Apéndice A Documentación Lexer

Expresiones Regulares

Token	Expresión Regular
ELSE	‘else’
IF	‘if’
INT	‘int’
RETURN	‘return’
VOID	‘void’
WHILE	‘while’
PLUS	‘+’
MINUS	‘-’
TIMES	‘*’
DIVIDE	‘/’
EQUALS	‘=’
LT	‘<’
LE	‘<=’
GT	‘>’
GE	‘>=’
EQ	‘==’
NE	‘!=’
SEMICOLON	‘;’
COMMA	‘,’
LPAREN	‘(‘
RPAREN	‘)’
LBRACKET	‘[‘
RBRACKET	‘]’
LKEY	‘{‘

RKEY	‘}’
ID	[a-zA-Z][a-zA-Z]*
NUM	[0-9][0-9]*
COMMENT	/’*’(/* | ‘*’* [~’*’/])’*’* ‘*’/
ENDFILE	‘$’

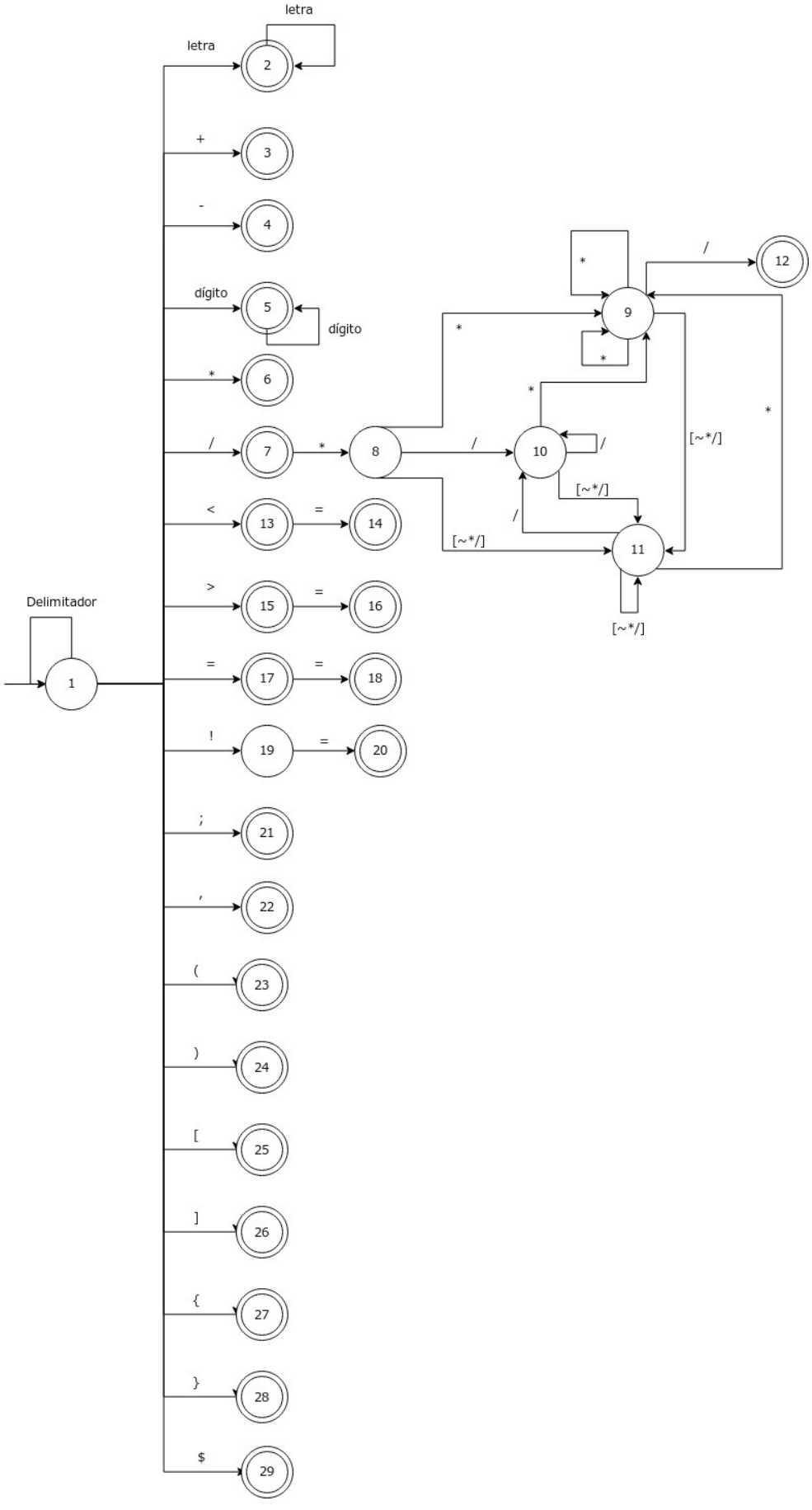
DFA

Tokens por estado final de DFA

Estado	Token
2	ID o palabra reservada (ELSE, IF, INT, RETURN, VOID, WHILE)
3	PLUS
4	MINUS
5	NUM
6	TIMES
7	DIVIDE
12	COMMENT
13	LT
14	LE
15	GT
16	GE
17	EQUALS
18	EQ
20	NE
21	SEMICOLON
22	COMMA

