Proyecto 1: Exregan

Manual Técnico

2023

ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 1

Damaris Julizza

Muralles Véliz

202100953

CONTENIDO

INTRODUCCION

Este manual, describe todos los aspectos técnicos del programa, a manera de familiarizar a la persona interesada con la lógica implementada en el desarrollo de este.

DESCRIPCIÓN GENERAL DEL PROGRAMA

Este programa es un sistema capaz de realizar el Método del Árbol y el Método de Thompson de expresiones regulares en notación polaca o prefija.

Permite realizar un análisis léxico y sintáctico para los archivos con lenguaje OLC que contendrán las expresiones regulares a ser analizadas para crear lo requerido en cada método.

DETALLES DE DESARROLLO

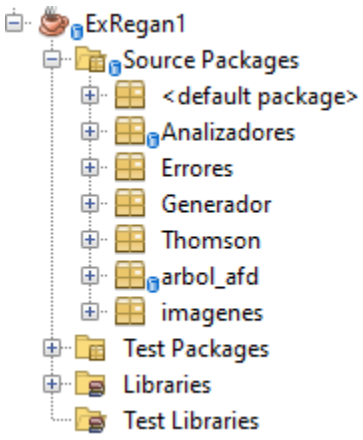
Este programa se desarrolló en lenguaje Java, a continuación, se listan las versiones del IDE y librerías utilizadas en el desarrollo:

- IDE: Apache NetBeans 12.6
- Java versión 8
- JDK 17.0.2
- Librería java-cub-11b
- Librería java-cup-11b-runtime
- Librería jflex-full-1.7.0

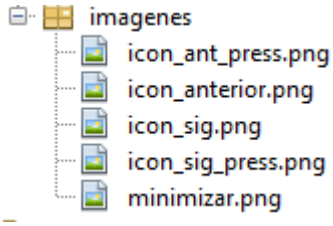
Lo detallado anteriormente, puede asegurar el correcto funcionamiento del programa.

LOGICA DEL PROGRAMA

En esta sección se explicará a detalle los paquetes, clases y archivos importantes dentro del programa:

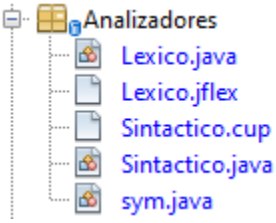


PAQUETE IMAGENES:



Este paquete contine todas las imágenes utilizadas en el diseño del programa.

PAQUETE ANALIZADORES



En este paquete se puede encontrar los archivos jflex y cup que contienen el código requerido para analizar el archivo de entrada para el programa, a continuación, se explica cada uno de ellos:

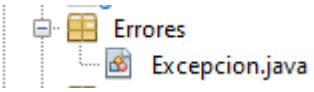
- ***LEXICO.JFLEX:***

  En este archivo se contiene el código necesario para realizar el análisis léxico de una entrada dada.

- ***SINTACTICO.CUP:***

  Este archivo contiene el código necesario para realizar el análisis sintáctico de cada componente léxico enviado por el analizador léxico, verificando que se encuentre en el orden correcto.

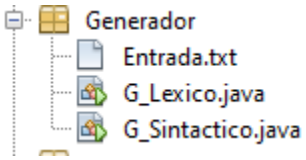
PAQUETE ERRORES



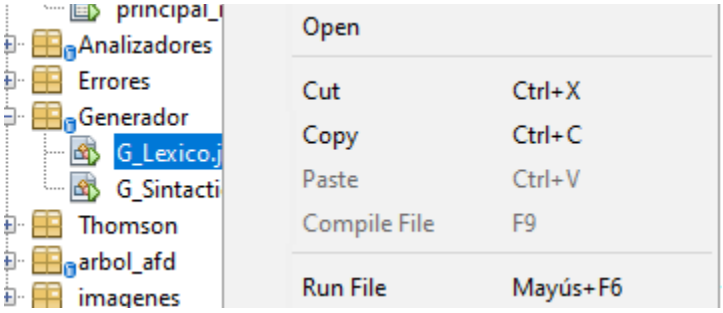
Este paquete contiene una clase denominada Excepcion que almacena todos los datos relevantes para un error, como el tipo de error, descripción, la fila y columna.

Esta clase es utilizada en los analizadores y en un método de la clase principal.

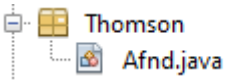
PAQUETE GENERADOR



Este paquete contiene dos clases que se encargan de generar el código en java para los archivos jflex y cup del paquete de analizadores, es decir que para cualquier cambio realizado en los archivos antes mencionados se debe de correr estos generadores para que los cambios puedan ser aplicados.

