МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное общеобразовательное учреждение высшего образования

«Ижевский государственный технический университет имени М.Т. Калашникова»

Кафедра «Программное обеспечение»

ОТЧЕТ

по заданию 6

по дисциплине «Теория автоматов и формальных языков»

Выполнил: студент группы Б04-191-3 Р.А. Гумметов

Принял: д.т.н., профессор М.А. Сенилов

Ижевск 2020

ОПИСАНИЕ ПРОГРАММЫ, РЕАЛИЗУЮЩЕЙ РАСПОЗНАЮЩИЙ АВТОМАТ

8.1 Вводная часть

Для проверки правильности построенного конечного распознавателя, написана программа. Программа реализует работу распознающего автомата и производит распознавание вводимых с клавиатуры цепочек. Программа написана на языкеC#.

8.2 Функциональное назначение

Программа имитирует работу конечного автомата. Программа применяется для распознавания входных цепочек символов праволинейной грамматики.

Для функционирования программы необходима любая ЭВМ, имеющая

транслятор языка C#.

Для работы программы требуются устройство ввода(клавиатура) и устройство вывода(дисплей)

8.3 Описание информации

В качестве входной информации выступают строки, вводимые с клавиатуры, состоящие из символов исходной грамматики и являющиеся строкой для распознавания. Информация о допустимости цепочек выводится на дисплей. Входные данные имеют формат: хАхВхС , где А, В, С - числа от 1 до 7. Если цепочка допустима, то на экране появится сообщение «Цепочка допустима». Если цепочка недопустима, то на экране появится сообщение «Цепочка не допустима».

8.4 Описание логики

Логику написанной программы иллюстрирует схема программы, представленная на рис. 1,рис. 2 и рис. 3

