Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Lenguajes formales y de programación

Primer semestre, 2022.

MANUAL TÉCNICO

Proyecto 2

Diego Andrés Huite Alvarez

202003585

28/04/2022

Contenido

INTRODUCCIÓN

El presente manual tiene como el explicar como funciona el analizador léxico y sintáctico elaborados para el proyecto 2 de LFP.

REQUERIMIENTOS

Para el correcto funcionamiento del software usted necesita del siguiente hardware:

- Mouse
- Teclado
- Memoria RAM (4gb mínimo)

Y en cuanto a software usted necesitará:

- Navegador
- Cualquier versión de Python superior a la 3.0.0
- Cualquier IDE para python.
- Sistema operativo tipo linux o windows

LÓGICA DEL PROGRAMA

- **Análisis previo del lenguaje**

Para comenzar el análisis del siguiente lenguaje:

```
{RESULTADO} "equipo1" {VS} "equipo2" {TEMPORADA} <numero-numero>
{JORNADA} numero {TEMPORADA} <numero-numero> [ -f nombre_archivo ]
{GOLES} condición "equipo" TEMPORADA <numero-numero>
{TABLA} {TEMPORADA} <numero-numero> [ -f nombre_archivo ]
{PARTIDOS} "equipo" {TEMPORADA} <numero-numero> [ -f nombre_archivo ] [ -ji numero ] [ -jf numero ]
{TOP} condición {TEMPORADA} <numero-numero> [ -n numero ]
{ADIOS}
```

Este lenguaje consta de 7 comandos, las palabras reservadas están dentro de las llaves, pero estas llaves no forman parte del lenguaje. Lo que se encuentra dentro de los corchetes puede venir o no venir, y los corchetes tampoco forman parte del lenguaje debemos de llevar a cabo un proceso que se llama “tokenización” que consiste en hallar los tokens del lenguaje. Para este caso en específico se llegó a la siguiente tabla de tokens:

	LISTA DE TOKENS	
Descripción	**Patrón**	**Regex**
nombre equipo	cualquier cadena entre un par de comillas dobles	"[^]"
guion	Caracter -	-
numero	digito unitario o más	d+
signo mayor	Caracter >	>
signo menor	Caracter <	<
nombre archivo	Letra o guion bajo con letras que pueden tener numeros entre ellos y guiones ba	[A-Za-z][_?0-9A-Za-z_?]*
palabra reservada: f	Una letra f	f
palabra reservada: ji	j seguida de una i	fi
palabra reservada: jf	j seguido de una f	jf
palabra reservada: n	una letra n	n
palabra reservada: RESULTADO	RESULTADO	RESULTADO
palabra reservada: VS	VS	VS
palabra reservada: TEMPORADA	TEMPORADA	TEMPORADA
palabra reservada: JORNADA	JORNADA	JORNADA
palabra reservada: GOLES	GOLES	GOLES
palabra reservada: LOCAL	LOCAL	LOCAL
palabra reservada: VISITANTE	VISITANTE	VISITANTE
palabra reservada: TOTAL	TOTAL	TOTAL
palabra reservada: TABLA	TABLA	TABLA
palabra reservada: PARTIDOS	PARTIDOS	PARTIDOS
palabra reservada: TOP	TOP	TOP
palabra reservada: SUPERIOR	SUPERIOR	SUPERIOR
palabra reservada: INFERIOR	INFERIOR	INFERIOR
palabra reservada: ADIOS	ADIOS	ADIOS
	REGEX: "[^"]*" | d+ | S | L (w | d)*	L = [A-Za-z]
		w = {_, L}
		S = {<,>,-}

Una vez teniendo la tabla de tokens podemos generar la siguiente expresión regular:

\ "[^"]*" | d+ | S | L (w | d)*

donde

S = {<,>,-}

L = [A-Za-z]

