MANUAL TÉCNICO

ANALIZADOR LÉXICO

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERIA

LENGUAJES FORMALES Y DE PROGRAMACIÓN

Jessica E. Botón P. | 201800535 | 18/09/2019

INTRODUCCIÓN

El programa “Grafico de Localizaciones” ha sido creado utilizando Visual como IDE y C# como lenguaje de programación. Se ha utilizado la Programación Orientada a Objetos y la temática de países saturados.

Se trata de un programa con interfaz gráfica en el que se le ofrecen al usuario una barra menú con distintas opciones a escoger en base a lo que él desea realizar. Siendo la principal, la carga de texto para las pestañas del programa, para posteriormente realizar el análisis léxico.

Esta es una práctica implementada en la Facultad de ingeniería, por motivos de aprendizaje, para retar el estudiante de Ciencias y Sistemas, y de esa manera, evaluar su lógica y aprendizaje.

CONTENIDO TÉCNICO

REQUISITOS PARA SU EJECUCIÓN:

- Si se desea correr el programa desde el IDE Visual Studio, deberá descargar e instalar el programa en el equipo deseado (versión 2013 y posteriores). Con una memoria RAM optima de 1GB. Espacio de disco duro óptimo de 10GB. Es compatible con los sistemas operativos de Microsoft Windows, MS-DOS.
- Si se desea ejecutar el programa desde su ejecutable. Únicamente deberá ingresar al lugar donde tenga el ejecutable .exe y Posteriormente de doble 'click'.

DICCIONARIOS DE METODOS UTILIZADOS:

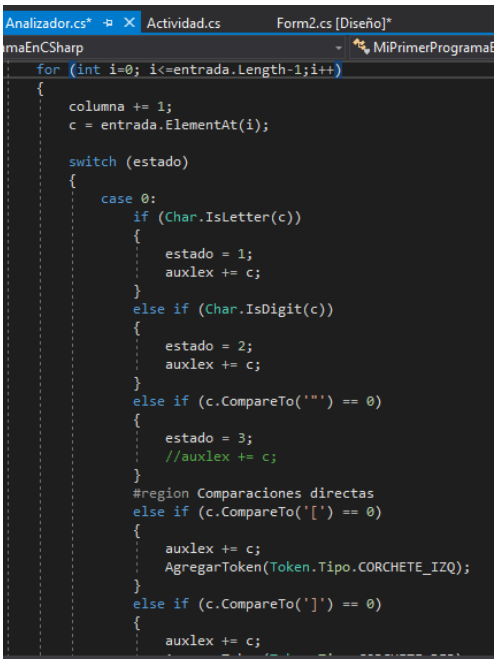
Clase – Token

- Public enum Tipo{}: En esta clase de tipo enum, definimos los tipos de token's que pueden existir.
- Token(Tipo tipoDelToken, String val, int f, int c): Este constructor se utiliza para definir los valores que tomarán las variables.
- GetTipo(): Este método devuelve el tipo de carácter que es el lexema indicado;

Clase – Error

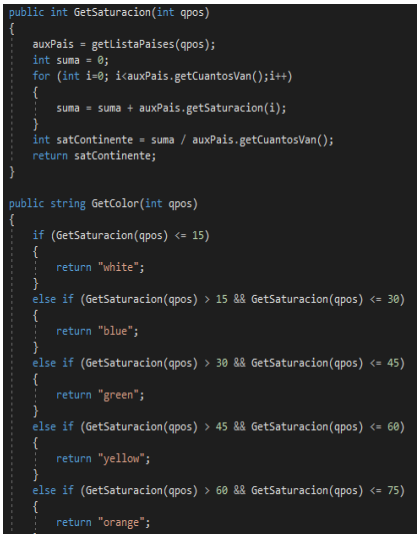
- Public enum Tipo{}: En esta clase de tipo enum, definimos los tipos de token's que pueden existir.
- Error(Tipo tipoDelError, String val, int f, int c): Este constructor se utiliza para definir los valores que tomarán las variables.
- GetTipo(): Este método devuelve el tipo de carácter que es el lexema indicado;

Clase – Analizador



- ListaTokens(String entrada): Este método es el más importante del analizador, pues es donde se aplica el Autómata Finito Determinista. Contiene un case, con los distintos estados del AFD. Por medio de un for, lee carácter por carácter y nos dirige a los estados necesarios, o realiza las aceptaciones automáticas.
  En caso de hallar un espacio en blanco o una tabulación, no realiza nada. Pero, al hallar un salto de línea, aumenta el contador de las filas y se reinicia él contador de la columna.
  Nos devuelve en listado de Tokens generados.
- AgregarError(Token.Tipo tipo): En este método, se agrega él error ocurrido, se regresa al estado qo y se limpiar el lexema.
- AgregarToken(Token.Tipo tipo): En este método, se agrega él token aceptado, se regresa al estado qo y se limpiar el lexema.

