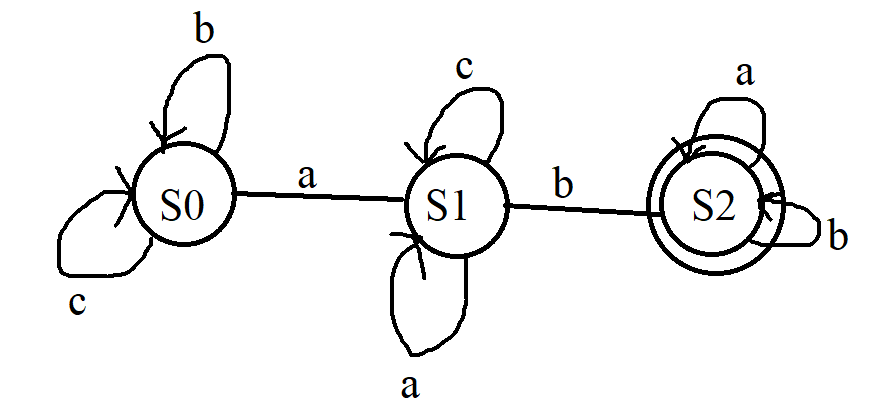


г) все цепочки, включающие по крайней мере один символ а и один символ b;

X – {a, b, c};

S – {S0, S1, S2};

S0 – начальное состояние;



Стр. 185 задания 19, 21

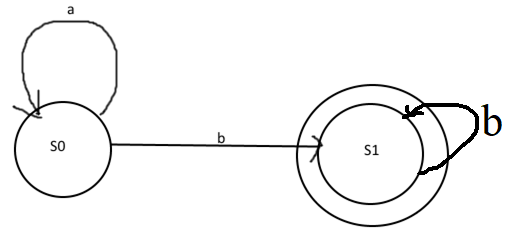
19.

X – {a, b};

S – {S0, S1};

S0 – начальное состояние;

a\*b\*

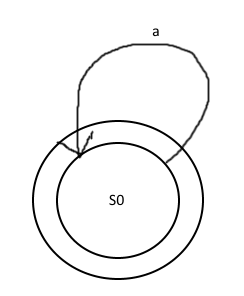


a\*a\*

X – {a};

S – {S0};

S0 – начальное состояние;

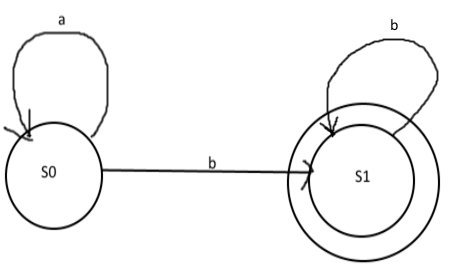


S0 – начальное состояние;

a\*+b\*

X – {a, b};

S – {S0, S1};

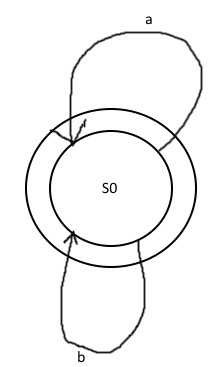


(a+b)\*

X – {a, b};

S – {S0};

S0 – начальное состояние;

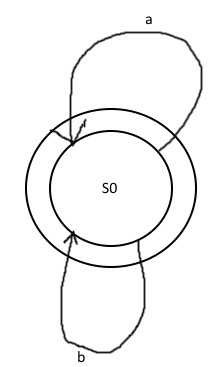


(a\*b\*)\*

X – {a, b};

S – {S0};

S0 – начальное состояние;

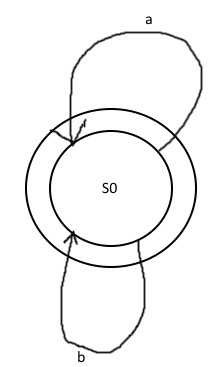


(a\*+b\*)\*

X – {a, b};

S – {S0};

S0 – начальное состояние;

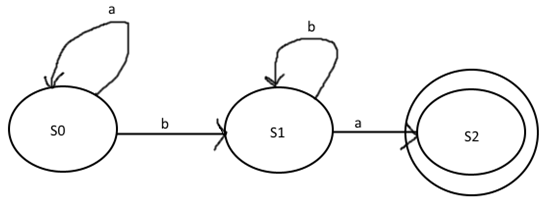


a\*b+b\*a

X – {a, b};

S – {S0,S1,S2};

S0 – начальное состояние;



21.