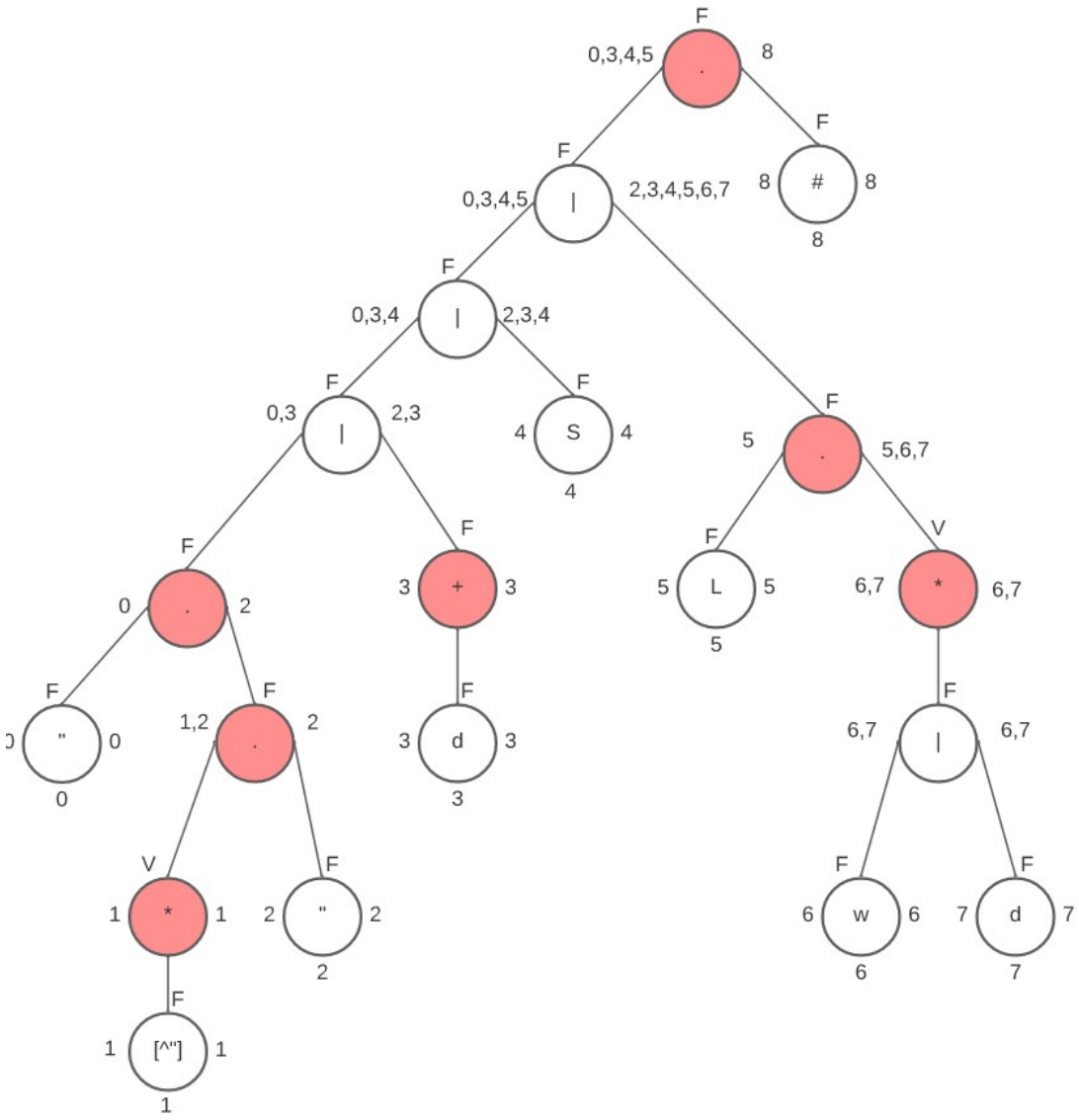
w = {_,L}

Generación de AFD

Para generar el AFD se debe de implementar un símbolo de finalización a la expresión regular. Siendo este símbolo “$”. La nueva expresión regular luce así:

\("[^"]*" | d+ | S | L (w | d)*)#

Se procede a generar el árbol binario para esta expresión:



Teniendo el árbol podemos generar la tabla de siguientes.

Tabla de siguientes:

Hoja	Terminal	Siguientes
0	“	1,2
1	“[^”]”	1,2
2	“	8
3	d	3,8
4	S	8
5	L	6,7,8
6	w	6,7,8
7	d	6,7,8
8	#	-

Generamos las transiciones posibles:

