

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ESCUELA DE CIENCIAS Y SISTEMA

ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 1

SECCIÓN A

AUX. MYNOR RENE RUIZ GUERRA



MANUAL TÉCNICO

DANIEL ROLANDO SOTZ ALVARADO

201430496

1ER. SEMESTRE 2023

INTRODUCCION

El siguiente proyecto analiza un lenguaje específico ya detallado por medio de la lectura de un archivo con extensión “olc” o por medio del ingreso directamente en el área de texto que posee la aplicación. Este aplicativo es capaz de realizar el Método del Árbol y el Método de Thompson, para que los estudiantes puedan verificar que las respuestas de las tareas y exámenes del curso son correctas. En el archivo ingresado acepta definición de conjuntos, expresiones regulares y lexemas para evaluar, todos estos nos ayudarán a construir los árboles, autómatas finitos deterministas como no deterministas de los métodos mencionados anteriormente. Finalmente también muestra los resultados como imágenes, archivos dot y json.

Objetivo General

Aplicar los conocimientos sobre la fase de análisis léxico y sintáctico de un compilador para la construcción de un software que permita generar análisis por medio del método del árbol.

Objetivos Específicos:

- Aplicar los conceptos del método de Árbol
- Programar el proceso de reconocimiento de lexemas dados mediante el uso de AFD
- Aplicar conceptos de método de Thompson de expresiones regulares

Alcances del Proyecto

La intención del proyecto es que el encargado del desarrollo de esta aplicación aprenda a utilizar las herramientas que nos proveen las librerías de JFlex y Cup, a la vez que se pone en práctica también los conceptos aprendidos en las clases magistrales de cursos anteriores como del curso actual organización de lenguajes y compiladores.

Especificación

técnica

Los requerimientos mínimos para uso del programa son:

- Lenguajes implementados: JAVA, HTML, DOT
- IDE Usado: NetBeans versión 8.2
- Editor de Código: Visual Studio Code
- Sistema Operativo: Windows 10 Home (64 bits)
- Interfaz Gráfica: AWT, SWING, HTML
- Librerías implementadas: AWT, SWING, JFlex, CUP

Lógica del programa:

El programa sigue un flujo de trabajo, para el desarrollo de la aplicación, a continuación, se muestra el flujo de forma general:

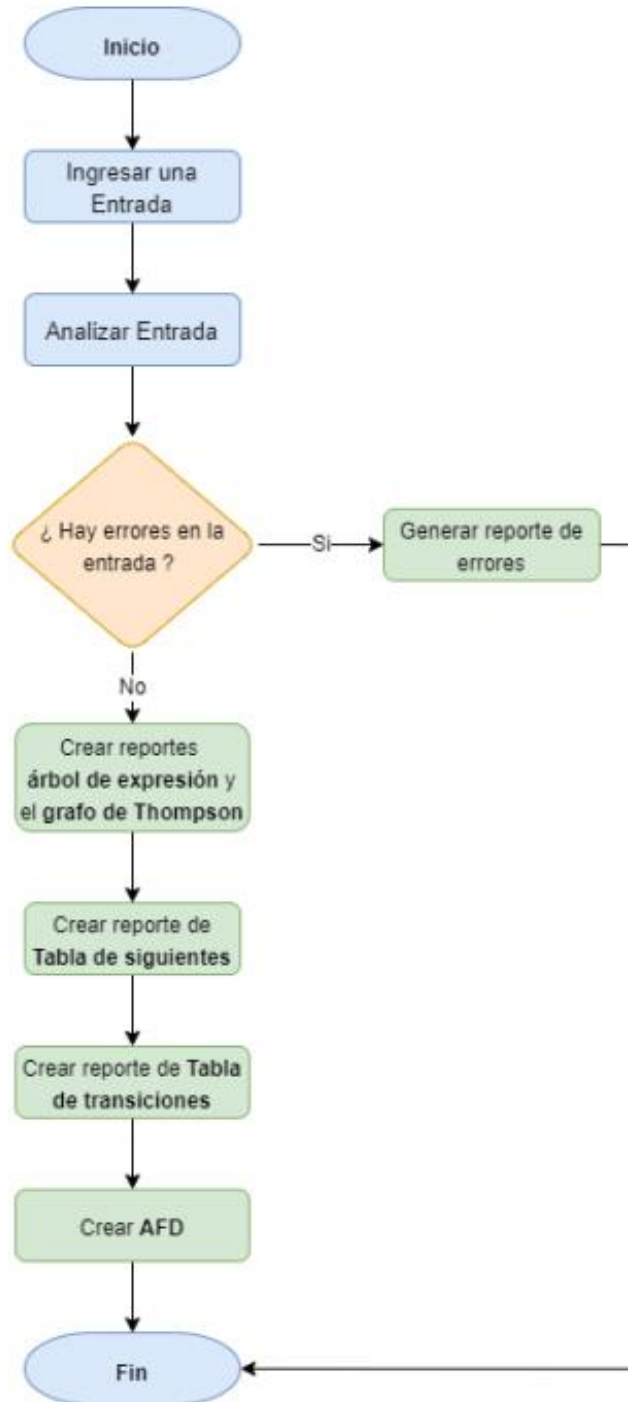
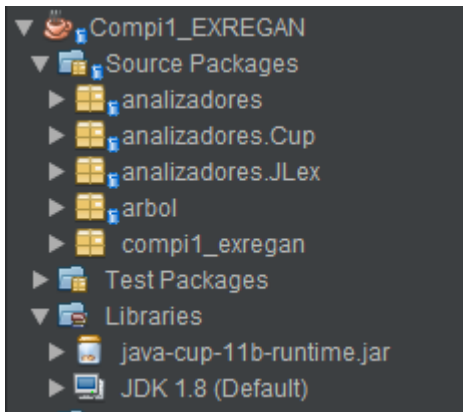


