

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERIA
INGENIERIA EN CIENCIAS Y SISTEMAS

Manual Técnico

FERNANDO AUGUSTO ARMIRA RAMÍREZ - 201503961

Manual

En el siguiente manual se dará a conocer más a detalladamente las características y el desarrollo del software REGEXIVE.

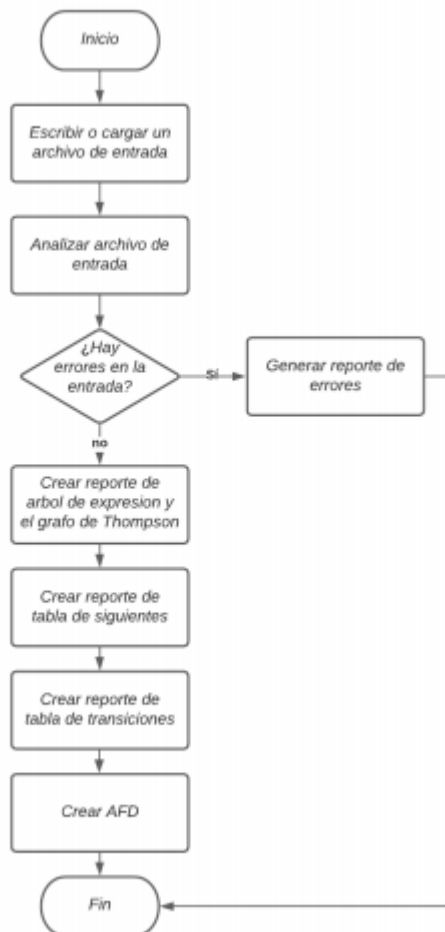
I. Descripción

A. Flujo del programa

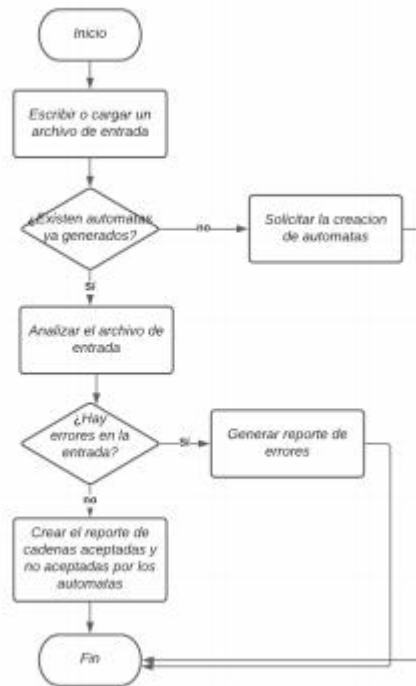
REGEXIVE es un intérprete que a través de un archivo de entrada utilizando los métodos del árbol y Thompson respectivamente genera el autómata finito determinista y no determinista de una expresión regular, posteriormente se podrán evaluar cadenas utilizando las expresiones regulares anteriormente definidas en el lenguaje.

Cabe mencionar que las expresiones regulares tienen que estar en notación polaca o prefija.

A continuación, el flujo del programa para la generación de los autómatas.



Flujo del programa para la evaluación de cadenas.



El programa en conclusión es un analizador léxico y sintáctico.

REGEXIVE fue desarrollado en el lenguaje de programación Java, mas específicamente se utilizaron las librerías Flex y Cup para la realización del analizador léxico y sintáctico, para la visualización gráfica de los distintos reportes se utilizó graphviz.



A continuación, un ejemplo de la estructura del lenguaje de los archivos de entrada OLC.

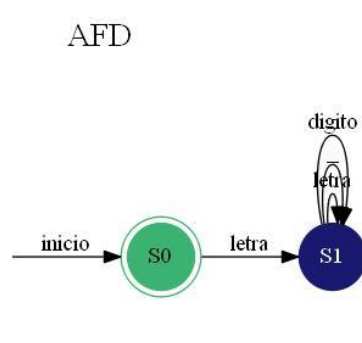
```
{
////// CONJUNTOS
CONJ: nombre_conjunto -> notacion; //la notación se define en la sección 4.1.2
CONJ: nombre_conjunto -> notacion;
tld -> Expresión_regular_en_prefijo;
tld -> Expresión_regular_en_prefijo;
// Mas sentencias
%%
%%
tld: "Lexema de entrada";
tld : "Otro Lexema";
// Mas sentencias
}
```

II. Análisis léxico

Como se mencionó anteriormente se utilizó la herramienta JFLEX.

De cada expresión regular definida se genera el AFD, con sus respectivos reportes, el cual nos permitirá validar cadenas.

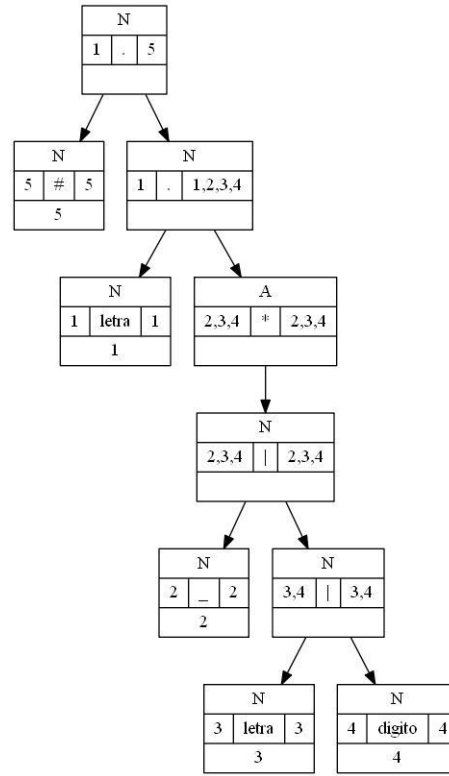
A. Autómata finito determinista



B. Reportes

A continuación, los reportes generados utilizados por el método del árbol.

- Árbol de la expresión:



- Tabla de siguientes:

Hoja	Numero	Siguientes
_	2	2,3,4,5
letra	3	2,3,4,5
digito	4	2,3,4,5
letra	1	2,3,4,5
#	5	-

- Tabla de transiciones:

Estado	letra	_	digito
S0	S1	-	-
S1	S1	S1	S1

III. Análisis sintáctico

El programa cuenta con análisis sintáctico. Se utilizo la herramienta Cup para poder generar

A continuación, tenemos la gramática del lenguaje utilizado.:

INICIO :{ BLOQUE1 BLOQUE2 BLOQUE3 }

BLOQUE1: DEFCONJ DEFEXPR

BLOQUE2: BLOQUE2 %%
|%%

BLOQUE3: DEFCADENA

DEFCONJ:DEFCONJ Conj : Id -> NOTACION ;
|Conj : Id -> NOTACION ;

NOTACION: letra ~ letra ;
| numero ~ numero;
| simbolo ~ simbolo;
| letra NOTACION2;
| numero NOTACION2;
| simbolo NOTACION2;

NOTACION2:| NOTACION2 , letra
| NOTACION2 , numero
| NOTACION2 , simbolo
| , letra
| , numero
| , simbolo

DEFEXPR:DEFEXPR Id -> EXPRESION;
|Id -> EXPRESION;

EXPRESION: . EXPRESION EXPRESION
| | EXPRESION EXPRESION
| * EXPRESION

| + EXPRESION
| ? EXPRESION
| { Id }
| { Cadena }

DEFCADENA:DEFCADENA Id : Cadena;
|Id : Cadena;

Para el manejo y recuperación de errores tanto léxicos como sintácticos se utilizó el método pánico.