s0 = {0,3,4,5}

sig(0)=sig(")={1,2} -> s1

sig(3)=sig(d)={3,8} -> s2*

sig(4)=sig(S)={8} -> s3*

sig(5)=sig(L)={6,7,8} -> s4*

s1 = {1,2}

sig(1)=sig([^"])={1,2} -> s1

sig(2)=sig(")={8} -> s3*

s2* = {3,8}

sig(3)=sig(d)={3,8} -> s2*

s3* = {8}

s4* = {6,7,8}

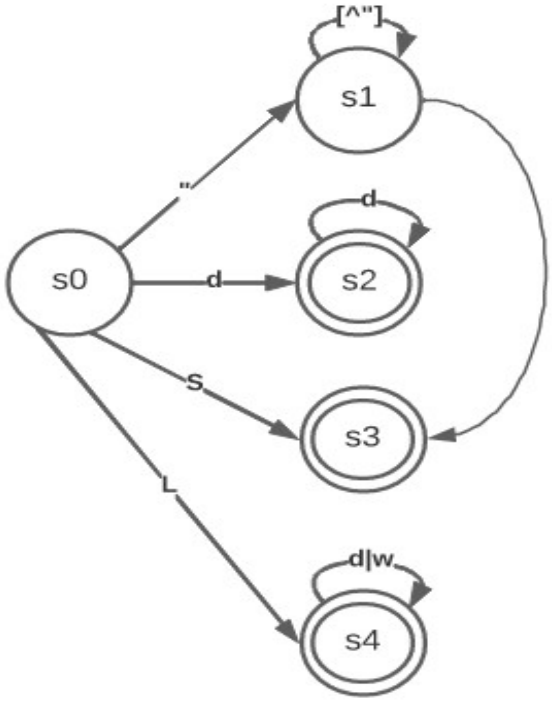
sig(6)=sig(w)={6,7,8} -> s4*

sig(7)=sig(d)={6,7,8} -> s4*

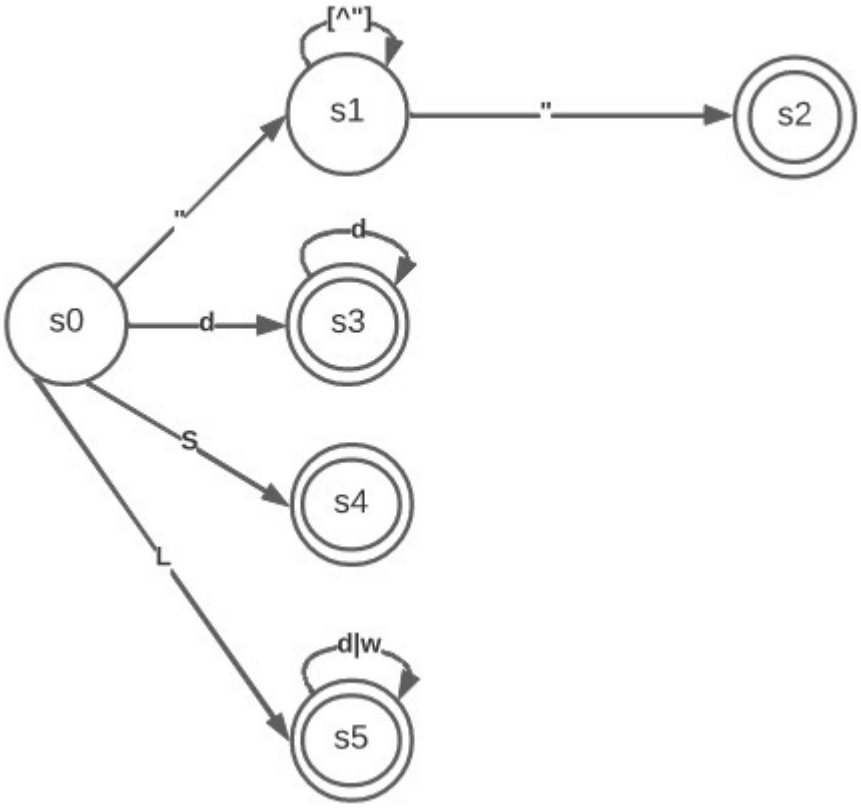
Tabla de transiciones

Estados/ terminales	“	[^”]	d	S	w	L
s0	s1	-	s2	s3	-	s4
s1	s3	s1	-	-	-	-
s2*	-	-	s2	-	-	-
s3*	-	-	-	-	-	-
s4*	-	-	s4	-	s4	-

Se procede a dibujar el AFD



Por motivos prácticos se amplió el AFD de la siguiente manera:



Este fue el automata utilizado para la codificación del programa

ANÁLISIS SINTÁCTICO

Gramática de tipo 2 utilizada para elaborar el análisis sintáctico del proyecto:

```
S :: = INICIO

INICIO ::= RESULTADO|JORNADA|GOLES|TABLA|PARTIDOS|TOP|ADIOS

RESULTADO ::= reservada_resultado cadena reservada_VS cadena reservada_temporada menorQUE numero guion numero mayorQUE

JORNADA ::= reservada_jornada numero TEMPORADA menorQUE numero guion numero mayorQUE BANDERA1

BANDERA1 ::= guion f name_archivo|epsilon

GOLES ::= reservada_goles CONDICION_GOLES cadena reservada_temporada menorQUE numero guion numero mayorQUE

CONDICION_GOLES ::= reservada_local| reservada_visitante|reservada_total

TABLA ::= reservada_tabla reservada_temporada menor_QUE numero guion numeroero4_digitos mayorQUE BANDERA1

PARTIDOS ::= reservada_partidos cadena reservada_temporada menorQUE numeroero4_digitos guion numero mayorQUE LISTA_BANDERAS

BANDERA2 ::= guion j i numero|epsilon

BANDERA3 ::= guion j f numero|epsilon

BANDERA4 ::= guion n numero|epsilon

LISTA_BANDERAS ::= BANDERA1|BANDERA2|BANDERA3 LISTA_BANDERAS_

LISTA_BANDERAS_ ::= BANDERA1|BANDERA2|BANDERA3 LISTA_BANDERAS_

LISTA_BANDERAS_ ::= epsilon

TOP ::= reservada_top CONDICION_TOP reservada_temporada menorQUE numero guion numero mayorQUE BANDERA4

CONDICION_TOP ::= reservada_superior|reservada_inferior

BANDERA3 ::= guion n numero|epsilon

ADIOS ::= reservada_adios
```