Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Organización de Lenguajes y Compiladores 1

Sección B

Fecha: 06/03/2022

**MANUAL TÉCNICO**

Nombre: Carné:

Gerson Rubén Quiroa del Cid 2020 00166

Índice

[Requisitos del sistema 4](#_Toc97113625)

[Para Windows 4](#_Toc97113626)

[Mac OS 4](#_Toc97113627)

[Linux 4](#_Toc97113628)

[Introducción 5](#_Toc97113629)

[Interfaz gráfica 6](#_Toc97113630)

[Clase Inicio (Main) 6](#_Toc97113631)

[Botón crear nuevo archivo y abrir un archivo existente 6](#_Toc97113632)

[Botón guardar como y guardar 7](#_Toc97113633)

[Botón analizar 8](#_Toc97113634)

[Jlabel, combo box y botón siguiente 9](#_Toc97113635)

[Clase Archivos 10](#_Toc97113636)

[Función analizador léxico y sintáctico: 10](#_Toc97113637)

[Archivos jflex y jcup 11](#_Toc97113638)

[Jflex 12](#_Toc97113639)

[Jcup 13](#_Toc97113640)

[Paquete clases 13](#_Toc97113641)

[Clase AFD 13](#_Toc97113642)

[Función tablaTransciones 14](#_Toc97113643)

[Función funciónEvaluar 14](#_Toc97113644)

[Clase AFN 14](#_Toc97113645)

[Función add 15](#_Toc97113646)

[Clase AST 15](#_Toc97113647)

[Función add 16](#_Toc97113648)

[Función anulables y primeroUltimos 17](#_Toc97113649)

[Función siguientes y recorridoSiguientes 17](#_Toc97113650)

[Función transiciones y nuevaTransicion 18](#_Toc97113651)

[Clase Conjunto 18](#_Toc97113652)

[Clase Error 18](#_Toc97113653)

[Clase ExpresionRegular 19](#_Toc97113654)

[Clase Html 19](#_Toc97113655)

[Clase RegExpEvaluar 20](#_Toc97113656)

[Clase Siguiente 20](#_Toc97113657)

[Clase Transicion 20](#_Toc97113658)

# Requisitos del sistema

Para la instalación de la aplicación, su computadora y/o laptop debe cumplir

como mínimo los siguientes requerimientos:

## Para Windows

* Windows Vista SP2 (8u51 y superiores)
* Windows Server 2008 R2 SP1 (64 bits)
* Windows Server 2012 y 2012 R2 (64 bits)
* RAM: 128 MB
* Procesador: Mínimo Pentium 2 a 266 MHz
* Exploradores: Internet Explorer 9 y superior, Firefox

## Mac OS

* Mac con Intel que ejecuta Mac OS X 10.8.3+, 10.9+
* Privilegios de administrador para la instalación
* Explorador de 64 bits
* Se requiere un explorador

## Linux

* Oracle Linux 7.x (64 bits)2(8u20 y superiores)
* Red Hat Enterprise Linux 7.x (64 bits)2(8u20 y superiores)
* Suse Linux Enterprise Server 12.x (64 bits)2(8u31 y superiores)
* Ubuntu Linux 12.04 LTS, 13.x
* Ubuntu Linux 15.10 (8u65 y superiores)

**IMPORTANTE:** Independientemente del sistema operativo del usuario, es necesario que previamente instale Java en su computadora.

**Editor utilizado:** NetBeans IDE 8.2

**Versión de Java utilizado:** JDK "1.8.0\_241"

# Introducción

El manual técnico tiene como finalidad de explicar el funcionamiento de cada función utilizado en el presente programa, con el objetivo de que no haya confusión al leer el código del programa y cualquier persona con conocimiento básico de programación en Java pueda comprender lo que se hizo en cada línea de código. Por lo cual se recomienda que toda persona que quiera leer o modificar el código vea este manual para ahorraste tiempo y así sea más fácil su comprensión.