Clase – Continente



- Continente (int cantidad): Este constructor es utilizado para instanciar los arreglos que representan nuestros atributos. Entre ellos, la lista de países que cada continente tendrá.
- setContinente (string nombre): Este método es utilizado para agregar un nuevo continente a la lista
- GetSaturacion(int qpos): obtiene la lista de países del continente que corresponde a la posición recibida como parámetro y realiza la suma de todas las saturaciones para posteriormente dividirlo entre el número de países de la lista.
- String GetColor(int qpos): este método nos devuelve el color corresponde al país, según su nivel de saturación.
- `int posPaisMenosSaturado(int qpos):` recorre la lista de países correspondientes y obtiene la posición del país con menor saturación.

Clase – País

- Pais (int cantidad): Este constructor es utilizado para instanciar los arreglos que representan nuestros atributos: nombre, población, saturación y bandera.
- setPais (string nombre): Este método es utilizado para agregar un nuevo país a la lista.
- String GetColor(int qpos): este método nos devuelve el color corresponde al país, según su nivel de saturación.

Clase – Graficador

- `graficar(Continente lista, String nombreGrafo):` Este método es utilizado para crear las rutas donde se generará el archivo.dot y la imagen.png.
  Es aquí donde se crea la sintaxis que generará la imagen. Esto recorriendo la lista de continentes.
- `generarDot(String rdot, String rpng)`: Este método es utilizado para agregar el código generado en el método anterior al archivo.dot

Clase – Form 1

- button1_Click (`object sender, EventArgs e`): contiene acciones y métodos que se ejecutan siempre, como: limpiar los componentes del formulario, realizar el análisis léxico y limpiar el código en html. También, dependiendo del resultado del análisis lexico, ejecuta acciones.

- htmlTokens (LinkedList<token> tokens): Este método es utilizado para ingresar el código html generado, en base a los tokens y los errores obtenidos.

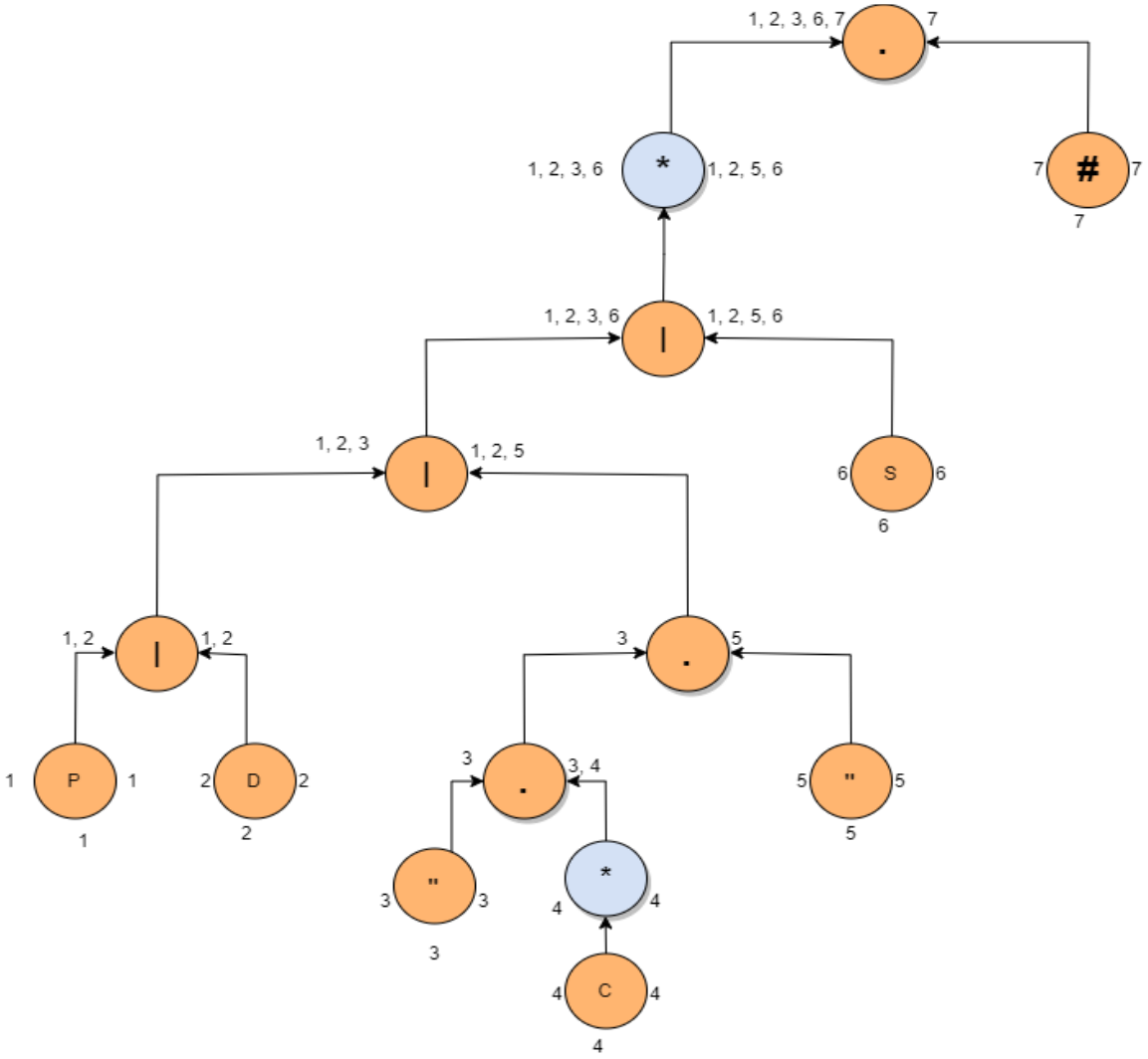
- htmlErrores (LikedList<error> errores): Este método es utilizado para ingresar el código html generado, en base a los tokens y los errores obtenidos.

