ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

**БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ**

**ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

(НИУ «БелГУ»)

ИНСТИТУТ ИНЖЕНЕРНЫХ И ЦИФРОВЫХ ТЕХНОЛОГИЙ

КАФЕДРА МАТЕМАТИЧЕСКОГО

И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ

**Отчет по лабораторной работе 3**

**по дисциплине: «Теория автоматов»**

студента очного отделения

4 курса 12001801 группы

Капустина Виктора Сергеевича

Проверил(а):

доц. Великая Яна Геннадьевна

Белгород 2021

**Вариант 4**

**Цель работы:** изучить основные способы задания регулярных языков, алгоритмы построения, преобразования, анализа и реализации конечных распознавателей.

**Задания:**

**1**. Построить минимальный детерминированный конечный распознаватель заданного языка.

**2**. Написать программу-распознаватель.

Исходные данные: строка.

Результат: “допустить” – если строка представляет собой цепочку заданного языка, “отвергнуть” – в противном случае.

**3**. Написать программу, которая оставляет в исходном текстовом файле только цепочки заданного языка.

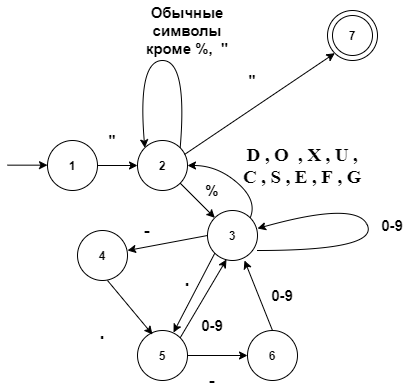
**4**. Написать программу, которая исключает из исходного текстового файла подстроки, являющиеся цепочками заданного языка.

**Индивидуальное задание: *Язык управляющих строк.***

Управляющая строка начинается и заканчивается кавычками, содер­жит обычные символы и спецификации. Спецификация начинается символом *%* и заканчивается символом преобразования. Между *%* и символом преоб­разования может находиться формат, представляющий собой последователь­ность цифр, возможно, разделённых на две части точкой, перед которой или за которой может находиться знак минус. Символами преобразования явля­ются: *D , O , X , U , C , S , E , F , G* . Символом *%* может начинаться только спецификация.

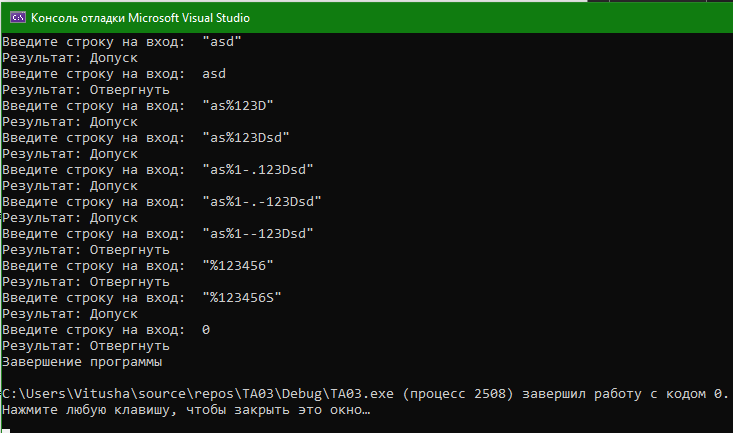
**Ход работы**

**1**. Построить минимальный детерминированный конечный распознаватель заданного языка.

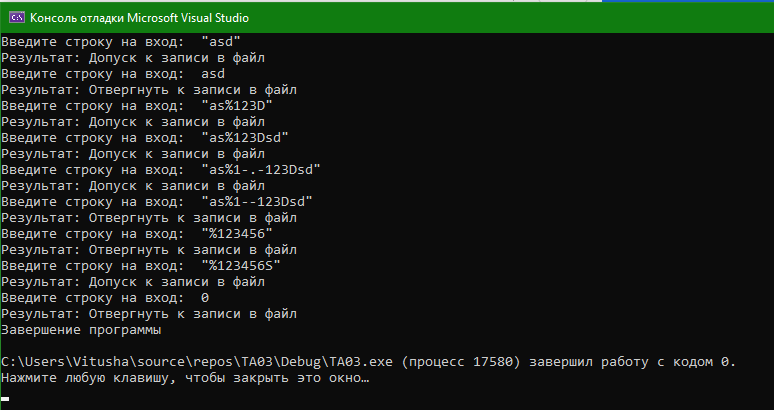
  
Рис. 1. Схема автомата

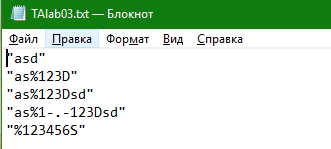
**2**. Написать программу-распознаватель.

