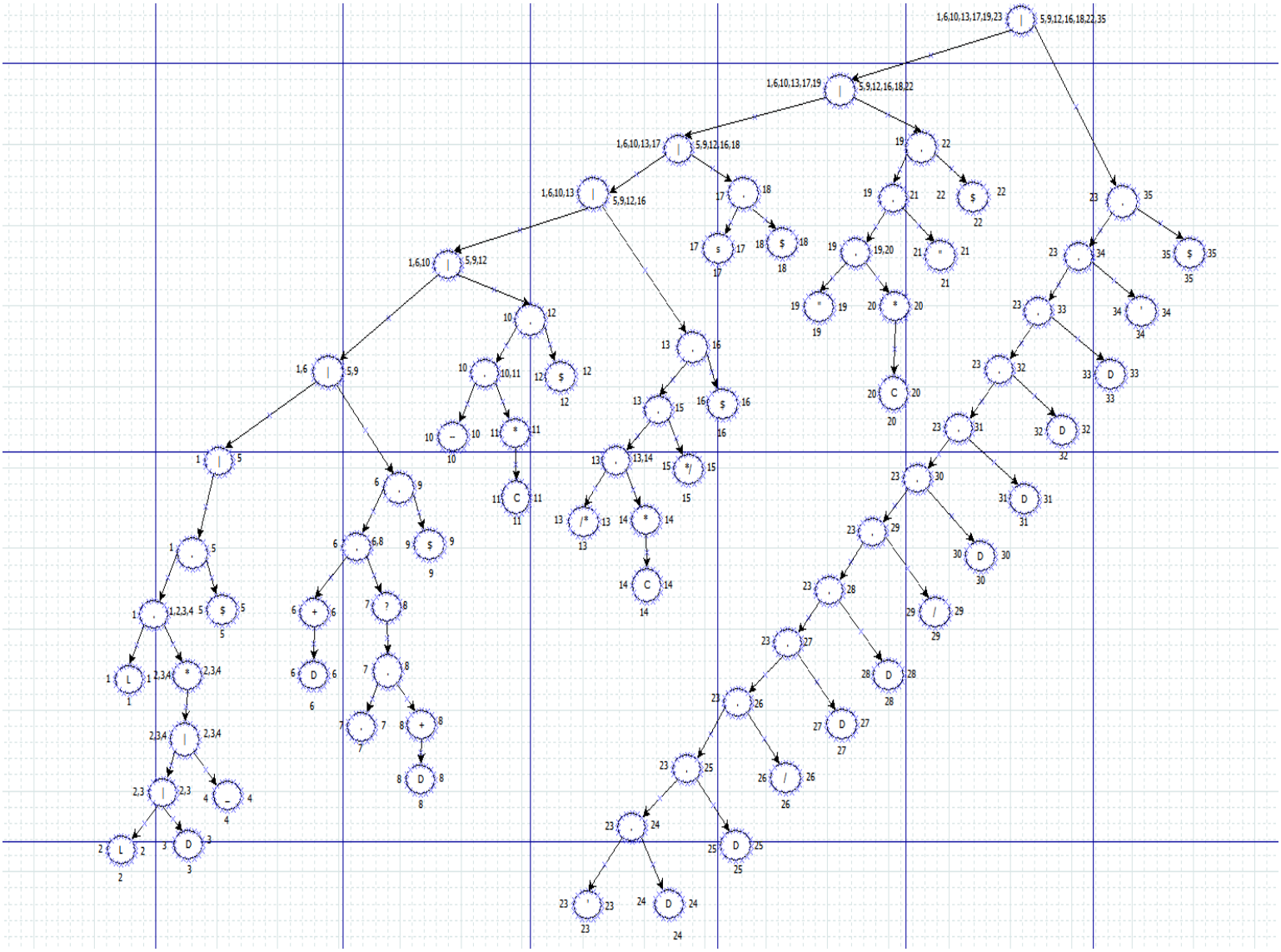
Manual técnico

Método del árbol



Tablas siguientes

simbolo	i	siguiente(i)	simbolo	i	siguiente(i)
L	1	2,3,4,5	"	21	22
L	2	2,3,4,5	$	22	----------
D	3	2,3,4,5	'	23	24
_	4	2,3,4,5	D	24	25
$	5	----------	D	25	26
D	6	6,7,9	/	26	27
.	7	8	D	27	28
D	8	8,9	D	28	29
$	9	----------	/	29	30
--	10	11,12	D	30	31
c	11	11,12	D	31	32
$	12	----------	D	32	33
/*	13	14,15	D	33	34
c	14	14,15	'	34	35
*/	15	16	$	35	----------
$	16	----------			
s	17	18			
$	18	----------			
"	19	20,21			
c	20	20,21			