AUTOMATA FINITO DETERMINISTA

ER: (P | D | “. C* .” | S)*

- L = {a-zA-Z}
- P = {continente, país, nombre, población, bandera, saturación, grafica}
- S = { {, }, ; , : , %, , \t, \n }
- D = {0-9}
- C = {x\x E A cualquier carácter existente}

MÉTODO DEL ÁRBOL



NOTAS:

- El color anaranjado pertenece a los nodos, cuya anulabilidad es Falsa (No anulables)
- El color celeste pertenece a los nodos, cuya anulabilidad es Verdadera (Anulables)

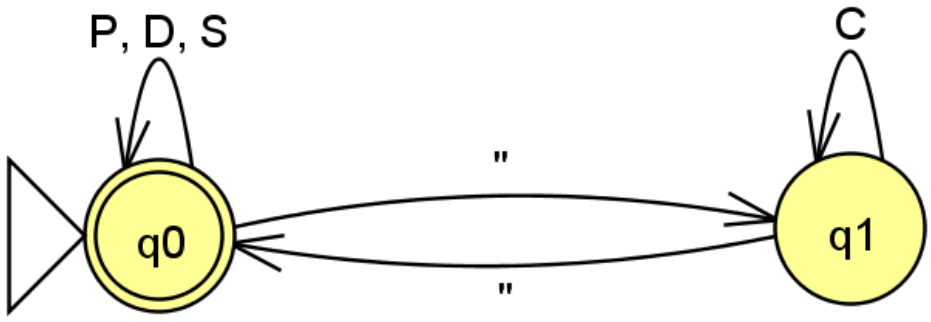
Tabla de siguientes:

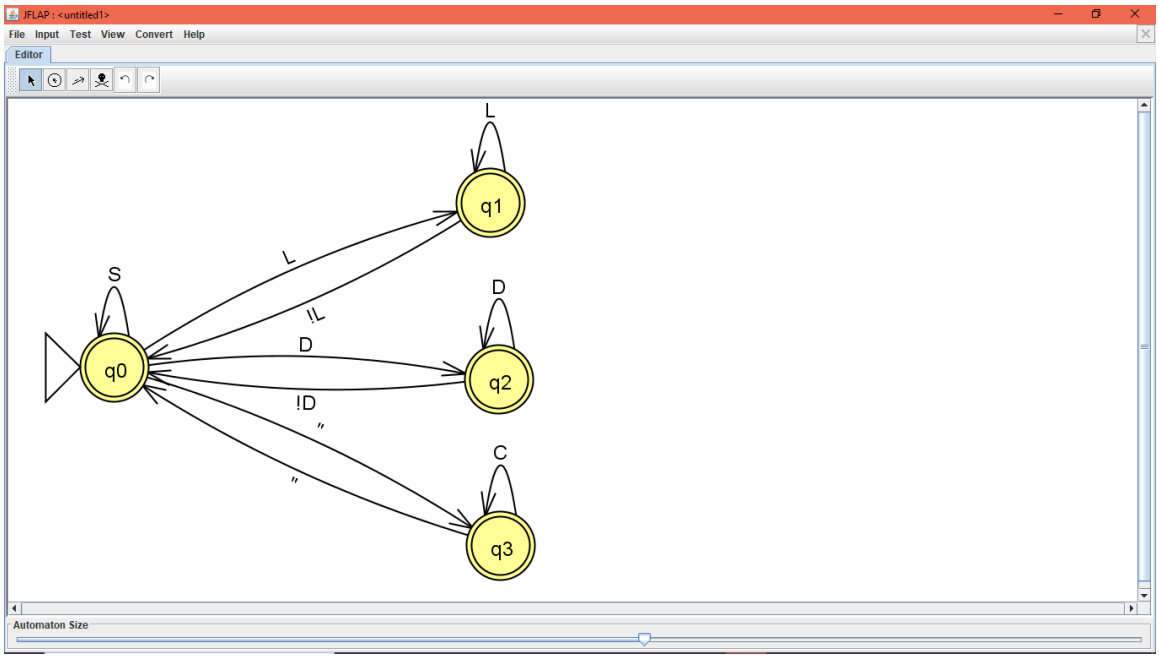
TERMINALES	NUMERACIÓN	SIGUIENTES
P	1	1, 2, 3, 6, 7
D	2	1, 2, 3, 6, 7
“	3	4, 5,
C	4	4, 5,
“	5	1, 2, 3, 6, 7
S	6	1, 2, 3, 6, 7
#	7	