а) Построить регулярные выражения, задающих множество всех таких слов

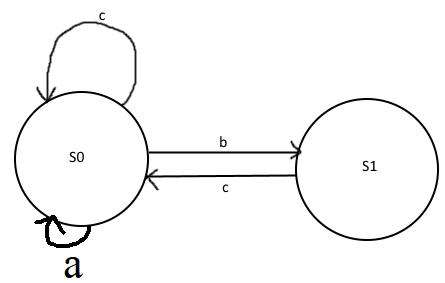
над словарем E = {a, b, c}, в которых за символом b обязательно следует символ

X – {a, b, c};

S – {S0, S1};

S0 – начальное состояние;

(a/c/bc)\*



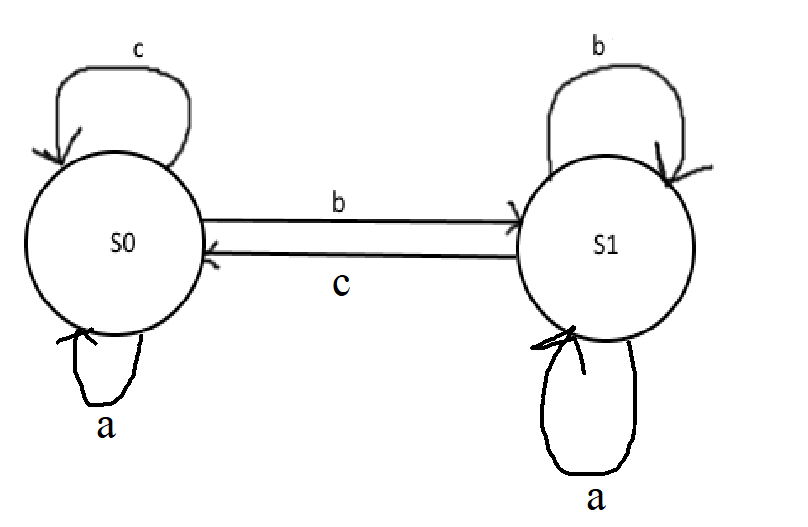
б) Построить регулярные выражения, задающих множество всех таких слов над словарем E = {a, b, c}, в которых за символом b не может находиться символ c;

X – {a, b, c};

S – {S0, S1};

S0 – начальное состояние;

(a/c)\*(b/a)\*



Код:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApplication1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

int S = 0; //переменная состояния

private void Form1\_Load(object sender, EventArgs e)

{

dataGridView1.Width = 240; //размеры таблицы

dataGridView1.Height = 150;

dataGridView1.Columns.Add("column-1", "t"); //создание 1 столбца

dataGridView1.Rows.Add("x"); //добавление строки "х"

dataGridView1.Rows.Add("S"); //добавление строки "S"

dataGridView1.Columns.Add("column-2", "0"); //создание 2 столбца

dataGridView1[1, 1].Value = "0"; //внесение значения в ячейку 1-го столбца 1 строки

label1.Text = "";

label2.Text = "";

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

int a = 1; // счетчик строк

int b = 1; // счетчик столбов

void KA(char symbol) //работа конечного автомата

{

switch (S)

{

case 0: //1 состояние

if (symbol == 'a') //проверка введенного символа

{

S = 0;

newcol();

dataGridView1[a + 1, b].Value = S; //внесение значения S в ячейку a+1 столбца и b строки

dataGridView1[a, b - 1].Value = "a";//внесение значения x в ячейку a столбца и b-1 строки

a++; //переход к новому столбцу

// в последующих состояниях проивзолятся такие же действия

}

else if (symbol == 'c')

{

S = 1;

newcol();

dataGridView1[a + 1, b].Value = S;

dataGridView1[a, b - 1].Value = "c";

a++;

}

else if (symbol == 'b')

{

S = 0;

newcol();

dataGridView1[a + 1, b].Value = S;

dataGridView1[a, b - 1].Value = "b";

a++;

}

else

{

MessageBox.Show("Входная последовательность некорректная");

S = -1;

}

break;

case 1:

if (symbol == 'a')

{

S = 2;

newcol();

dataGridView1[a + 1, b].Value = S;

dataGridView1[a, b - 1].Value = "a";

a++;

}

else {

MessageBox.Show("Входная последовательность некорректная");

S = -1;