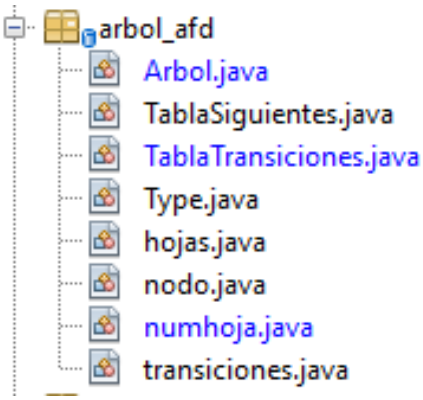


PAQUETE THOMSON



Este paquete contiene un único archivo, el cual es una clase java que guarda parámetros como el estado inicial, el final, la transición y la transición con épsilon, que son elementos importantes para la generación de un AFND.

PAQUETE ARBOL_AFD



Este paquete contiene todos los archivos con el código necesario para crear las gráficas de los métodos requeridos, árbol sintáctico, tabla de siguientes tablas de transiciones, AFD y AFND. A continuación, se explica más a detalle el contenido de cada clase:

- ***TYPE:***

  Este archivo es una clase que hace uso del método enum, para listar los distintos tipos de nodos que puede haber en el árbol sintáctico.

- ***HOJAS:***

  Esta clase contiene tres métodos que manejan la información importante de todos los nodos de tipo hoja.

- ***NUMHOJA:***

  Esta clase recibe la cadena de la expresión regular y se encarga de encontrar la cantidad de hojas presentes en dicha expresión, para posterior mente asignar este valor a cada nodo hoja al llamar a su método getNum.

- ***NODO:***

  Esta clase almacena toda la información importante y necesaria para cada nodo presente en el árbol, de esta forma se puede organizar de forma jerárquica a los nodos.

  Esta clase también contiene dos métodos llamados getNode y follow, los cuales organizan la información y nos devuelven el nodo raíz del árbol y los siguientes de cada nodo correspondientemente.

- ***TRANSICIONES:***

  Esta clase guarda la información importante para las transiciones del autómata, como el estado inicial, el siguiente y el elemento con el cual se realiza la transición.

- ***ARBOL:***

  Esta clase primero forma una lista de todos los caracteres de la expresión regular y luego les asigna los valores que les corresponden pasando dichos valores a una lista de nodos.

  Posee un método imTree que forma una cadena con el código requerido en graphviz para diseñar el diagrama del árbol, posee también un método Thomson que realiza la misma acción que el anterior descrito.

  El método TransicionesThomson es un método recursivo que es llamado para constantemente dependiendo de si el nodo es una cerradora positiva, de kleene, etc. para formar las transiciones con épsilon caracteristicas de este método.

  Por último, contiene el método GenerarDot que convierte la cadena en un archivo dot y el método generarpng que convierte este archivo en una imagen png.

- ***TABLASIGUIENTES:***

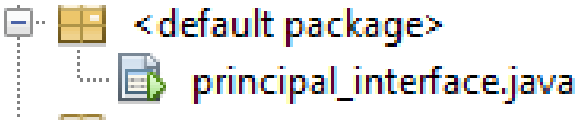
  Como su nombre lo indica esta clase organiza la información correspondiente a la tabla de siguientes para cada hoja presente en el árbol sintáctico.

Cuenta con métodos como printTable, GenerarDot y generarpng que tienen el mismo propósito que en el anterior inciso.

- ***TABLATRANSICIONES:***

  Como su nombre lo indica esta clase se encarga de formar una tabla de transiciones para el método del árbol. Posee cuatro métodos para realizar las gráficas, el método impTable y impAFD generan una cadena que diseña la tabla de transiciones y el autómata finito para la expresión trabajada, mientras que GenerarDot y generarpng cumplen la misma función que en lo anterior descrito.

DEFAULT PACKAGE



Contiene una única clase que modela la interfaz y las funciones que esta realiza, los métodos utilizados se explican a continuación:

- ***CARPETAS:***

  Este método se encarga de actualizar el contenido existente en todas las carpetas creadas para almacenar los reportes, es decir los archivos dot y las imágenes. La cadena con los nombres de los archivos se escribe en los Jtextarea de la interfaz.

- ***MOSTRARARCHIVOS:***

  Este método es llamado por el método carpetas y requiere como parámetro una ruta. Este método realizará una lista de los archivos contenidos en esa ruta dada y por medio de un ciclo los convertirá en una cadena que será retornada.

- ***BORRARIMAGEN:***

  Este método se encarga de borrar todo lo contenido en la lista global de imágenes que es utilizada en el visualizador de imágenes.

