

MANUAL TECNICO

Entornos que el que se desarrolló EXREGAN

Windows

- Procesador: Intel Core i7-10510U CPU @ 1.80GHz 3.10GHz
- Memoria: 16GB RAM.
- Disco memoria: 500GB SSD
- java versión "1.8.0_341"

librerías para el look and feel:

- flat-1.03.jar

librerías para el manejo de archivos Json:

- gson-2.8.0-javadoc.jar
- gson-2.8.0-sources.jar
- gson-2.8.0.jar

librerías para realizar los análisis léxico y sintáctico:

- java-cup-11b-runtime.jar
- java-cup-11b.jar
- jflex-1.6.1.jar

CLASES:

- **AFD.java:** Esta clase pertenece a los objetos de clase AFD, en ella se encuentran métodos para el manejo de información perteneciente a un autómata finito determinista dentro del programa.
- **Analizador_lexico:** Esta clase, generada a partir de la clase GLexico, pertenece a la herramienta JFlex, herramienta externa empleada para el análisis léxico archivo de entrada. Este archivo se genera apartir del archivo A_Lexico.jflex.
- **Árbol.java:** Esta clase pertenece a los objetos de la clase Árbol, en ella se encuentran los métodos para crear arboles binarios capaces de almacenar información de las expresiones regulares, para posteriormente crear AFD's a partir de esa información.
- **Conjunto.java:** Esta clase pertenece al os objetos de la clase Conjunto, en ella se almacenan los identificadores y los elementos de los conjuntos declarados en los archivos

de entrada del programa.

- **EstadoAFD.java:** Esta clase pertenece a los objetos de la clase EstadoAFD, que representan los estados de un autómata finito determinístico.
- **Expresiones.java:** Esta clase pertenece a los objetos del tipo Expresiones, los cuales se encargan de almacenar las expresiones regulares y sus respectivos árboles.
- **GLexico:** Esta clase se encarga de tomar la información declarada en el archivo A_Lexico.jflex y lo convierte en lenguaje Java de programación comprensible para el compilador java.
- **GSintactico:** Esta clase se encarga de tomar la información declarada en el archivo A_Sintactico.cup y lo convierte en lenguaje Java de programación comprensible para el compilador java.
- **Interfaz.java:** Esta clase se encarga de almacenar la información y los componentes de la interfaz grafica del proyecto.
- **Nodo.java:** Esta clase se encarga de almacenar la información correspondiente a un nodo hoja perteneciente a un objeto de la clase árbol.

6

- **OLCProyecto201902888.** Java: Esta clase es la clase principal del programa, esta se encarga de crear una instancia de la clase Interfaz para poder utilizar las herramientas del programa.
- **Símbolos.java:** Esta clase, generada a partir de la clase GSintactico, pertenece a la herramienta CUP, una herramienta externa utilizada para el análisis sintáctico del archivo de entrada. En ella se almacenan los símbolos declarados en los archivos A_Sintactico.cup y A_Lexico.jflex.
- **Sintactico.java:** Esta clase, generada a partir de la clase GSintactico, pertenece a la herramienta CUP, una herramienta externa utilizada para el análisis sintáctico del archivo de entrada. En ella se almacenan los símbolos declarados en los archivos A_Sintactico.cup y A_Lexico.jflex.
- **TransicionAFD.java:** Esta clase pertenece a los objetos de la clase TransicionAFD, los cuales se encargan de representar las transiciones de un automata finito determinista.
- **validacion_cadenas.java:** Esta clase pertenece a los objetos del tipo Validacion, los cuales se encargan de almacenar la expresión a la que se desea evaluar una cadena y la cadena

a evaluar, los cuales son definidos en el campo de texto.

METODOS EMPLEADOS

- **AbrirbtnActionPerformed():** Este método se encarga de abrir un archivo existente.
- **AnalizarbtnActionPerformed():** Este método se encarga de invocar los métodos que se encargan del análisis de las cadenas que deben ser validadas.
- **AnteriorbtnActionPerformed():** Este método se encarga de cambiar la imagen actual del visualizador a la imagen que le antecede.
- **AnulabilidadEspeciales():** Este método se encarga de calcular la anulabilidad de los nodos de los árboles que son operadores.
- **AsignacionSiguietes():** Este método se encarga de asignar los siguientes a cada nodo del árbol para calcular las transiciones.
- **CalcSiguietes():** Este método se encarga de invocar a los métodos que calculan los siguientes y generan la tabla de siguientes.
- **CalcularAnulabilidad():** Este método se encarga de calcular la anulabilidad de cada uno de los nodos del árbol de símbolos que no sean operadores.
- **CalcularPrimerosUltimos():** Este método se encarga de invocar a los métodos que calculan los primeros y los últimos de cada nodo.
- **crearTransiciones():** Este método se encarga de crear las transiciones de los autómatas.
- **GeneracionDirectorios():** Este método se utiliza para crear los directorios utilizados por el programa.
- **GenerarAFD():** Este método se encarga de graficar los autómatas finitos deterministas de las expresiones regulares evaluadas en el archivo de entrada.
- **GenerarbtnActionPerformed():** Este método de encarga de invocar a los métodos de generación del método del árbol y autómatas.
- **GenerarDot():** Este método se encarga de generar los archivos .dot utilizados para realizar las gráficas.
- **GenerarTablaTransiciones():** Este método se encarga crear las transiciones y asignarlas a los automatas.

- **GraficaNodos():** Este método se encarga de escribir nodo a nodo en el archivo .dot.
- **GraficarSiguietes():** Este método se encarga de graficar la tabla de siguientes de los autómatas finitos determinísticos.
- **GraficarSintactico():** Este método se encarga invocar el método que escribe en el archivo .dot los nodos de los árboles de expresiones y grafica el árbol.
- **GraficarTablaSiguietes():** Este método se encarga de iniciar la traficación de la tabla de siguientes.
- **GraficarTransiciones():** Este método se encarga de graficar la tabla de transiciones de los autómatas finitos deterministas.
- **GuardarbtnActionPerformed():** Este método se encarga de guardar los cambios efectuados a un archivo de entrada.
- **GuardarComobtnActionPerformed():** Este método se encarga de guardar el archivo actual como un archivo nuevo.
- **ImprimirTablaTransiciones():** Este método se encarga de graficar la tabla de transiciones del automata finito determinista de la expresiones regulares.
- **NuevoArchivobtnActionPerformed():** Este método se encarga de generar un archivo nuevo.
- **PrimerosEspeciales():** Este método se encarga de calcular los primeros de los nodos especiales (nodos que contienen operadores).
- **recorrerArbol():** Este método se encarga de recorrer el Árbol de símbolos.
- **RecorrerSiguietes():** Este método se encarga de recorrer el árbol para calcular los siguientes.
- **reportar():** Este método se encarga de generar las imágenes jpg a partir de los archivos .dot.
- **SeleccionTipoActionPerformed():** Este método se encarga de cambiar el tipo de visualización de imágenes.
- **SiguientebtnActionPerformed():** Este método se encarga de cambiar la imagen actual del visualizador a la imagen que le sigue.
- **Siguietes():** Este método se encarga de calcular los siguientes de cada nodo.

- **UltimosEspeciales():** Este método se encarga de calcular los últimos de los nodos especiales (nodos que contienen operadores).