МИНОБРНАУКИ РОССИИ

ФГБОУ ВО Череповецкий государственный университет

Институт информационных технологий

Кафедра: МПО ЭВМ

Дисциплина: Теория автоматов и формальных языков.

ЛАБОРАТОРНАЯ РАБОТА № 3-4

Выполнил:

студент гр. 1ПИб-01-1оп-21

Трофимов Н.В.

Проверил: Ганичева Оксана Георгиевна

Череповец, 2022 г.

**Задание к лабораторной работе № 3-4**

Тема: «Регулярные множества и регулярные выражения»

Выполнить задания из учебника Карпов Ю.Г. «Теория автоматов»

1. Стр.183 задание 1. Под буквами е, ж,

Задание 2. Под буквами а, б, в, г.

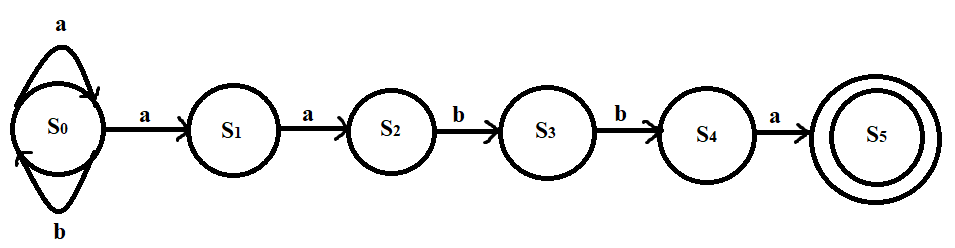
1. Стр. 185 задания 19, 21
2. Выполнить задание по вариантам.

Построить регулярное выражение, которое определяет язык, соответствующий вашему варианту (см таблицу, вариант по списку группы). Для заданного регулярного выражения построить детерминированный конечный автомат. На основе конечного автомата написать программу для распознавания строк, принадлежащих языку, определяемому регулярным выражением.

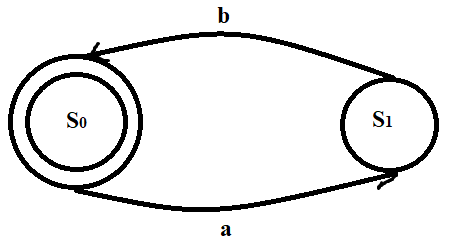
Задание 1

Построить конечный автомат с входным алфавитом V = {a,b}, распознающий:

Е) все цепочки, заканчивающиеся кодом aabba



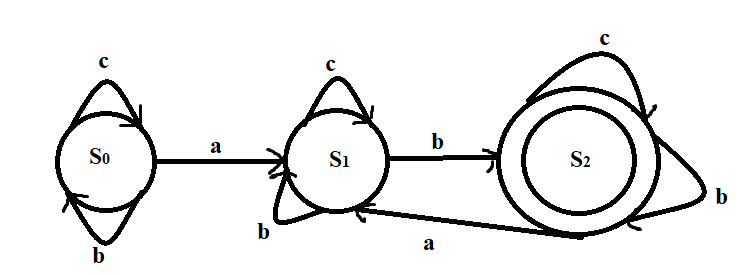
Ж) все цепочки, в которых за каждым a непосредственно следует b



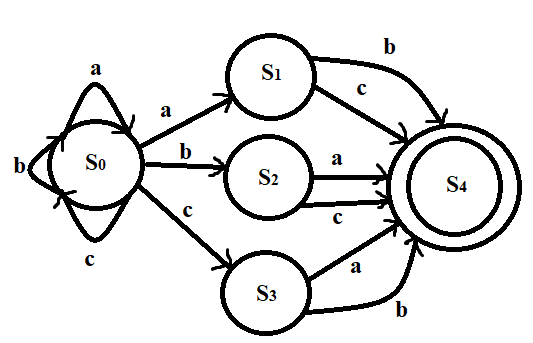
Задание 2

Построить конечный автомат с входным алфавитом V = {a,b,c} , распознающий:

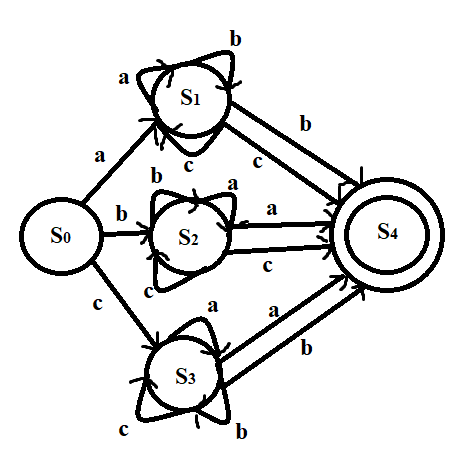
А) все цепочки, в которых за каждым a когда-нибудь в будущем следует b



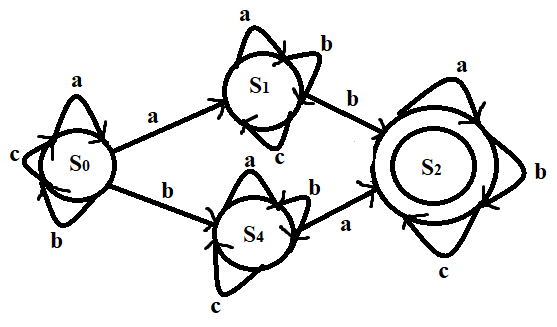
Б) все цепочки, в которых две последние буквы не совпадают



В) все цепочки, начинающиеся и заканчивающиеся различными символами



Г) все цепочки, включающие по крайней мере один символ a и один символ b



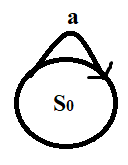
Задание 19

Построить автоматы, распознающие языки, задаваемые регулярным выражениям:

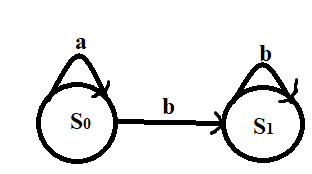
1. a\*b\*



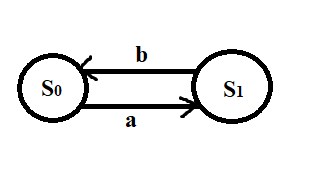
1. a\*a\*



1. a\*+b\*



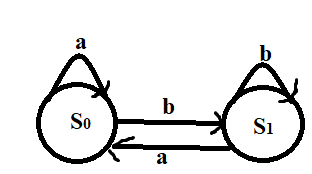
1. (a+b)\*



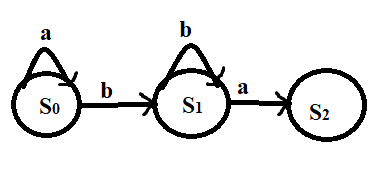
1. (a\*b\*)\*



1. (a\*+b\*)\*



1. a\*b+b\*a



Задание 21

Построить регулярное выражение, задающее множество всех таких слов над словарем {a,b,c}, в которых за символом b:

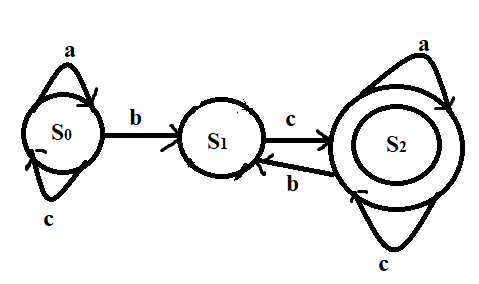
а) обязательно стоит символ с

б) не может стоять символ с

Построить конечные автоматы, распознающие соответствующие языки.

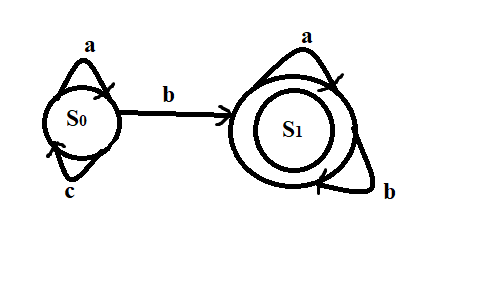
Пункт А

Регулярное выражение – (a\*c\*(bc)\*a\*c\*)\*



Пункт Б

Регулярное выражение – a\*c\*ba\*b\*



Задание 3, вариант 26

Алфавит языка – V = {a,b,c}

Описание языка - множество слов, в которых ни в одном слове нет двух и более букв a подряд.