- ***CARGARIMAGEN:***

  Este método requiere una lista con la dirección de las imágenes que se quieren visualizar, estas imágenes se convertirán a imágenes tipo ico y se agregarán a la lista de imágenes.

- ***GEN_ARBOL:***

  Este método se encarga de llamar a los métodos de las clases del paquete árbol_afd para generar las graficas correspondientes a cada expresión regular definida en la entrada.

- ***ANALISISSIN:***

  Este método es el encargado de analizar la cadena de entrada en la expresión definida, para esto se hace uso de la librería regedit de java. En caso haya errores hace uso de la una lista de errores utilizando la clase excepcion en el paquete de errores para indicar en que parte no cumple la cadena con la expresión regular evaluada.

- ***GENERARJSON:***

  Este requiere una cadena y genera un archivo .json en una dirección indicada.

- ***GENERARHTML:***

  Este método genera una cadena con código html que diseña la pagina de reporte de errores, para ello hace una comparación en donde se indica que se ejecute este método cada vez que la lista de errores no este vacía y luego crea el archivo .html en la ruta indicada.

- ***T_CERRARMOUSECLICKED:***

  Este método ejecuta el comando para detener la ejecución del programa.

- ***T_MINMOUSECLICKED:***

  Este método ejecuta el comando para minimizar el frame.

- ***MENUMOUSEPRESSED Y MENUMOUSEDRAGGED:***

  Estos métodos son los encargados de hacer posible que la ventana o frame de la interfaz se mueva en la pantalla al presionar la barra de menú y arrastrarla. De esta forma la ventana no se queda estática.

- ***BOT_NUEVOMOUSECLICKED:***

  Este evento se encarga de borrar el texto contenido en el jtextarea que corresponde al área de texto.

- ***BOT_ABRIRMOUSECLICKED:***

  Este evento hace uso de la librería JFileChooser para abrir una ventana de navegación de archivos y poder seleccionar el archivo de texto que se requiere.

Se le pasa un parámetro para que solo abra archivos .olc y la dirección de este archivo se guarda en una variable global para uso en otros métodos.

- ***BOT_GUARDARMOUSECLICKED:***

Este evento toma la ruta global guardada y sobrescribe en dicho archivo el texto del jtextarea. En caso de que la ruta no contenga ninguna dirección se llama al método de Bot_Guardar_C.

- ***BOT_GUARDAR_CMOUSECLICKED:***

Hace uso de la librería JFileChooser de tipo SAVE_DIALOG el cual abre un explorador de archivos para guardar archivos con el mismo funcionamiento que en otros programas de nuestro equipo.

- ***BOT_ANALIZARMOUSECLICKED:***

Este evento tomara la lista de cadenas que se quieren evaluar y la lista de expresiones regulares definidas. Buscará que la expresión regular en la que se requiere el análisis exista y en caso contrario creará un JOptionPane con el mensaje para solicitar la creación de este.

Al existir se mandarán las listas de información necesaria al método de AnalisisSin explicado anteriormente y luego se creará una cadena con el formato que suelen tener los archivos json con los resultados del análisis y llamar al método correspondiente para crear dicho archivo.

- ***BOT_GENERARMOUSECLICKED:***

Este evento ejecuta a los analizadores de jflex y cup para validar el archivo de entrada y se pasara la información obtenida de estas a listas globales para usar en los otros métodos. Si la lista de errores no se encuentra vacía se llama al método de generarHTML para reportarlo y en caso contrario se mandará a llamar al método de gen_arbol para crear los árboles, tablas de siguientes, transiciones, AFD y AFND.

- ***BOT_ANTERIORMOUSECLICKED:***

Esta acción hará que se busque en la lista de imágenes el índice anterior al de la imagen actual que se visualiza y la inserta al Jtextpane que corresponde al visualizador.

- ***BOT_SIGUIENTEMOUSECLICKED:***

  Realiza la acción contraria a la anterior, es decir que buscara en la lista de imágenes el siguiente índice al de la imagen actual en el visualizador y luego la insertara.

- ***BOX_OPCIONESACTIONPERFORMED:***

  Este evento pertenece al menú desplegable o combo box y dependiendo el item seleccionado buscara en la carpeta correspondiente a esos reportes y creara una lista de imágenes por medio del método cargarimagen antes mencionado.

- ***METODOS MOUSEENTERED:***

  Todo método de este tipo contiene el código necesario para que el jlabel, jbotton,etc, que contenga este evento en la interfaz, cambie el color de fondo y texto cada vez que el mouse pase sobre él.

- ***METODOS MOUSEEXITED:***

  Este método contiene el código necesario para que el jlabel, jbotton,etc, que contenga este evento en la interfaz, cambie el color de fondo y texto al que tenia originalmente cuando el mouse no este sobre él.