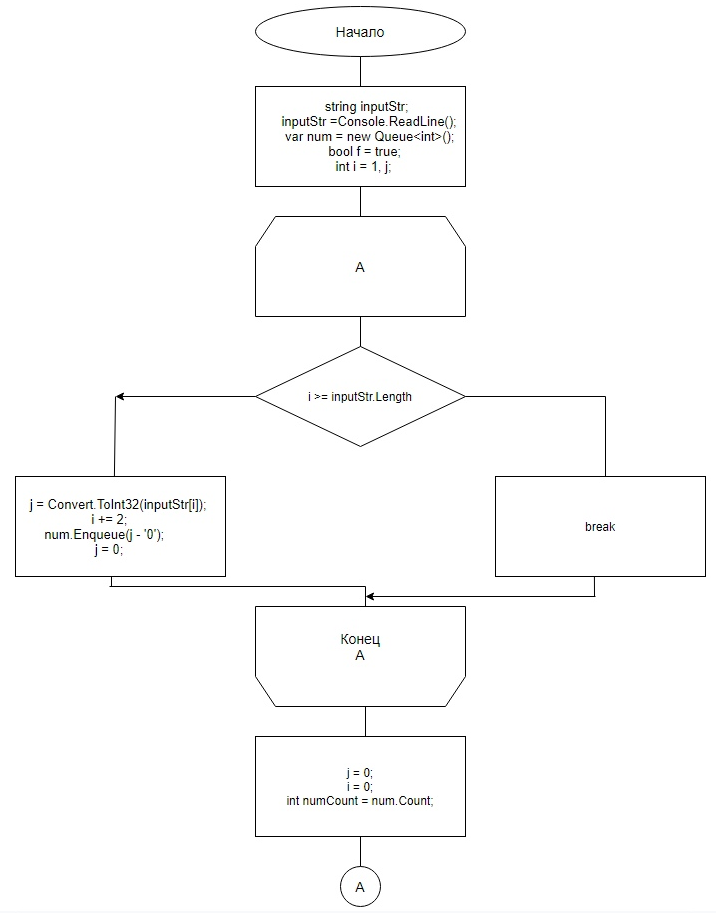


Рис. 1

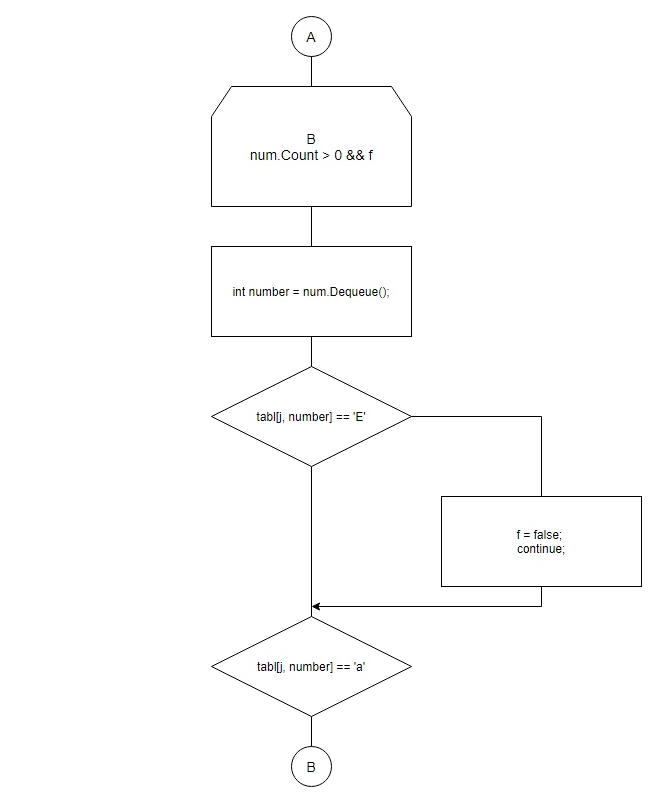


Рис. 2

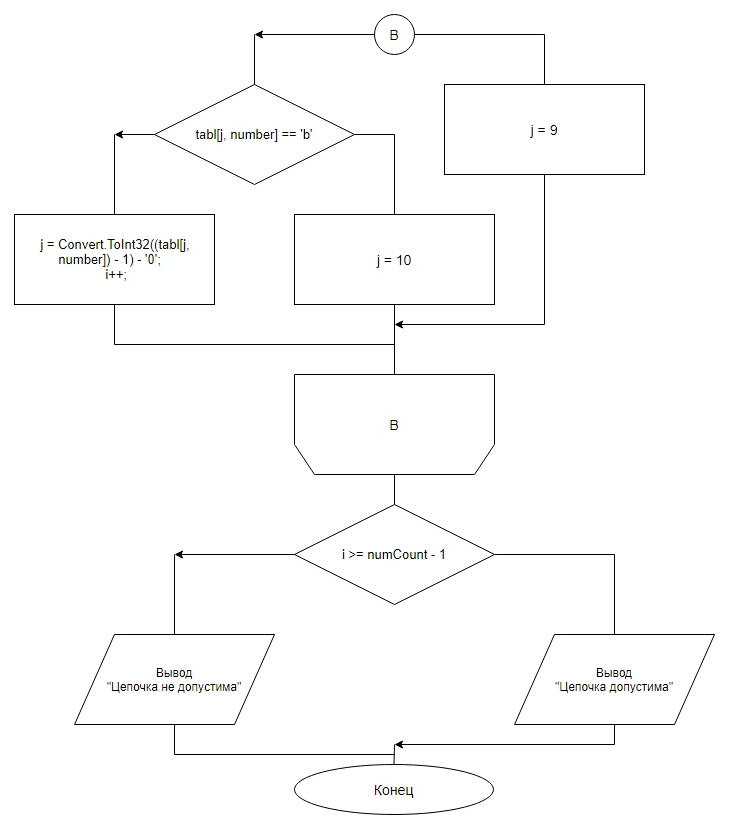


Рис. 3

9. ОПИСАНИЕ КОНТРОЛЬНОГО ПРИМЕРА

9.1 Назначение

Контрольный пример предназначен для тестирования программы, реализующей конечный автомат.

9.2 Исходные данные

Исходные данные - цепочка символов. В нее входят символы из множества: {x1,x2,x3,x4,x5,x6,x7}.

Построим цепочки символов, для контрольного примера, исходя из право-линейной грамматики.

Для проверки правильности работы автомата нужно проверить его с помощью допустимых цепочек.

Что бы получить допустимую цепочку символов необходимо взять одно из правил, в левой части которого стоит начальный символ S.

Выписать все терминальные символы из этого правила и если в конце стоит нетерминал, то перейти к одному из правил, в левой части которого стоит этот нетерминал.

Выписать терминальные символы из этого правила и если в конце стоит нетерминал, то перейти к новому правилу и т.д., пока мы не дойдем до правила, правая часть которого кончается терминалом.