# Interfaz gráfica

Estas clases son las que conforman la interfaz gráfica, solo se utilizó una clase y algunas ventanas emergente de confirmación o notificación.

## Clase Inicio (Main)

Esta clase es la principal del programa donde inicialmente se ejecutará el menú con las diferentes opciones que puede elegir el usuario, dependiendo de la opción que se elija el usuario se ejecutarán las diferentes funciones detalladas más adelante.

Texto

Descripción generada automáticamente

## Botón crear nuevo archivo y abrir un archivo existente

Dependiendo de que botón se precios se abrirá una ventana para que el usuario escriba la ruta donde quiere que se cree el nuevo archivo, o una ventana donde podrá elegir el archivo que quiera abrir.

Texto

Descripción generada automáticamente

## Botón guardar como y guardar

Estos botones como su nombres lo indican, el primero se desplegará una ventana donde el usuario ingresará una ruta donde quiera que el archivo se guarde y el segundo botón para simplemente guardar en el mismo archivo ya creado.

Captura de pantalla de computadora

Descripción generada automáticamente

## Botón analizar

En cuanto a la interfaz, este botón es el más importante ya que en él se realizan todas las operaciones y llamadas a métodos para la creación de todo el programa, como lo son el analizador léxico y sintáctico, así como también la creación del árbol, autómata finito no determinista y el autómata finito determinista.

Texto

Descripción generada automáticamente

## Jlabel, combo box y botón siguiente

Estos 3 componentes se utilizaron para el manejo de las gráficas previamente creadas mediante el botón de analizar. Con el combo box podemos movernos por las diferentes carpetas, con el botón siguiente podemos pasar por cada gráfica creada y en el label se mostrará cada gráfica.

Texto

Descripción generada automáticamente

# Clase Archivos

Esta clase es la encargada de guardar diferentes archivos como lo son archivos json, así como también los métodos para guardar un nuevo archivo y guarda como.

Texto

Descripción generada automáticamente

## Función analizador léxico y sintáctico:

Estas funciones son las más importantes en esta clase, ya que por acá pasa el analizador léxico y sintáctico que generamos mediante jflex y jcup.

En la función del analizador léxico, se lee cada token y se valida si es un token perteneciente al lenguaje o en su defecto un token desconocido, que generará un error léxico.

Por otra parte, en la función del analizador sintáctico, se analiza el archivo de entrada y éste devuelve un string que dirá si el contenido ha sido aceptado o por su defecto, que el contenido contiene un error de tipo sintáctico.

Texto

Descripción generada automáticamente

# Archivos jflex y jcup

Estos archivos y clases se crearon con ayuda de las librerías ya mencionadas, dentro de estos dos archivos creamos la lógica de nuestros analizadores tanto léxico como sintáctico. Una vez escrito dichos archivos, en una clase main, compilamos los archivos y éstos crearán las clases necesarias que serán nuestros analizadores.

## Jflex

Imagen de la pantalla de un computador

Descripción generada automáticamente con confianza baja

## Jcup

Texto

Descripción generada automáticamente

# Paquete clases

# Clase AFD

Esta clase es la que nos ayuda a pasar de la tabla de transiciones previamente hecha en la clase AST, hacia una estructura de tipo AFD, para luego poder recorrer el autómata con las cadenas a evaluar y poder saber si la cadena es aceptada o no.

Una captura de pantalla de un celular con texto e imagen

Descripción generada automáticamente con confianza media

## Función tablaTransciones

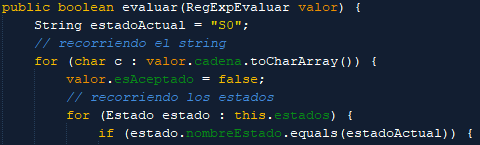
Esta función recibe la tabla de transiciones proporcionada por la clase AST y esta crea una estructura a partir de la tabla.

