

Trabalho de Compiladores - Linguagem L

Nome:

Luiz Gustavo Bragança dos Santos

Ricardo Xavier Sena

Rithie Natan Carvalhaes Prado

Analizador Léxico

Observação:

L = a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, x, w, y, z, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, X, W, Y, Z

d = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

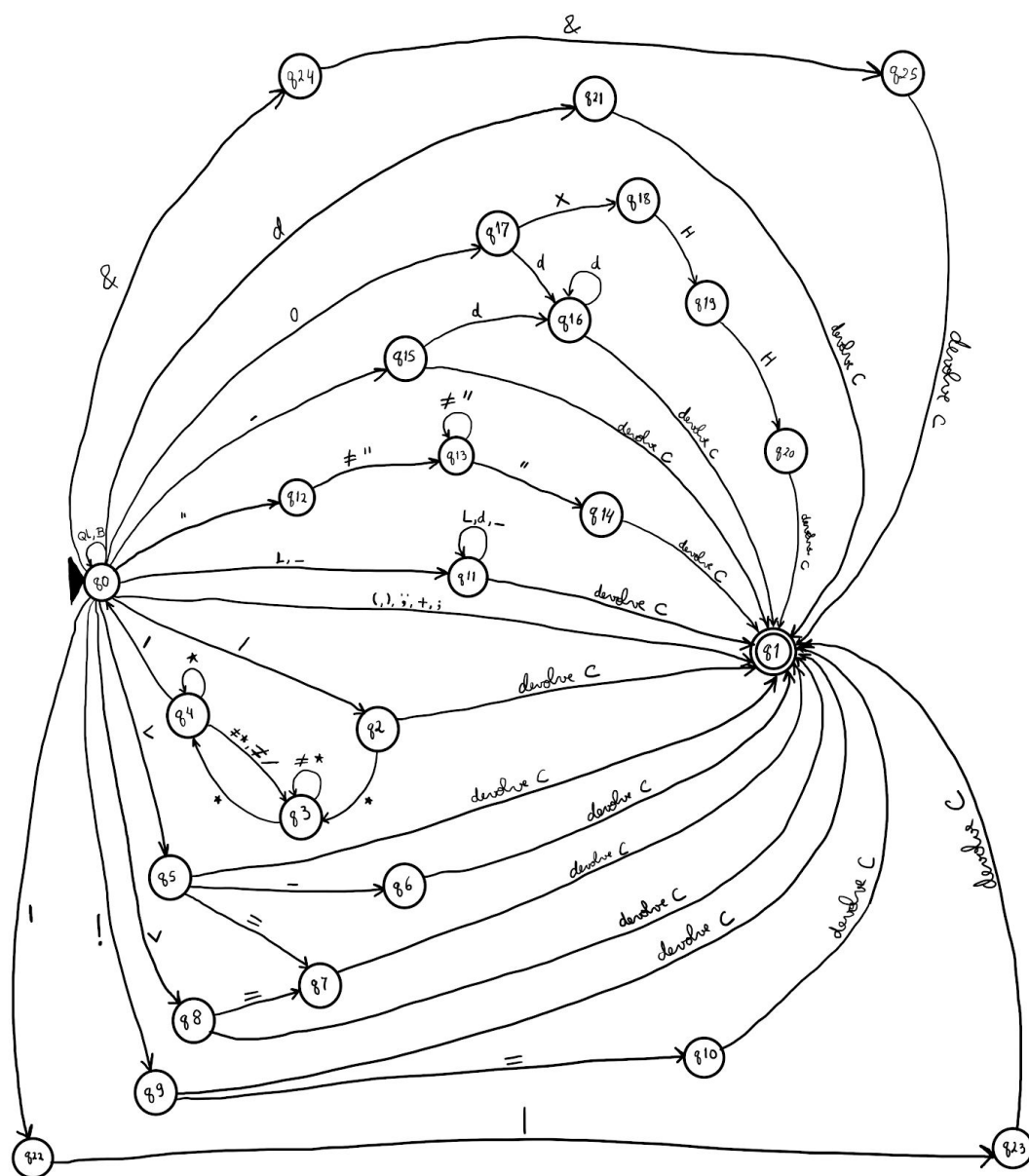
Este 'H' é exclusivo para tratar somente os caracteres do hexadecimal.

H = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Númeração	Σ	Lexema
0	id	$(L _) (L d _)^*$
1	const	$(\text{' ' ' ('L' 'd' ' ' ' ' ':' ',' ';' '&' ':' '(' ')'} '[' ']' '{' '}' '+' '-' ' ' ' ' '!' '?' '>' '<' '=' '\'$
2	final	final
3	int	int
4	byte	byte
5	string	string
6	while	while
7	if	if
8	else	else
9	and	&&
10	or	
11	not	!

12	atribuição	<-
13	igual	=
14	abre_parentese	(
15	fecha_parentese)
16	menor	<
17	maior	>
18	diferente	!=
19	maior_igual	>=
20	menor_igual	<=
21	virgula	,
22	mais	+
23	menos	-
24	multiplicação	*
25	divisão	/
26	ponto_virgula	;
27	begin	begin
28	endwhile	endwhile
29	endif	endif
30	endelse	endelse
31	readln	readln
32	write	write
33	writeln	writeln
34	TRUE	TRUE
35	FALSE	FALSE
36	boolean	boolean

Autômato



Gramática

S → ARQUIVO

ARQUIVO → {DECLARAÇÃO; | COMANDO}*

DECLARAÇÃO → (int | byte | string | boolean) { id | ATRIBUIÇÃO {, id | ATRIBUIÇÃO}}+ |
final ATRIBUIÇÃO1;

COMANDO → ATRIBUIÇÃO; | REPETIÇÃO | TESTE | ; | LEITURA; | ESCRITA;

ATRIBUIÇÃO → id <- EXPRESSÃO

ATRIBUIÇÃO1 → id <- (TRUE | FALSE | const)

REPETIÇÃO → while('EXPRESSÃO') {begin {COMANDO}* endwhile | COMANDO}

TESTE → if('EXPRESSÃO') {begin {COMANDO}* endif | COMANDO} {else {begin
{COMANDO}* endif | COMANDO}}

LEITURA → readln('id\n')

ESCRITA → write('{EXPRESSÃO}') | writeln('{EXPRESSÃO}\n')

EXPRESSÃO → EXPRESSÃO1 [(> | >= | < | <= | = | !=) EXPRESSÃO1]*

EXPRESSÃO1 → [(+ | -)] EXPRESSÃO2 [(+ | < | '|') EXPRESSÃO2]*

EXPRESSÃO2 → VALOR [(* | / | '&&') VALOR]*

VALOR → id | const | EXPRESSÃO | !EXPRESSÃO | ('EXPRESSÃO') | TRUE | FALSE