9.3 Результаты расчета

Итак, получаем допускающие цепочки:

1) S→x4x7x3x2

Отсюда получаем цепочку: x4x7x3x2

2) S→ x5x2x0

Отсюда получаем цепочку: x5x2x0

3) S→x7x4x5

Отсюда получаем цепочку: x7x4x5

Для полной проверки автомата получим несколько недопустимых цепочек. Их можно получить, если выписывать терминалы, не доходя до терминала, который стоит последним в правиле.

Или же если записать терминал, которого нет в правой части ни одного из правил, в левой части которых стоит необходимый нетерминал.

Недопустимые цепочки:

4) x5x2x1

5) x4x7x3x0

6) x4x7x5

Проверим полученные цепочки на минимальном автомате, построив соответствующие последовательности переходов автомата:

1) x4x7x3x2

1-2-3-4-5

5 - допускающее состояние, значит, цепочка допускается

2) x5x2x0

1-4-5-1

1- допускающее состояние, значит, цепочка допускается

3) x7x4x5

1-6-7-8

8 - допускающее состояние, значит, цепочка допускается

4) x5x2x1

1-4-5-Er

Er - отвергающее состояние, значит, цепочка отвергается

5) x4x7x3x0

1-2-3-4-Er

Er - отвергающее состояние, значит, цепочка отвергается

6) x4x7x5

1-2-3-Er

Er - отвергающее состояние, значит, цепочка отвергается

9.4 Результаты испытания программы

Результаты испытания программы представлены на рис.4-9.

1.png

Рис. 4. Контрольный пример 1

2.png

Рис.5. Контрольный пример 2

3.png

Рис.6. Контрольный пример 3

4.png

Рис.7. Контрольный пример 4

5.png

Рис.8. Контрольный пример 5

6.png

Рис.9. Контрольный пример 6

Сравним результаты работы программы с результатами работы автомата:

|  |  |  |  |
| --- | --- | --- | --- |
| № | Входная цепочка | Результат работы программы | Результат работы автомата |
| 1 | x4x7x3x2 | Цепочка допускается | Цепочка допускается |
| 2 | x5x2x0 | Цепочка допускается | Цепочка допускается |
| 3 | x7x4x5 | Цепочка допускается | Цепочка допускается |
| 4 | x5x2x1 | Цепочка не допускается | Цепочка не допускается |
| 5 | x4x7x3x0 | Цепочка не допускается | Цепочка не допускается |
| 6 | x4x7x5 | Цепочка не допускается | Цепочка не допускается |

Приложение

Текст программы на языке C#

using System;

using System.Collections.Generic;

class Program

{

static void Main()

{

var tabl = new char[,]

{

{'E', 'E', 'E', 'E', '2', '4', 'E', '6' },

{'E', 'E', 'E', 'E', 'E', 'E', 'E', '3' },

{'E', 'E', 'E', '4', 'E', 'E', '4', 'E' },

{'E', 'E', '5', 'E', 'E', 'b', 'E', 'E' },

{'1', 'E', 'E', 'E', 'E', 'E', 'E', 'b' },

{'E', 'a', 'E', 'E', '7', '7', 'E', 'E' },

{'E','E','E','E','E', '8', 'E','E' },

{'E', '9', 'E','E','E','E','E','E' },

{'E','E','E', 'b', 'E','E','E','E' },

{'E','E','E','E','E','E','E', '9' },

{'E','E','E','E','E','E','E','E' },

{'E','E','E','E','E','E','E','E' }

};

string inputStr;

inputStr = Console.ReadLine();

var num = new Queue<int>();

bool f = true;

int i = 1, j;

while (true)

{

if (i >= inputStr.Length) break;

j = Convert.ToInt32(inputStr[i]);

i += 2;

num.Enqueue(j - '0');

j = 0;

}

j = 0;

i = 0;

int numCount = num.Count;

while (num.Count > 0 && f)

{

int number = num.Dequeue();

if (tabl[j, number] == 'E') { f = false; continue; }

if (tabl[j, number] == 'a') j = 9;

else if (tabl[j, number] == 'b') j = 10;

else j = Convert.ToInt32((tabl[j, number]) - 1) - '0';

i++;

}

if (i >= numCount - 1) Console.WriteLine("Цепочка допустима");

else Console.WriteLine("Цепочка не допустима");

}

}