23	LPAREN
24	RPAREN
25	LBRACKET
26	RBRACKET
27	LKEY
28	RKEY
29	ENDFILE

DFA


letra
letra
2
+
3
-
4
dígito
5
dígito
*
6
/
7
*
8
/
10
/
[~*/]
*
9
*
*
/
12
*
<
13
=
14
[~*/]
11
>
15
=
16
Delimitador
1
=
17
=
18
!
19
=
20
[~*/]
;
21
,
22
(
23
)
24
[
25
]
26
{
27
}
28
$
29

Recuperación de Errores

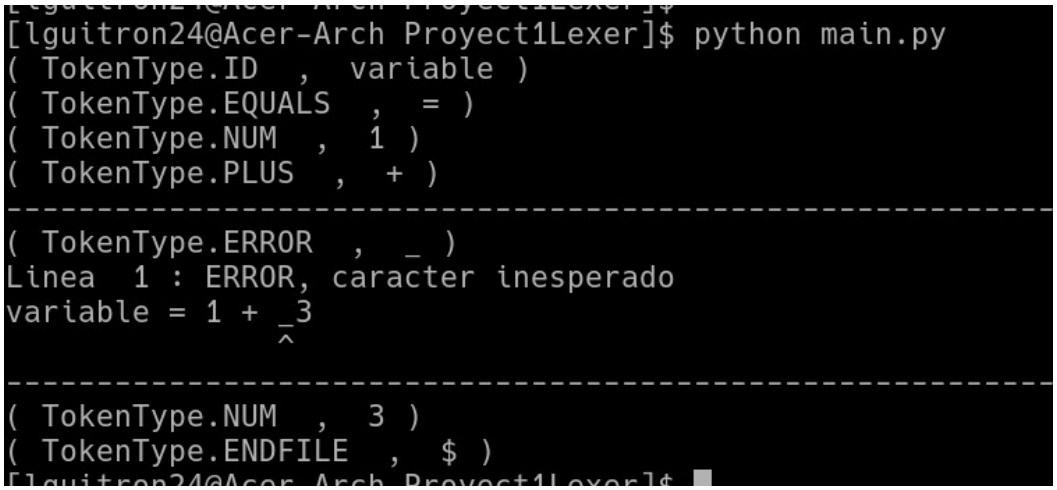
En esta implementación el lexer revisa únicamente la cadena actual sin revisar los caracteres que vienen a después. Por la razón anterior, los errores identificados por el lexer corresponderán a caracteres invalidos.

Ejemplos:

Cadena	Salida Lexer	Notas
contador = contador + 3indice	(TokenType.ID, contador) (TokenType.EQUALS, =) (TokenType.ID, contador) (TokenType.PLUS, +) (TokenType.NUM, 3) (TokenType.ID, indice)	A pesar de que la cadena no sea válida en el lenguaje C-, el lexer identificará únicamente Tokens válidos, por lo que la identificación de este tipo de error será realizada por el parser.
variable = 1 + _3	(TokenType.ID, variable) (TokenType.EQUALS, =) (TokenType.NUM, 1) (TokenType.PLUS, +) (TokenType.ERROR, _) (TokenType.NUM, 3)	El caracter '_' no está definido, por lo que es detectado como un caracter ilegal por el lexer.

Al detectar un caracter inválido se imprimirá un mensaje de error indicando la línea y la posición de este caracter. El lexer continuará detectando tokens de forma normal inmediatamente después de la expresión inválida.

La impresión en consola de la línea
variable = 1 + _3
es la siguiente:


[lguitron24@Acer-Arch Proyect1Lexer]$ python main.py
(TokenType.ID , variable)
(TokenType.EQUALS , =)
(TokenType.NUM , 1)
(TokenType.PLUS , +)
(TokenType.ERROR , _)
Linea 1 : ERROR, caracter inesperado
variable = 1 + _3
(TokenType.NUM , 3)
(TokenType.ENDFILE , $)