



INSTITUTO POLITÉCNICO NACIONAL

Escuela Superior de Cómputo

Ing. en sistemas computacionales



Nombre:

Josue Macias Castillo

Profesora:

Luz María Sánchez García

Grupo:

2CM1

Boleta:

2015301058

Materia:

Teoría Computacional

Fecha de entrega:

7 de abril de 2017

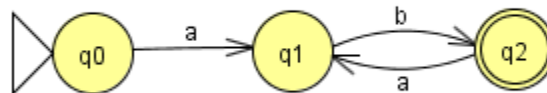
Introducción:

El propósito de este documento es reportar lo realizado en la práctica número 3 en la cual se debe implementar un autómata finito determinista que acepte una expresión regular.

Planteamiento del problema:

Se pide implementar una solución en el lenguaje de programación a nuestro gusto que tenga en la entrada cadenas para que valide con una expresión regular y muestre las transiciones por las que pasa en el autómata finito determinista.

Diseño de la solución:



Implementación de la solución:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Text.RegularExpressions;

namespace Practica_3
{
    class Program
    {
        static void Main(string[] args)
        {
            while (true)
            {
                //Declaracion de la variable que contiene la expresion
                regular string pattern = "^(ab)+$";
                string cadena;
                Console.WriteLine("La expresion regular es: (ab)+");
                Console.WriteLine("Ingresa una cadena: ");
                cadena = Console.ReadLine();
                //Compara la cadena ingresada con la expresion regular
                bool resultado = Regex.IsMatch(cadena, pattern);
                //Si el resultado de la comparacion es true
                //toma la cadena como valida
                if (resultado == true)
                {
                    Console.WriteLine("Cadena valida");
                }
            }
        }
    }
}
```

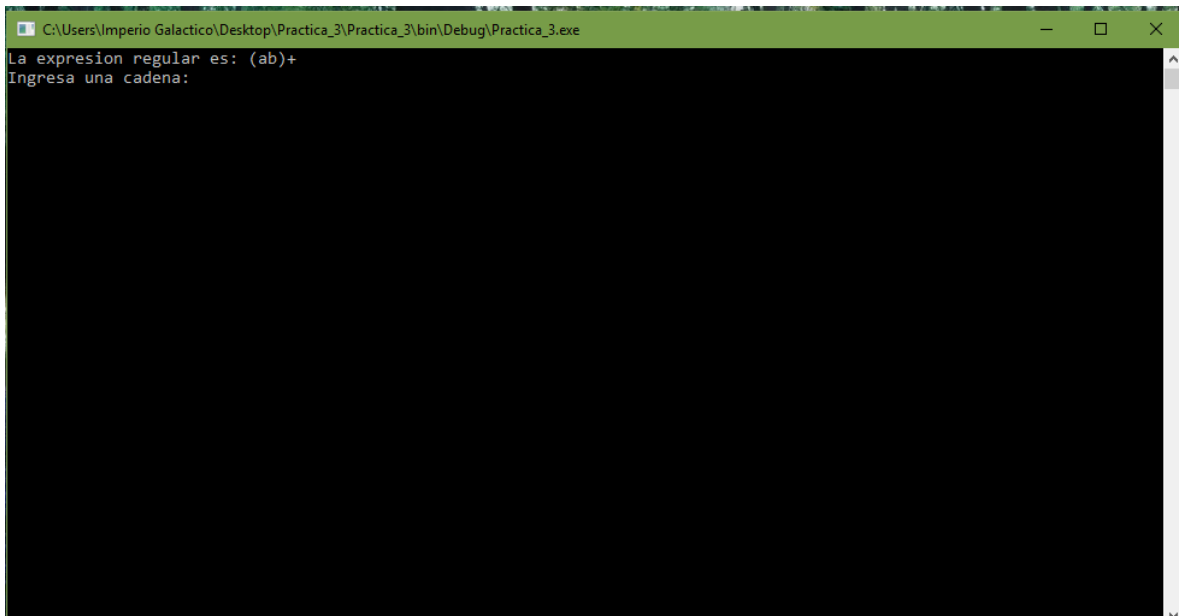
```