Разрабатываемая программа должна получать на вход строку, и в ходе распознавания выдать допуск (если строка представляет собой цепочку заданного языка) или отвергнуть (в ином случае)

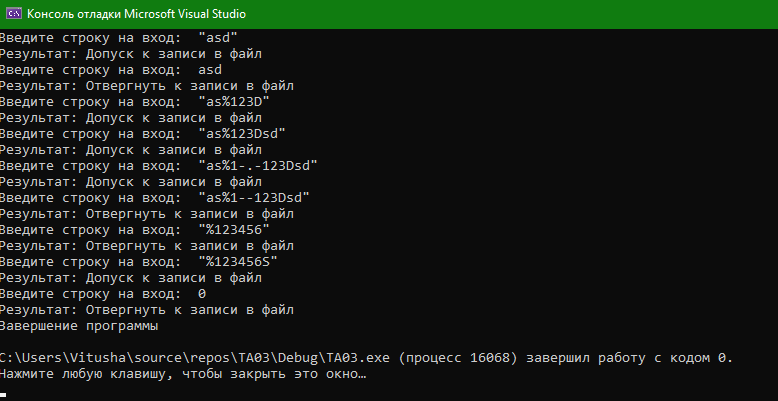
  
Рис. 2 Результат работы программы

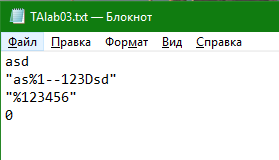
**3**. Написать программу, которая оставляет в исходном текстовом файле только цепочки заданного языка.

  
Рис. 3 Результат работы программы

  
Рис. 4 Записанные в файл данные

**4**. Написать программу, которая исключает из исходного текстового файла подстроки, являющиеся цепочками заданного языка.

  
Рис. 5 Результат работы программы

  
Рис. 6 Записанные в файл данные

**Вывод:** В ходе данной лабораторной работы мною были изучены основные способы задания регулярных язы­ков, алгоритмы построения, преобразования, ана­лиза, а также реализации конечных распознавателей.

**Листинг конечной программы**

#include <iostream>

#include <string>

#include <cstdlib>

#include <cstdio>

#include <iostream>

#include <fstream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "RU");

string str = "";

while (str != "0")

{

cout << "Введите строку на вход: ";

cin >> str;

int statement = 1;

int i = 0;

int last = str.length();

bool flag1 = true;

if (str[i] == '\"' && last >= 2) {

i = 1;

statement = 2;

while ((flag1 == true) && (i <= last) ) {

switch (statement)

{

case 2:

if (str[i] != '\"' && str[i] != '%')

{

statement = 2;

i++;

}

else if (str[i] == '%')

{

statement = 3;

i++;

}

else if (str[i] == '"')

{

statement = 7;

i++;

}

else

{

flag1 = false;

}

break;

case 3:

if (str[i] == 'D' || str[i] == 'O' || str[i] == 'X' || str[i] == 'U' || str[i] == 'C' || str[i] == 'S' || str[i] == 'E' || str[i] == 'F' || str[i] == 'G')

{

statement = 2;

i++;

}

else if (str[i] >= '0' && str[i] <= '9')

{

statement = 3;

i++;

}

else if (str[i] == '-')

{

statement = 4;

i++;

}

else if (str[i] >= '.')

{

statement = 5;

i++;

}

else {

flag1 = false; }

break;

case 4:

if (str[i] == '.') {

statement = 5;

i++;

}

else flag1 = false;

break;

case 5:

if (str[i] >= '0' && str[i] <= '9') {

statement = 3;

i++;

}

else if (str[i] == '-') {

statement = 6;

i++;

}

else {

flag1 = false; }

break;

case 6:

if (str[i] >= '0' && str[i] <= '9') {

statement = 3;

i++;

}

else {

flag1 = false; }

break;

case 7:

flag1 = false;

break;

}

}

}

else

{

statement = 0;

flag1 = false;

}

cout << "Результат: ";

if (statement != 7) {

ofstream out("C:\\Users\\Vitusha\\Desktop\\TAlab03.txt", ios::app);

if (out.is\_open())

{

out << str << endl;

}

out.close();

cout << "Отвергнуть к записи в файл" << endl;

}

else cout << "Допуск к записи в файл" << endl;

}

cout << "Завершение программы" << endl;

return 0;

}

**Конец листинга**