Diagrama de flujo para la creación de autómatas finitos determinista

Distribución de paquetes:



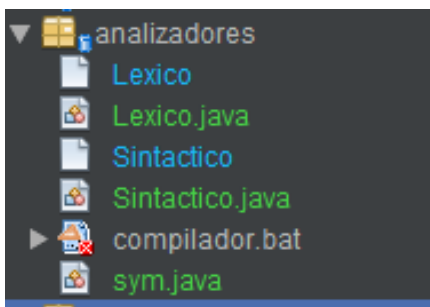
Esta es la distribución general del proyecto a grandes rasgos

Analizadores: Donde esta los archivos que JFlex y Cup leerán y generarán

Analizadores.Cup y analizadores.JLex: Donde estarán alojados las librerías necesarias para el proyecto

Y en los demás donde estará la lógica e interfaz del sistema.

Para mayor entendimientos aquí se especifica la carpeta analizadores, pues ella contiene las partes fundamentales del proyecto:



Léxico: Archivo que almacenará el código fuente con el cual JLex generará el analizador léxico que queremos

Sintáctico: Archivo donde se define la sintaxis del lenguaje usado.

Léxico.java: Archivo generado por JLex en lenguaje java para ser usado por Cup luego

Sintactico.java: Archivo generado por CUP en lenguaje java.

Compilador.bat: guardará los comandos que deben ejecutarse para solicitarle a JLex y a Cup que generen los analizadores.

Clases de la lógica del proyecto:



ER y declaración de terminales Utilizados en el JFLEX

```
BLANCO = [ \r\t]+
D= [0-9]+
DD=[0-9]+("."[ 0-9]+)?
COMENTARIO = "//".*
COMENTMULTI    = "<!"([^/!>]|[^!>]|[^/!>])*"!>"
CADENA = [""]([^\\"\\n|\\\\"")*[""]
MAY = [A-Z]
MIN = [a-z]
COMILLASDOBLES = [""]
COMILLA = ['']
FINSECCION = "%%"
IDENTIFICADOR = [A-Za-z]["_0-9A-Za-z"]*
CESPECIAL= [\32-\125]

%%

"CONJU" {return new Symbol(sym.RCONJU,yyline,yychar,yytext());}

{BLANCO} {}
{COMENTMULTI} {System.out.println("Este es un coment "+yytext()+"", en
{COMENTARIO} {}
```

```

";" {return new Symbol(sym.PTCOMA,yyline,yychar, yytext());}
{CADENA} {return new Symbol(sym.CADENA,yyline,yychar, yytext());}
\.. {return new Symbol(sym.CONCATENACION,yyline,yychar, yytext());}
{" {return new Symbol(sym.LLABRE,yyline,yychar, yytext());}
}" {return new Symbol(sym.LLCIERRA,yyline,yychar, yytext());}
{FINSECCION} {return new Symbol(sym.FINSECCION,yyline,yychar, yytext());}
{COMILLA} {return new Symbol(sym.COMILLA,yyline,yychar, yytext());}
":" {return new Symbol(sym.DOSPUNTOS,yyline,yychar, yytext());}
"- " {return new Symbol(sym.GUION,yyline,yychar, yytext());}
">" {return new Symbol(sym.MAYORQ,yyline,yychar, yytext());}
"," {return new Symbol(sym.COMA,yyline,yychar, yytext());}
"~" {return new Symbol(sym.RANGO,yyline,yychar, yytext());}
"| " {return new Symbol(sym.DISYUNCION,yyline,yychar, yytext());}
"*" {return new Symbol(sym.CEROMAS,yyline,yychar, yytext());}
"+" {return new Symbol(sym.UNOMAS,yyline,yychar, yytext());}
"?" {return new Symbol(sym.CEROUNO,yyline,yychar, yytext());}
"\n"|"\"|'|"\" {return new Symbol(sym.ESP,yyline,yychar, yytext());}
\n {yychar=1;}
{COMILLASDOBLES} {return new Symbol(sym.COMILLASDOBLES,yyline,yychar, yytext());}
{MAY}|{MIN}|{D} {return new Symbol(sym.ALFABETO,yyline,yychar, yytext());}
{D} {return new Symbol(sym.DE,yyline,yychar, yytext());}
{DD} {return new Symbol(sym.PTCOMA,yyline,yychar, yytext());}
{IDENTIFICADOR} {return new Symbol(sym.PTCOMA,yyline,yychar, yytext());}
{CESPECIAL} {return new Symbol(sym.PTCOMA,yyline,yychar, yytext());}