Transición de estados

N	sig	estados	TR	Aceptacion
s0		1,6,10,13,17,19,23		
	sig(1)	2,3,4,5	s1	
	sig(6)	6,7,9	s2	
	sig(10)	11,12	s3	
	sig(13)	14,15	s4	
	sig(17)	18	s5	
	sig(19)	20,21	s6	
	sig(23)	24	s7	
s1		2,3,4,5		
	sig(2)	2,3,4,5	s1	
	sig(3)	2,3,4,5	s1	
	sig(4)	2,3,4,5	s1	

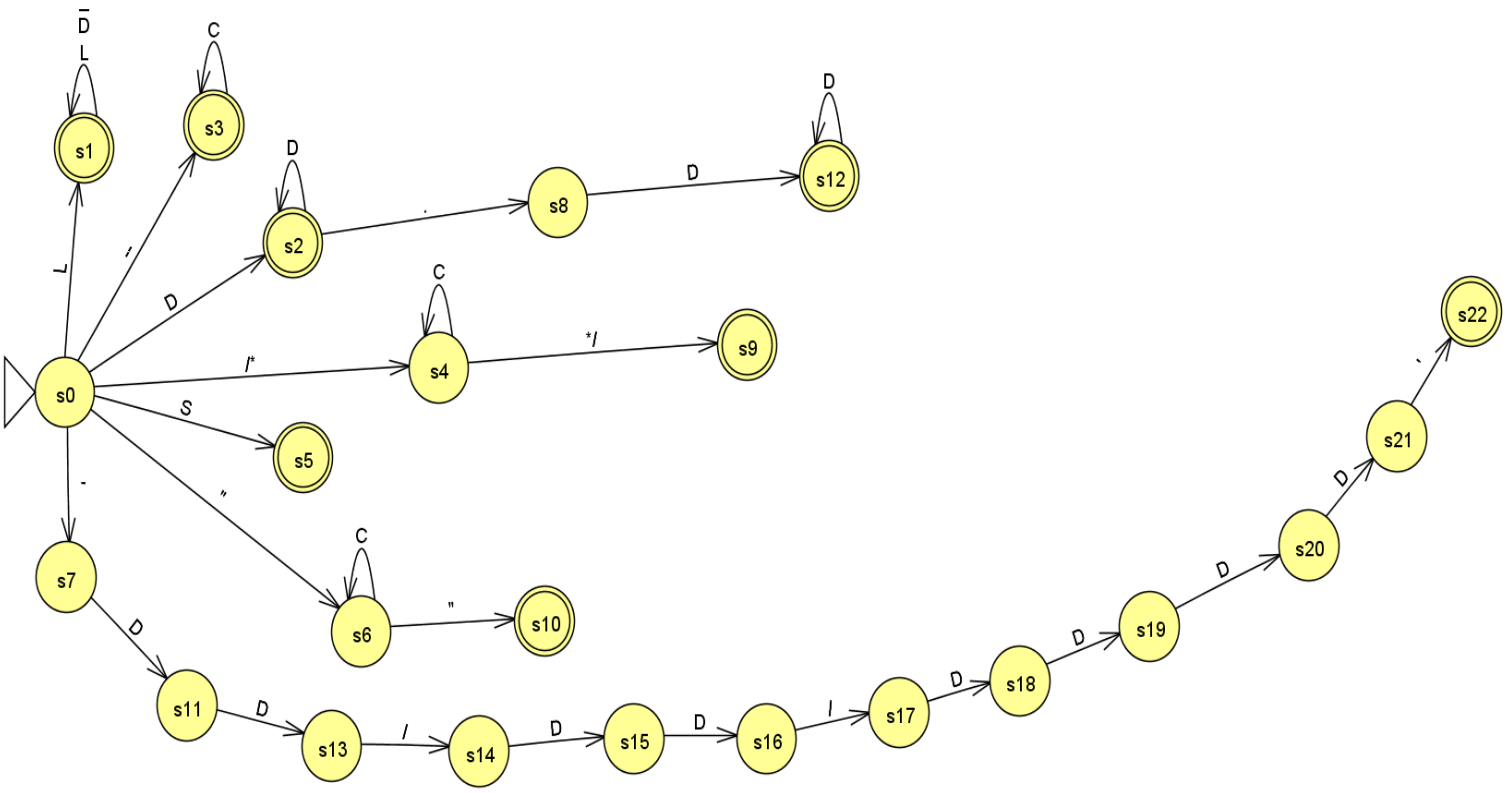
	sig(5)	*****	*****	Aceptacion
s2		6,7,9		
	sig(6)	6,7,9	s2	
	sig(7)	8	s8	
	sig(9)	*****	*****	Aceptacion
s3		11,12		
	sig(11)	11,12	s3	
	sig(12)	*****	*****	Aceptacion
s4		14,15		
	sig(14)	14,15	s4	
	sig(15)	16	s9	Aceptacion
s5		18		
	sig(18)	*****	*****	Aceptacion
s6		20,21		
	sig(20)	20,21	s6	
	sig(21)	22	s10	Aceptacion
s7		24		
	sig(24)	25	s11	
s8		8		
	sig(8)	8,9	s12	
s11		25		
	sig(25)	26	s13	
s12		8,9		
	sig(8)	8,9	s12	
	sig(9)	*****	*****	Aceptacion
s13		26		
	sig(26)	27	s14	
s14		27		
	sig(27)	28	s15	
s15		28		
	sig(28)	29	s16	
s16		29		
	sig(29)	30	s17	
s17		30		
	sig(30)	31	s18	
s18		31		
	sig(31)	32	s19	
s19		32		
	sig(32)	33	s20	
s20		33		