        //Compara el primer caracter de la cadena con "a"
        if (cadena.Substring(0,1) == "a")
        {
            //Imprime las transiciones del automata
            Console.WriteLine("q0 ---> q1 ---> a");
            for (int i = 1; i < cadena.Length; i++)
            {
                if (cadena.Substring(i,1) == "b")
                    Console.WriteLine("q1 ---> q2 ---> b");
                else if (cadena.Substring(i,1) == "a")
                    Console.WriteLine("q2 ---> q1 ---> a");
            }
        }
        //Si es false la cadena es invalida
        else
            Console.WriteLine("Cadena Invalida");
    }
    Console.ReadKey();
}
}
}
}

```

Funcionamiento:

Al iniciar el programa se muestra la siguiente información en la consola.



Nos pide ingresar una cadena que sea válida con la expresión regular que se muestra, por ejemplo, tratemos con una cadena simple: ab

```
C:\Users\Imperio Galactico\Desktop\Practica_3\Practica_3\bin\Debug\Practica_3.exe
La expresion regular es: (ab)+
Ingresa una cadena:
ab
Cadena valida
q0 ---> q1 ---> a
q1 ---> q2 ---> b
La expresion regular es: (ab)+
Ingresa una cadena:
```

Se observa que es una cadena valida con la expresión regular y nos muestra las transiciones por las que pasa en el autómata.

Ahora probemos con la cadena abababa

```
C:\Users\Imperio Galactico\Desktop\Practica_3\Practica_3\bin\Debug\Practica_3.exe
La expresion regular es: (ab)+
Ingresa una cadena:
abababa
Cadena Invalida
La expresion regular es: (ab)+
Ingresa una cadena:
```

Como se observa la cadena no corresponde con la expresión regular por esa misma razón no muestra sus transiciones.

Ahora probemos con otra cadena, por ejemplo: abababababababab

```
C:\Users\Imperio Galactico\Desktop\Practica_3\Practica_3\bin\Debug\Practica_3.exe
La expresion regular es: (ab)+
Ingresa una cadena:
abababababababab
Cadena valida
q0 --> q1 --> a
q1 --> q2 --> b
q2 --> q1 --> a
q1 --> q2 --> b
q2 --> q1 --> a
q1 --> q2 --> b
q2 --> q1 --> a
q1 --> q2 --> b
q2 --> q1 --> a
q1 --> q2 --> b
q2 --> q1 --> a
q1 --> q2 --> b
q2 --> q1 --> a
q1 --> q2 --> b
q2 --> q1 --> a
q1 --> q2 --> b
La expresion regular es: (ab)+
Ingresa una cadena:

```

De igual forma es una cadena válida y muestra sus transiciones

Por ultimo probemos con una cadena que contenga caracteres ajenos a la expresión regular, por ejemplo: abc117

```
C:\Users\Imperio Galactico\Desktop\Practica_3\Practica_3\bin\Debug\Practica_3.exe
La expresion regular es: (ab)+
Ingresa una cadena:
abc117
Cadena Invalida
La expresion regular es: (ab)+
Ingresa una cadena:

```

Como se esperaba nos dice que la cadena es invalida.

Conclusión:

En esta práctica aprendí a como validar y mostrar las transiciones con ayuda de un autómata finito determinista todo esto lo logre con ayuda del lenguaje de programación C# ya que cuenta con librerías de expresiones regulares al igual que java, además de que me da más experiencia con este lenguaje y aprendo a implementar los autómatas.

Bibliografía:

Bibliografía

Jeff Ferguson, B. P. (2003). *La biblia de C#*. Madrid: Anaya Multimedia (Grupo Anaya. S.A.).

John E. Hopcroft, R. M. (2007). *Teoria de autómatas, lenguajes y computación*. Madrid: PEARSON EDUCACION S.A.