```

GRAMATICA UTILIZADA

```

inicio ::= LLIZQ instrucciones LLDER
        | error

instrucciones ::= SecDefER FinalSeccion SecLexemas

FinalSeccion ::= FINSECCION FinalSeccion
              | FINSECCION

SecDefER ::= conjunto SecDefER
          | er secDefER
          | conjunto
          | er
          | error SecDefER
          | error

SecLexemas ::= lexema SecLexemas
            | lexema
            | error SecLexemas
            | error

lexema ::= IDENTIFICADOR DOSPUNTOS CADENA PUNTOYCOMA

conjunto ::= RESERVADA DOSPUNTOS IDENTIFICADOR GUION MAYORQ notacionConj PUNTOYCOMA

notacionConj ::= ALFABETO COMA notacionConj
              | ALFABETO RANGO ALFABETO
              | ALFABETO
              | CESPECIAL COMA notacionConj
              | CESPECIAL RANGO CESPECIAL
              | CESPECIAL

er ::= IDENTIFICADOR GUION MAYORQ ERpolaca PUNTOYCOMA

```

```

ERpolaca ::= CEROMAS LLAVEIZQ IDENTIFICADOR LLAVEDER
| CEROMAS ESP
| CEROMAS CADENA
| CEROMAS ERpolMu1
| UNOMAS LLAVEIZQ IDENTIFICADOR LLAVEDER
| UNOMAS ESP
| UNOMAS CADENA
| UNOMAS ERpolMu1
| CEROUNO LLAVEIZQ IDENTIFICADOR LLAVEDER
| CEROUNO ESP
| CEROUNO CADENA
| CEROUNO ERpolMu1
| CONCATENACION CADENA ERpolaca
| CONCATENACION ERpolaca CADENA
| CONCATENACION LLAVEIZQ IDENTIFICADOR LLAVEDER CADENA
| CONCATENACION CADENA LLAVEIZQ IDENTIFICADOR LLAVEDER
| CONCATENACION CADENA CADENA
| CONCATENACION ESP CADENA
| CONCATENACION CADENA ESP
| CONCATENACION ERpolaca ERpolaca
| CONCATENACION LLAVEIZQ IDENTIFICADOR LLAVEDER ERpolaca
| CONCATENACION ERpolaca LLAVEIZQ IDENTIFICADOR LLAVEDER
| CONCATENACION LLAVEIZQ IDENTIFICADOR LLAVEDER LLAVEIZQ IDENTIFICADOR LLAVEDER
| CONCATENACION ESP ERpolaca
| CONCATENACION LLAVEIZQ IDENTIFICADOR LLAVEDER ESP
| CONCATENACION ESP LLAVEIZQ IDENTIFICADOR LLAVEDER
| CONCATENACION ERpolaca ESP
| CONCATENACION ESP ERpolaca
| CONCATENACION ESP ESP
| DISYUNCION CADENA ERpolaca
| DISYUNCION ERpolaca CADENA
| DISYUNCION LLAVEIZQ IDENTIFICADOR LLAVEDER CADENA
| DISYUNCION CADENA LLAVEIZQ IDENTIFICADOR LLAVEDER
| DISYUNCION CADENA CADENA
| DISYUNCION ESP CADENA
| DISYUNCION CADENA ESP
| DISYUNCION ERpolaca ERpolaca
| DISYUNCION LLAVEIZQ IDENTIFICADOR LLAVEDER ERpolaca
| DISYUNCION ERpolaca LLAVEIZQ IDENTIFICADOR LLAVEDER
| DISYUNCION LLAVEIZQ IDENTIFICADOR LLAVEDER LLAVEIZQ IDENTIFICADOR LLAVEDER
| DISYUNCION ESP ERpolaca
| DISYUNCION LLAVEIZQ IDENTIFICADOR LLAVEDER ESP
| DISYUNCION ESP LLAVEIZQ IDENTIFICADOR LLAVEDER
| DISYUNCION ERpolaca ESP
| DISYUNCION ESP ERpolaca
| DISYUNCION ESP ESP

```