	sig(33)	34		s21	
s21		34			
	sig(34)	35		s22	aceptacion

Tabla de transiciones

Estados	**L**	**D**	**c**	**--**	**.**	**/***	***/**	**s**	**"**	**_**	**'**	**/**	**AC.**
s0	s1	s2		s3		s4		s5	s6		s7		
s1	s1	s1								s1			A
s2		s2			s8								A
s3			s3										A
s4			s4				s9						
s5													A
s6			s6						s10				
s7		s11											
s8		s12											
s9													A
s10													A
s11		s13											
s12		s12											A
s13												s14	
s14		s15											
s15		s16											
s16												s17	
s17		s18											
s18		s19											
s19		s20											
s20		s21											
s21											s22		
s22													A

Autómata finito Determinista



Gramática

S: sentencias

sentencias: sentencia sentenciaP

sentenciaP: sentencia sentenciaP

| Epsilon

sentencia: Create

| Select

| insert

| delete

| update

Create: CREAR TABLA ID (parametros);

parametros: parametro parametrosP

parametrosP: , parametro parametrosP

| Epsilon

parametro: ID tipo

tipo: Entero

| Cadena

| Flotante

| Fecha

insert: INSERTAR EN ID VALORES(valores);

valores: valor valoresP

valoresP: , valor valoresP

|Epsilon

valor: NUMERO

| CADENA

| FECHA

| DECIMAL

Select: SELECCIONAR campos DE tablas where ;

campos: campo camposP

camposP: , campo camposP

| Epsilon

campo: ID campoP

| *

campoP-> . campoB

| como

campoB -> ID campoBprima

| *

campoBprima -> como

|Epsilon

como: COMO ID

| Epsilon

tablas: ID tablaP

tablaP: , ID tablaP

| Epsilon

where: DONDE condiciones

| Epsilon

condiciones: condicion condicionesP

condicionesP: Y condicion condicionesP

| O condicion condicionesP

| Epsilon

condicion: ID condicionP

condicionP: simbolo valorP

| .ID simbolo valorP

valorP: valor

| ID.ID

simbolo: =

| =>

| =<

| !=

| <

| >

delete: ELIMINAR DE ID where;

update: ACTUALIZAR ID ESTABLECER (asignaciones) where;

asignaciones: asignacion asignacionesP

asignacionesP: , asignacion asignacionesP

| Epsilon

asignacion: ID = valor

Diccionario de clases

Form1:

Clase donde esta toda la parte funcional grafica.

Analizador

AnalizadorLexico: esta clase es la encargada del funcionamiento del analizador léxico el cual lee carácter por carácter y este las recibe y verifica si pertenece al lenguaje o no.

AnalizadorSintactico: esta clase es la encargada del funcionamiento del analizador sintáctico el cual recibe la lista de tokens y verifica que el orden este correctamente, también generamos las tablas, consultas y el árbol de derivación.

Nodo: esta clase es la encargada de almacenar los nodos del árbol de derivación.

Clases:

Columna: clase encargada de almacenar los índices de las tablas.

Errores: clase encarga de almacenar los errores léxicos y sintácticos.

Tabla: clase la cual se almacena la tabla creada.

Tipo: clase tipo enum el cual tiene todos los tipos de token.

Token: clase encargada de almacenar los datos de los tokens reconocidos.