Регулярное выражение – a, b\*c\*, (b\*c\*ab\*c\*)\*

KA = (X, S, S0, F, ∂)

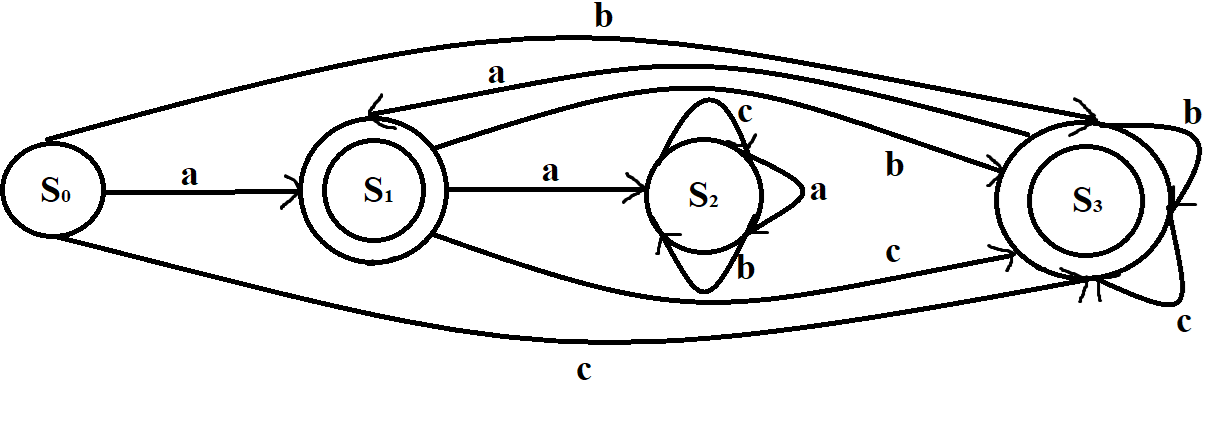
X = {a, b, c}

S – {S0, S1, S2, S3}

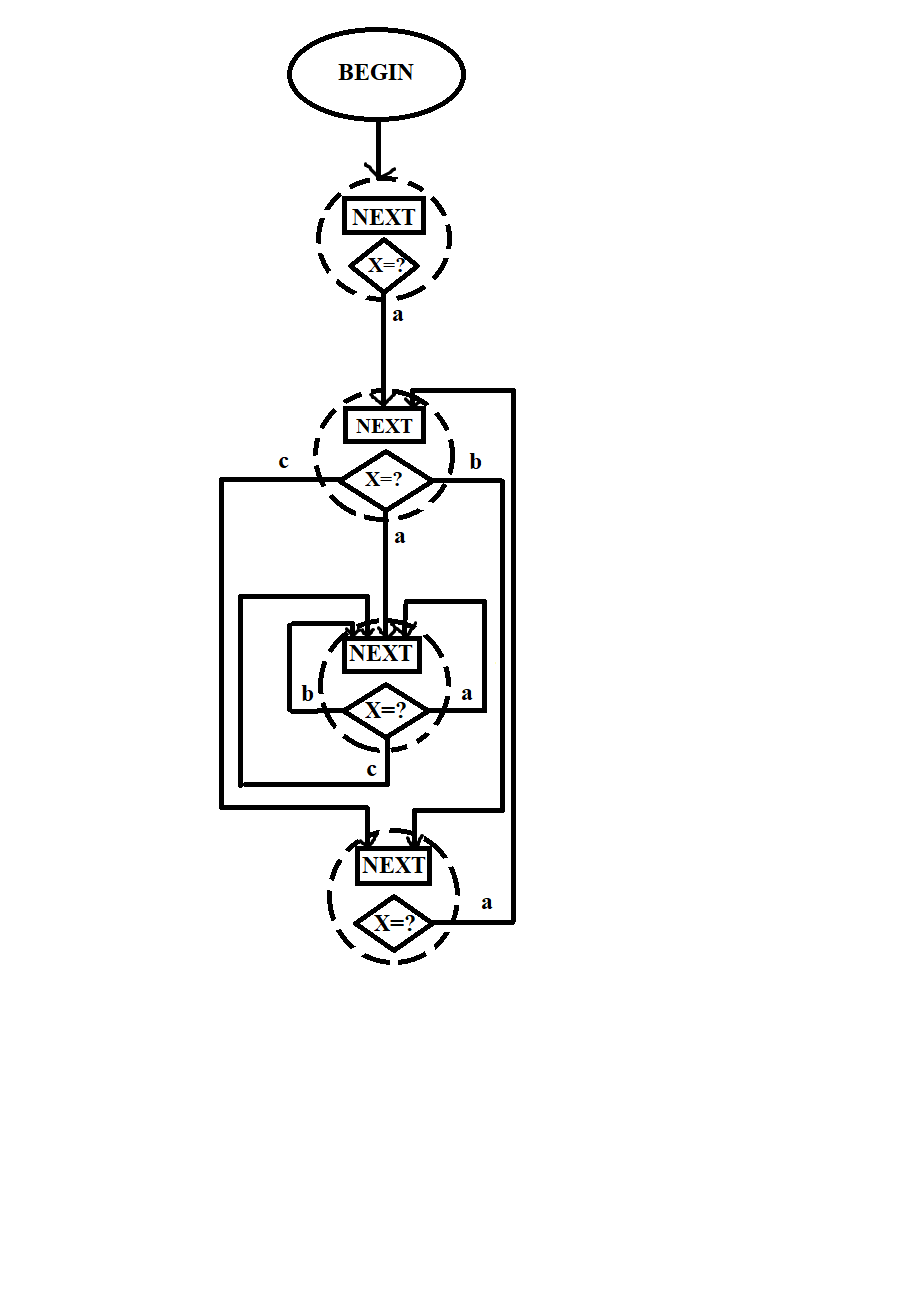
F = {S1, S3}

∂: (S0, b) -> S3, (S0, a) -> S1, (S0, c) -> S3, (S1, a) -> S2, (S1, b) -> S3, (S1,c) -> S3, (S2, a) -> S2, (S2, b) -> S2, (S2, c) -> S2, (S3, b) -> S3, (S3, c) -> S3, (S3, a) -> S1.