}

break;

case 2:

if (symbol == 'b')

{

S = 3;

newcol();

dataGridView1[a + 1, b].Value = S;

dataGridView1[a, b - 1].Value = "b";

a++;

}

else

{

MessageBox.Show("Входная последовательность некорректная");

S = -1;

}

break;

case 3:

if (symbol == 'a')

{

S = 4;

newcol();

dataGridView1[a + 1, b].Value = S;

dataGridView1[a, b - 1].Value = "a";

a++;

}

else

{

MessageBox.Show("Входная последовательность некорректная");

S = -1;

}

break;

case 4:

if (symbol == 'a') //проверка введенного символа

{

S = 0;

newcol();

dataGridView1[a + 1, b].Value = S; //внесение значения S в ячейку a+1 столбца и b строки

dataGridView1[a, b - 1].Value = "a";//внесение значения x в ячейку a столбца и b-1 строки

a++; //переход к новому столбцу

// в последующих состояниях проивзолятся такие же действия

}

else if (symbol == 'c')

{

S = 1;

newcol();

dataGridView1[a + 1, b].Value = S;

dataGridView1[a, b - 1].Value = "c";

a++;

}

else if (symbol == 'b')

{

S = 0;

newcol();

dataGridView1[a + 1, b].Value = S;

dataGridView1[a, b - 1].Value = "b";

a++;

}

else

{

MessageBox.Show("Входная последовательность некорректная");

S = -1;

}

break;

default:

break;

}

}

void newcol() //создание нового столбца

{

int n = dataGridView1.Columns.Count - 1;

dataGridView1.Columns.Add(Convert.ToString(a), Convert.ToString(n));

}

void clear() //очистка таблицы

{

dataGridView1.Rows.Clear(); //удаление строк

for (int i = dataGridView1.Columns.Count; i >= 3; i--) //удаление столбцов

{

dataGridView1.Columns.RemoveAt(2);

}

dataGridView1.Rows.Add("x"); //возврат таблицы к изначальному состоянию

dataGridView1.Rows.Add("S");

dataGridView1[1, 1].Value = "0";

a = 1;

S = 0; //возврат к нулевому состоянию

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e) //ввод в таблицу

{

clear();

string posled = textBox1.Text; //взятие из текстбокса строки символов

for (int i = 0; i < posled.Length; i++) //проверка корректности символа

{

if (posled[i] != 'a' && posled[i] != 'b' && posled[i] != 'c')

{

MessageBox.Show("Введен некорректный символ");

return;

}

}

for (int i = 0; i < posled.Length; i++) //посимвольная передача символов

{

KA(posled[i]);

}

}

private void button2\_Click(object sender, EventArgs e)

{

label1.Text = "";

label2.Text = "";

string[] posled = richTextBox1.Text.Split('\n'); //взятие из текстбокса строки символов

for (int i = 0; i < posled.Length; i++) //проверка корректности символа

{

for (int j = 0; j < posled[i].Length; j++)

{

if (posled[i][j] != 'a' && posled[i][j] != 'b' && posled[i][j] != 'c')

{

MessageBox.Show("Введен некорректный символ");

return;

}

}

}

for (int i = 0; i < posled.Length; i++) //посимвольная передача символов

{

S = 0;

for (int j = 0; j < posled[i].Length; j++)

{

switch (S)

{

case 0: //0-ое состояние

if (posled[i][j] == 'a') //проверка введенного символа

{

S = 0;

}

if (posled[i][j] == 'c')

{

S = 1;

}

if (posled[i][j] == 'b')

{

S = 0;

}

break;

case 1:

if (posled[i][j] == 'a')

{

S = 2;

}

break;

case 2:

if (posled[i][j] == 'b')

{

S = 3;

}

break;

case 3:

if (posled[i][j] == 'a')

{

S = 4;

}

break;

case 4: //4-ое состояние

if (posled[i][j] == 'a')

{

S = 0;

}

if (posled[i][j] == 'c')

{

S = 1;

}

if (posled[i][j] == 'b')

{

S = 0;

}

break;

default:

break;

}

}

if (S == 4 || S==0)

{

label1.Text += posled[i] + '\n';

}

else

{

label2.Text += posled[i] + '\n' ;

}

}

}

private void richTextBox1\_TextChanged(object sender, EventArgs e)

{

}

private void label4\_Click(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void label7\_Click(object sender, EventArgs e)

{

}

}

}

