МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное общеобразовательное учреждение высшего образования

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

Кафедра «Программное обеспечение»

ОТЧЕТ

по заданию 7

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

1. ОПИСАНИЕ КОНТРОЛЬНОГО ПРИМЕРА

1.1 Назначение

Контрольный пример предназначен для тестирования программы, реализующей конечный автомат.

1.2 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.

1.3 Результаты расчета

Итак, получаем допускающие цепочки:

1) S→x4x7x3A→x2

Отсюда получаем цепочку: x4x7x3x2

2) S→ x5B → x2E → x0

Отсюда получаем цепочку: x5x2x0

3) S→x7x4A → x5

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

1.4 Результаты испытания программы

Результаты испытания программы представлены на рис.4-9.

1.png

Рис. 1. Контрольный пример 1

2.png

Рис.2. Контрольный пример 2

3.png

Рис.3. Контрольный пример 3

4.png

Рис.4. Контрольный пример 4

5.png

Рис.5. Контрольный пример 5

6.png

Рис.6. 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 |

Результаты испытания программы совпали с результатами работы автомата, что говорит о правильности построения минимального автомата и реализации программы.

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");

}

}