Граф конечного автомата

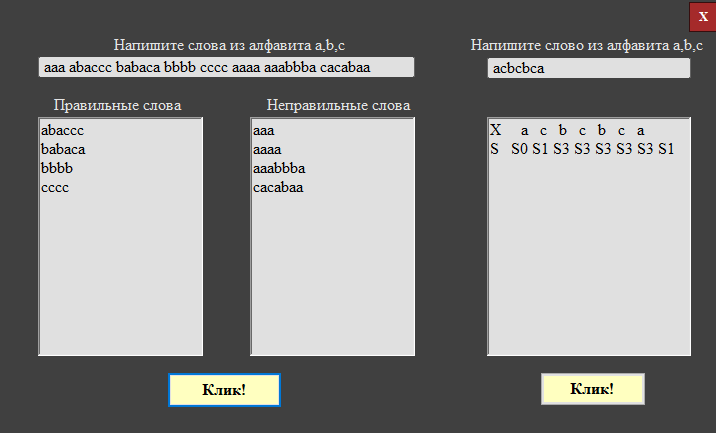


Структурная схема автомата

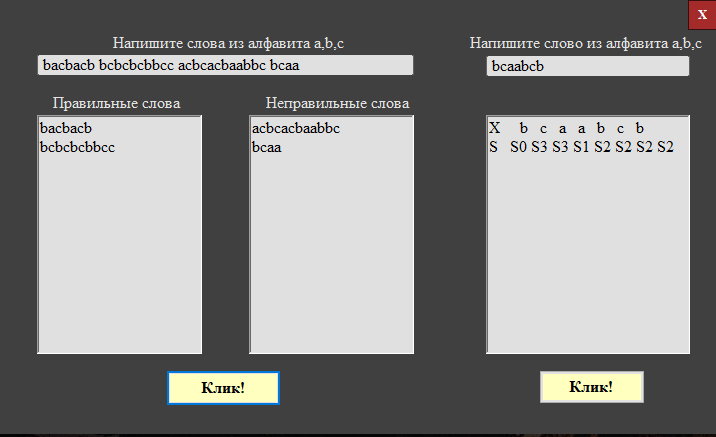


Результаты тестирования

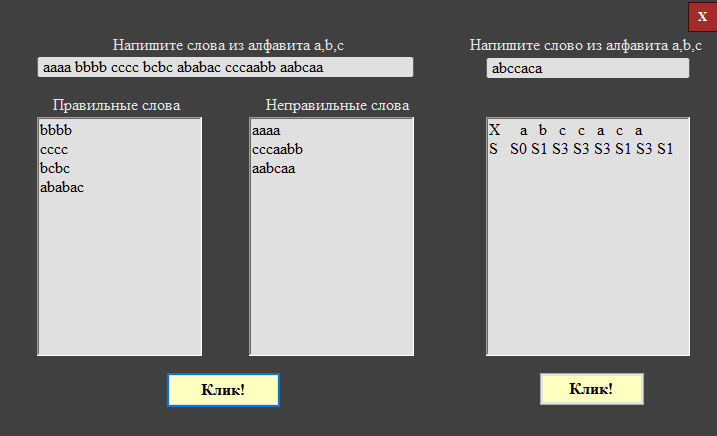
Тест 1



Тест 2



Тест 3



Код программы

#pragma endregion

private: System::Void textBox1\_TextChanged(System::Object^ sender, System::EventArgs^ e) {

textBox1->Multiline = true;

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

richTextBox1->Clear();

System::String^ str = textBox1->Text;

System::String^ s = "S0 ";

int status = 0;

int shag = 0;

for (int i = 0; i < str->Length; i++) {

switch (status) {

case 0:

switch (str[i]) {

case 'a':

status = 1;

s += "S1 ";

shag++;

break;

case 'b':

status = 3;

s += "S3 ";

shag++;

break;

case 'c':

status = 3;

s += "S3 ";

shag++;

break;

default:

break;

}

break;

case 1:

switch (str[i]) {

case 'a':

status = 2;

s += "S2 ";

shag++;

break;

case 'b':

status = 3;

s += "S3 ";

shag++;

break;

case 'c':

status = 3;

s += "S3 ";

shag++;

break;

default:

break;

}

break;

case 2:

switch (str[i]) {

case 'a':

status = 2;

s += "S2 ";

shag++;

break;

case 'b':

status = 2;

s += "S2 ";

shag++;

break;

case 'c':

status = 2;

s += "S2 ";

shag++;

break;

default:

break;

}

break;

case 3:

switch (str[i]) {

case 'a':

status = 1;

s += "S1 ";

shag++;

break;

case 'b':

status = 3;

s += "S3 ";

shag++;

break;

case 'c':

status = 3;

s += "S3 ";

shag++;

