

Manual Técnico

ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 1

Manual Técnico

Descripción

Se realizó un aplicación para la construcción autómatas finitos deterministas y no deterministas, implementando estructuras de datos, listas, recursividad y distintos métodos para su creación y generación de reportes gráficos capaces de hacer visibles el árbol binario, tabla de siguientes, tabla de transiciones y finalmente el AFD, se utilizo graphviz para la creación de los mismos, para la definición de estos se empleo el uso de jflex para manejo de tokens y cup para la gramática.

Datos técnicos

- Lenguaje utilizado: Java.
- IDE Utilizado: NetBeans Apache versión 16.
- Sistema operativo: Windows 10 (64 bits).

Dependencias:

- Java-cup-11b.jar
- Jflex.1.7.0.jar

Flujo de Trabajo

El usuario podrá cargar un archivo o escribirlo de 0 para su análisis, el botono análisis mandara a llamar a las distintas funciones que ejecuta la app.

Métodos utilizados

A continuación se dará una explicación del funcionamiento de los distintos métodos utilizados.

Manual Técnico

Paquete py1_201603028

Clase IG_Principal

1. **main:** hace un llamado a la clase Operaciones_Ig que almacena el método para compilar.
2. **Abrir_Archivo():** Permite abrir un archivo y mostrar su contenido en un componente de texto.
3. **guardarArchivo():** Permite guardar un archivo en su misma ubicación con formato .ocl.
4. **guardarArchivoComo():** Permite guardar un archivo con un nombre y ubicación deseada con formato .ocl.
5. **Analizador():** Realiza el análisis sintáctico de un texto ingresado en un componente de texto y muestra los resultados en otro componente. También genera reportes de errores.

Paquete ANALIZADOR

Clase Nodo

1. **generarCompilador():** Esta función genera el compilador del analizador léxico y sintáctico a partir de los archivos fuente en formato JFlex y Cup.
2. **getCodigoInterno():** Utilizando recursividad escribe la estructura para el diseño del árbol binario.
3. **getCodigo():** Función que genera la lista de siguientes recorriendo la lista HOJAS del objeto SIGUIENTES.
4. **getSO():** Genera la lista de transiciones para poder graficarlo, se realiza recorriendo la lista REDUCCIONES del objeto ESTADOS.
5. **Reducciones():** Realiza el mismo proceso que getSO para recorrer los estados y generarlos aplicando las reglas de unión y no repetidos.

Manual Técnico

Clase AFND

Objeto para la generación del AFND.

Clase ERRORES

Objeto de tipo ERRORES, utilizado para la creación de errores y almacenarlos en una lista.

Clase SIGUIENTES

Objeto de tipo SIGUIENTES, utilizada para la generación de la lista siguientes del AFD.

Clase ESTADOS

Objeto de tipo ESTADOS, utilizada para la generación de transiciones del AFD.

Clase Estado

Objeto de tipo Estado, utilizada para la generación del AFND.

Clase Transiciones

Objeto de tipo Transiciones, utilizada para la generación de transiciones en el AFND.

Clase Lexico

Generado por el archivo Lexico.jflexl, contiene los tokens.

Clase parser

Generado por el archivo sintactico.cup, contiene las gramáticas.

Clase sym

Generado por el achivo sintactico.cup, contiene los símbolos necesarios.

Manual Técnico

Clase Reportes

1. **Graficar_Tabla_Sig**(ArrayList<ArrayList<Object>> Tabla_Sig): Utilizada para graficar la tabla de siguientes.
2. **Tabla_Transiciones**(ArrayList<ESTADOS> Tabla_Tran, String Nombre): Función para la graficación de transiciones.
3. **Aceptacion**(ArrayList<SIGUIENTES> Tabla_Sig, ArrayList<ESTADOS> Tabla_Tran, String Nombre): Función utilizada para graficar el AFD.
4. **generarGraphviz**(): Función encargada de generar el AFND.
5. **GenerarReporte**(): Función que genera el html de reporte de errores.

Paquetes de reportes:

1. **AFD_201603028**: Almacena los reportes AFD.
2. **ARBOLES_201603028**: Almacena los arboles binarios generados.
3. **ERRORES_201603028**: Almacena el html del reporte de errores.
4. **SIGUIENTES_201603028**: Almacena las tablas de Siguientes.
5. **TRANSICIONES_201603028**: Almacena las tablas de transiciones.

Manual Técnico

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Proyecto [Compi1]

Primer semestre

Oscar Eduardo Morales Girón

201603028