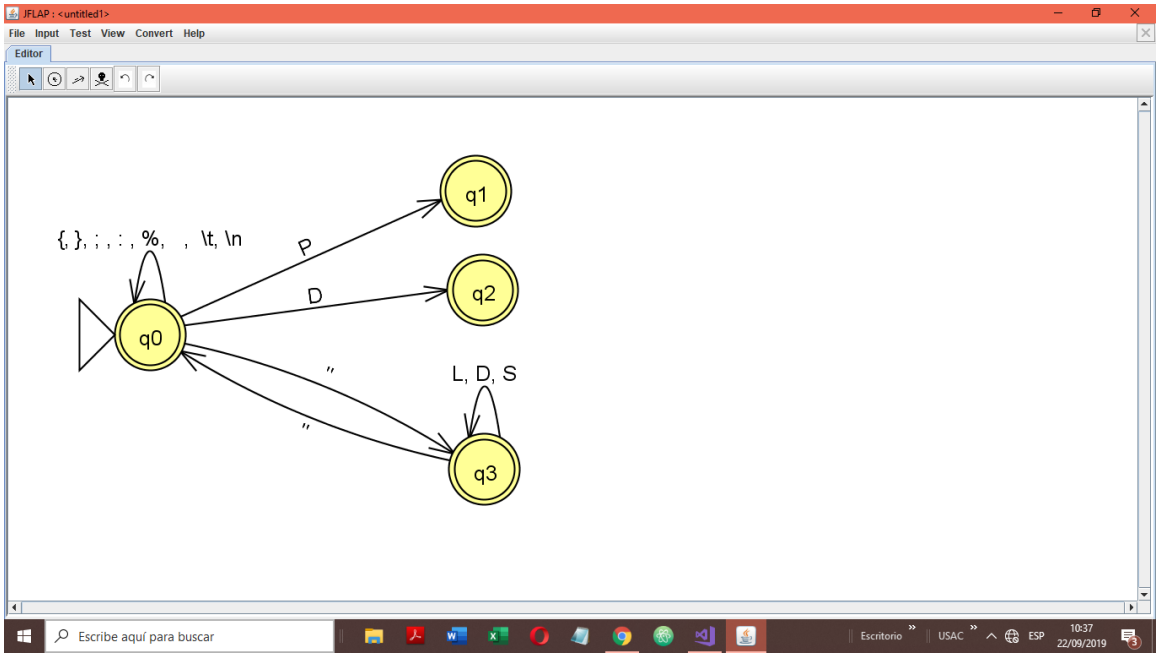
Tabla de transiciones:

ESTADOS/TERMINALES	P	D	“	C	S
q0 ={1, 2, 3, 6, 7}	q0	q0	q1={4, 5}	--	q0
q1={4, 5}	-	-	q0	q1	--

AFD:




JFLAP : <untitled1>
File Input Test View Convert Help
Editor
L
q1
L
S
!L
D
q0
D
q2
!D
"
"
C
q3
Automaton Size


JFLAP : <untitled1>
File Input Test View Convert Help
Editor
{, }, ; , : , %, , \t, \n
P
q1
D
q2
q0
"
L, D, S
"
q3
Escribe aquí para buscar
Escritorio
USAC
ESP
10:37
22/09/2019