САНКТ-ПЕТЕРБУРГСКИЙ НАЦИАОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННОЙ ТЕХНОЛОГИИ, МЕХАНИКИ И ОПТИКИ

КАФЕДРА ИНФОРМАТИКИ И ПРИКЛАДНОЙ МАТЕМАТИКИ

ЛАБОРАТОРНАЯ РАБОТА №1

«КОНЕЧНЫЕ АВТОМАТЫ

И РЕГУЛЯРНЫЕ ГРАММАТИКИ»

Выполнил

Кудряшов А.А.

Группа 2121

Проверил:

Лаздин Артур Вячеславович

Санкт-Петербург

2013 год

Вариант № 6

Задание:

Для регулярного выражения (выбрать из списка, согласно номеру в списке группы):

1. построить диаграмму переходов недетерминированного конечного автомата;
2. по полученному НКА построить детерминированный конечный автомат;
3. написать программу, реализующую распознаватель предложений языка, порождаемых регулярным выражением.

Продемонстрировать работу распознавателя на различных примерах.

Заданное регулярное выражение:

1. b(ab)\*c?

Диаграмма переходов НКА:

b

ε

a

b

ε

ε

ε

ε

ε

ε

c

Таблица соответствия состояний НКА и состояний эквивалентного ему ДКА:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | State | a | b | c |
| 1 | 1 | - | 2,3,6,7,9 | - |
| **2** | 2,3,6,7,9 | 3,4,6,7,9 | 2,3,6,7,9 | 3,6,7,8,9 |
| 3 | 3,4,6,7,9 | 3,4,6,7,9 | 2,3,5,6,7,9 | 3,6,7,8,9 |
| **4** | 3,6,7,8,9 | 3,4,6,7,9 | 2,3,6,7,9 | 3,6,7,8,9 |
| **5** | 2,3,5,6,7,9 | 3,4,6,7,9 | 2,3,5,6,7,9 | 3,6,7,8,9 |

Диаграмма переходов ДКА:

b

с

a

b

Текст программы:

class ResolverSentence

{

//структура, определяющая состояние

struct state

{

public state(bool final\_state, int name)

{

this.final\_state = final\_state;

this.name = name;

}

public bool Final\_state

{

get

{

return final\_state;

}

}

public int Name

{

get

{

return name;

}

}

private bool final\_state;

private int name;

}

//состояния для нашего ДКА

private state One = new state(false, 1);

private state Two = new state(true, 2);

private state Three = new state(false, 3);

private state Four = new state(true, 4);

private state Error = new state(false, -1);

//метод, реализующий распознаватель предложений

public void Check()

{

state CS = One;

StreamReader read = new StreamReader(@"C:\Users\Elina\Desktop\Line.txt");

int c = read.Read();

do

{

switch (CS.Name)

{

case 1:

if (c == 'b')

{

c = read.Read();

CS = Two;

}

else

CS = Error;

break;

case 2:

if (c == 'a')

{

c = read.Read();

CS = Three;

}

else if (c == 'c')

{

c = read.Read();

CS = Four;

}

else

CS = Error;

break;

case 3:

if (c == 'b')

{

c = read.Read();

CS = Two;

}

else

CS = Error;

break;

case 4:

CS = Error;

break;

}

}

while ((!CS.Equals(Error)) && (!c.Equals(-1)));

read.Close();

if (CS.Final\_state) Console.WriteLine("Принадлежит");

else Console.WriteLine("Не принадлежит");

}

}

Примеры входных цепочек и результаты их обработки программой-распознавателем:

|  |  |  |
| --- | --- | --- |
| **входная цепочка** | **результат программы** | **регулярное выражение** |
| b | принадлежит | принадлежит |
| bab | принадлежит | принадлежит |
| baba | не принадлежит | не принадлежит |
| babab | принадлежит | принадлежит |
| babc | принадлежит | принадлежит |
| babcc | не принадлежит | не принадлежит |
| bababababababababgc | не принадлежит | не принадлежит |