break;

default:

break;

}

break;

default:

break;

}

}

richTextBox1->Text += "X ";

for (int i = 0; i < str->Length; i++) {

richTextBox1->Text += " " + str[i] + " ";

}

richTextBox1->Text += "\nS ";

richTextBox1->Text += s;

}

private: System::Void textBox2\_TextChanged(System::Object^ sender, System::EventArgs^ e) {

textBox2->Multiline = true;

}

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e) {

richTextBox2->Clear();

richTextBox3->Clear();

System::String^ str = textBox2->Text;

array<String^>^ newstr = str->Split(' ');

for (int i = 0; i < newstr->Length; i++)

{

int status = 0;

String^ strr = newstr[i];

for (int j = 0; j < newstr[i]->Length; j++) {

switch (status) {

case 0:

switch (strr[j]) {

case 'a':

status = 1;

break;

case 'b':

status = 3;

break;

case 'c':

status = 3;

break;

}

break;

case 1:

switch (strr[j]) {

case 'a':

status = 2;

break;

case 'b':

status = 3;

break;

case 'c':

status = 3;

break;

}

break;

case 2:

switch (strr[j]) {

case 'a':

status = 2;

break;

case 'b':

status = 2;

break;

case 'c':

status = 2;

break;

}

break;

case 3:

switch (strr[j]) {

case 'a':

status = 1;

break;

case 'b':

status = 3;

break;

case 'c':

status = 3;

break;

}

break;

}

}

if (status == 1 || status == 3) {

richTextBox2->Text += newstr[i] + "\n";

}

else {

richTextBox3->Text += newstr[i] + "\n";

}

}

}

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e) {

this->Close();

}