МИНЕСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение высшего

профессионального образования

«Санкт-Петербургский государственный университет»

**ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ПРОЦЕССОВ УПРАВЛЕНИЯ**

Курсовая работа

по дисциплине

“Теория автоматов и формальных языков”

Работу выполнил студент

Голубев Роман Андреевич

232 группа

Санкт-Петербург

2019

**Постановка задачи**

Вариант 4:

Задать порождающую грамматику языка {1n 02n 1m 02m | n, m >= 0} и построить по ней распознающий МП НКА.

Разработать программу, распознающую цепочки этого языка на основе программной реализации МП НКА.

Программа должна выводить: входную цепочку, принадлежит ли она языку, а если нет, то указать первое несоответствие входной цепочки заданному языку. Реализация осуществляется на любом языке программирования.

В отчете следует представить: грамматику, формальное задание МП НКА, код программы и результаты тестирования в виде таблицы, показывающей входную цепочку, результат проверки и первое несоответствие языку (не менее 20 вариантов тестирования, включая правильные и неправильные входные цепочки).

**Грамматика**

S-> AB

A->1A00 | ε

B->1B00 | ε

**МП НКА**

P = (Q, Σ, Г, δ, 𝑞0, 𝑍0,{𝑞4})

Q: 𝑞0,q1,q2,q3, 𝑞4

Σ: {0,1}

Г: {0,1, Z0}

F: 𝑞4

δ:

δ(q0, 1 , Z0) = {(q0, 11Z0)}

δ(q0, 1 , 1) = {(q0, 111)}

δ(q0, 0 , 1) = {(q1, ε)}

δ(q1, 0 , 1) = {(q1, ε)}

δ(q1, 1 , Z0) = {(q2, 11Z0)}

δ(q1, ε , Z0) = {(q4, Z0)}

δ(q2, 1 , 1) = {(q2, 111)}

δ(q2, 0 , 1) = {(q3, ε)}

δ(q3, 0 , 1) = {(q3, ε)}

δ(q3, ε , Z0) = {(q4, Z0)}

δ(q0, ε, Z0) = {(q4, Z0)}

**Код Программы**

#include <iostream>

#include <vector>

#include <string>

using namespace std;

class NKA

{

private:

char\* stroka; // массив, который будет хранить нашу строчку

vector<int> stack; // стэк

int curr; // на каком мы элементе

int length; // длина строчки

public:

NKA(string x)

{

stroka = new char[x.size()+1];

length = x.size()+1;

strcpy(stroka, x.c\_str());

stack.push\_back(2);

curr = 0;

}

void Get\_stroka()

{

for(int i = 0; length>i; ++i)

cout << stroka[i];

cout << endl;

}

void q4() // Состояние q4

{cout << "Данная цепочка подходит" << endl;}

void q0 () // состояние q0

{

//cout << stroka[curr] << " "<< stack.back()<< endl;

if(stack.back() > 0 && stroka[curr] == '1' && curr < length)

{

stack.push\_back(1);

stack.push\_back(1);

curr++;

return q0();

}

else if(stack.back() == 1 && stroka[curr] == '0' && curr < length)

{

stack.pop\_back();

curr++;

return q1();

}

else {

cout << "Ошибка на элементе:" << curr+1 << endl;

}}

void q1 () // состояние q1

{

if(stack.back() == 1 && stroka[curr] == '0' && curr < length)

{

stack.pop\_back();

curr++;

return q1();

}

else if(stack.back() == 2 && stroka[curr] == '1' && curr < length)

{

stack.push\_back(1);

stack.push\_back(1);

curr++;

return q2(); }

else if (stack.back() == 2 && curr >=length-1)

{ return q4();}

else {cout << "Ошибка на элементе:" << curr+1 << endl; }

}

void q2() // состояние q2

{

if(stack.back() == 1 && stroka[curr] == '1' && curr < length)

{

stack.push\_back(1);

stack.push\_back(1);

curr++;

return q2();

}

else if(stack.back() == 1 && stroka[curr] == '0' && curr < length)

{

stack.pop\_back();

curr++;

return q3();

}

else {

cout << "Ошибка на элементе:" << curr+1 << endl;

}}

void q3 () // состояние q3

{

if(stack.back() == 1 && stroka[curr] == '0' && curr < length)

{

stack.pop\_back();

curr++;

return q3();

}

else if (stack.back() == 2 && curr >=length-1)

{ return q4();}

else {

cout << "Ошибка на элементе:" << curr+1 << endl;

}

}};

int main() {

string chain;

cin >> chain;

NKA test = NKA(chain);

// test.Get\_stroka();

test.q0();

return 0;

}

**Описание работы программы**

Вам предлагается ввести цепочку символов. Когда вы введете цепочку в консоль и она будет подходить к нашему языку, то в консоли выведется надпись “Данная цепочка подходит”. Если же введенная цепочка символов не подходит, то в консоли появится надпись “Ошибка на элементе:(порядковый номер символа, на котором программа остановилась)”.

В программе реализован метод на каждое состояние МП НКА, который осуществляет переход на следующее состояние.

**Тесты**

|  |  |
| --- | --- |
| **110000** | **Данная цепочка подходит** |
| **sf110w** | **Ошибка на элементе: 1** |
| **111000000110000** | **Данная цепочка подходит** |
| **110** | **Ошибка на элементе: 4** |
| **111100001** | **Ошибка на элементе: 10** |
| **110010s** | **Ошибка на элементе: 7** |
| **110011001100** | **Ошибка на элементе: 9** |
| **100110000** | **Данная цепочка подходит** |
| **111100000000100** | **Данная цепочка подходит** |
| **21010** | **Ошибка на элементе: 1** |
| **110000110000** | **Данная цепочка подходит** |
| **000011000011** | **Ошибка на элементе: 1** |
| **110001** | **Ошибка на элементе: 5** |
| **Ясдамвсеэкзамены** | **Ошибка на элементе: 1** |
| **100100** | **Данная цепочка подходит** |
| **220030** | **Ошибка на элементе: 1** |
| **11000020** | **Ошибка на элементе: 7** |
| **100100** | **Данная цепочка подходит** |
| **11110000000000000** | **Ошибка на элементе: 13** |
| **100** | **Данная цепочка подходит** |