Texto

Descripción generada automáticamente

## Función funciónEvaluar

Esta función es la que se encarga de recibir una cadena la cual queremos evaluar si es aceptada mediante los autómatas creados. Si es aceptada devuelve un true, de lo contrario regresa un false.



# Clase AFN

Esta clase es la encargada de crear el autómata finito no determinista con el método de Thompson. Este autómata se crea con ayuda del árbol, ya que se recorre el árbol y se construye a través de este.

Texto

Descripción generada automáticamente

## Función add

Esta función ayuda a crear el AFN, este método es extenso ya que para cada operación existe una forma estándar de hacerlo, por lo que para la concatenación (.), como para el or (|), como para las cerraduras, etc. Tiene su forma diferente de crearlos y por lo tanto se debe tener en cuenta cada posibilidad.

Texto

Descripción generada automáticamente

# Clase AST

Esta clase se encarga de crear el árbol binario que genera cada expresión regular ingresada en el archivo fuente. Esta clase es una de las más larga ya que dentro de ella están las funciones para crear el árbol, anulables, primeros y últimos, los siguientes y tabla de transiciones.

Captura de pantalla de computadora

Descripción generada automáticamente

## Función add

Esta función ayuda a crear el árbol binario mediante la expresión regular proporcionada por el usuario. Cabe resaltar que la expresión regular está dada en notación polaca, por lo que se creo de la misma manera el árbol. Para obtener la misma expresión se deberá de recorrer en pre-orden, si se quiere en notación normal se recorrerá en orden.

Texto

Descripción generada automáticamente

## Función anulables y primeroUltimos

Estas funciones como sus nombres lo indican, se utiliza para darle a cada nodo del árbol, sus anulables y primeros y últimos. Para poder ir en un orden ascendente, desde sus hojas hasta su raíz, se recorrió en post orden.

Texto

Descripción generada automáticamente

## Función siguientes y recorridoSiguientes

Estas funciones son las encargadas de crear la tabla de siguientes del árbol generado anteriormente. Cuando se termine de crear la tabla de siguientes, se llama a una función que crea un html con la tabla e información necesaria.

Texto

Descripción generada automáticamente

## Función transiciones y nuevaTransicion

Estas funciones crean la tabla de transiciones a partir de la tabla de siguientes. Si existe una nueva transición, se llama a la función nuevaTransicion sino se termina la recursividad y por último se llama a la función que creará el html con la tabla de transiciones.

Una captura de pantalla de un celular

Descripción generada automáticamente con confianza media

# Clase Conjunto

Esta clase nos servirá para guardar información de los conjuntos que se definen en el archivo de entrada.

Texto

Descripción generada automáticamente

# Clase Error

Esta clase es la encargada de guardar los errores que existan en el archivo de entrada tanto léxico como sintácticamente.

Texto

Descripción generada automáticamente

# Clase ExpresionRegular

Esta clase guarda las expresiones regulares que se definieron en el archivo de entrada.

Texto

Descripción generada automáticamente

# Clase Html

Esta clase se encarga de crear cada reporte que sea necesario en html. Los reportes que se crearán cuando se necesiten son los siguientes: reporte de errores (reportErrores()), reporte de siguientes (reportSiguientes()) y reporte de transiciones (reportTransicion()), a continuación se mostrará uno de ellos.

Texto

Descripción generada automáticamente

# Clase RegExpEvaluar

Esta clase guarda las cadenas a evaluar que se ingresaron en el archivo de entrada y la información importante como, por ejemplo, si es aceptada la cadena o no.

Captura de pantalla de computadora

Descripción generada automáticamente

# Clase Siguiente

Esta clase guarda información sobre los siguientes que nos proporcionan los árboles binarios.

Captura de pantalla de un celular

Descripción generada automáticamente

# Clase Transicion

Esta clase guarda información sobre cada transición, o más bien cada estado, dentro de esta clase se sabrá hacia que otro estado se irá dependiendo de la información que contenga.

Captura de pantalla de computadora

Descripción generada automáticamente