

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

10 de Marzo del 2021

# INTRODUCCIÓN

REGEXIVE 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 “.olc, .Olc, .OLC” 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 versiones más recientes)
- Sistema Operativo Windows 10 Home (pueden ser versiones anteriores como Windows 7 o Windows 8, así como Linux o MacOS)
- JFlex-full-1.7.o.jar (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 (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. **Recopilador.java:** Clase utilizada para, como su nombre lo indica, recopilar toda la información de la entrada a evaluar.
4. **Manejador.java:** Clase utilizada para manejar las instancias de objetos globales, tales como la ruta de graphviz, listas de expresiones, conjuntos, análisis y además, posee el método para generar el método del árbol.
5. **NodeArbol.java:** Clase utilizada para manejar toda la información de los nodos del árbol, así como sus elementos, como los primeros, últimos, anulabilidad y los símbolos a evaluar.
6. **Arbol.java:** Clase utilizada como estructura principal del método del árbol, acá están los métodos como la inserción de nodos, insertar los últimos, primeros, siguientes. Así también está la generación de la tabla de transiciones y los métodos que grafican el árbol, el afd y las tablas.
7. **ListaSiguiente.java:** Clase que posee la estructura para realizar la tabla de siguientes.
8. **ListaTransiciones.java:** Clase que posee la estructura para realizar la tabla de transiciones.
9. **Validacion.java:** Clase que posee la estructura necesaria para realizar la evaluación de cadena con la expresión regular.
10. **Pantalla.java:** Clase y Ventana que contiene el archivo main del proyecto. Sirve como interfaz de usuario para cargar el archivo de entrada y analizar dichas entradas en las demás clases.