

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Organización de Lenguajes y Compiladores #1



Elder Anibal Pum Rojas

201700761

Guatemala, 9 de Marzo del 2022

INTRODUCCIÓN

ExpAnalyzer nace de la necesidad de querer realizar, de forma automatizada y rápida, los métodos del árbol y Thompson para la clase de Organización de Lenguajes y Compiladores #1. El proyecto realiza dichos métodos, además de generar autómatas determinísticos y no determinísticos (árbol y Thompson respectivamente).

Se ingresa un archivo de extensión “.exp, .Exp, .EXP” como entrada y generará como salida el árbol de expresión completo, eso incluye los primeros, últimos, anulabilidad y el símbolo a evaluar. Además, genera como salida la tabla de siguientes, la tabla de estados y el AFD generado, así como también el AFND, producto del método de Thompson.

HERRAMIENTAS UTILIZADAS

- Java JDK 1.8 (Pueden ser incluso versiones más recientes)
- Sistema Operativo Windows (XP, Vista, 7, 8.1, 10) o también otros SO como cualquier distro de Linux o incluso MacOS.
- JFlex-full-1.7.0.jar o JLex (utilizado para la generación del análisis léxico)
- Java-cup-11b-runtime.jar (utilizado para la generación del análisis sintáctico)
- Netbeans 8.2 como IDE (puede usarse IntelliJ IDEA o Eclipse)

CLASES A UTILIZAR

1. **Lexico.java:** Clase generada por la herramienta JFlex/JLex (a partir del archivo Lexico, ya que contiene todas las reglas del analizador léxico para su correcto funcionamiento)
2. **Sintactico.java:** Clase generada por la herramienta Cup (a partir del archivo Sintactico, ya que contiene todas las reglas del analizador sintáctico para su correcto funcionamiento)
3. **Errores.java:** Clase especializada para recopilar y analizar los errores globales de los archivos, tanto léxicos como sintácticos.
4. **Validation.java:** Clase que posee la estructura necesaria para realizar la evaluación de cadena con la expresión regular.
5. **HOJA.java:** Clase que nos permite trabajar lo que es el método del árbol y también determinar la anulabilidad de algunos símbolos.
6. **HOJA_AFN.java:** Clase que nos permite trabajar los AFN creando hijos para el método del árbol, analizando cada símbolo de una expresión.
7. **AFD.java:** Clase que permite trabajar por completo el método del árbol, así como también graficar el AFD, los siguientes, las transiciones e incluso trabaja parte del AFN.
8. **AFN.java:** Clase que permite trabajar exclusivamente el AFN en conjunto con HOJA_AFN.
9. **Main.java:** La clase main es la ventana principal que permite interactuar con el proyecto, dentro de ella se encuentran las distintas funciones de los botones y el menú, además nos permite visualizar cada imagen generada con las clases anteriores.