

**Pontifícia Universidade Católica de Minas Gerais**

Bacharelado em Ciência da Computação

Fabio Silva Campos Melo

Gustavo Lescowicz Kotarsky

Lucas Dutra Ponce de Leon

Compilador Linguagem L

Parte 1 e 2

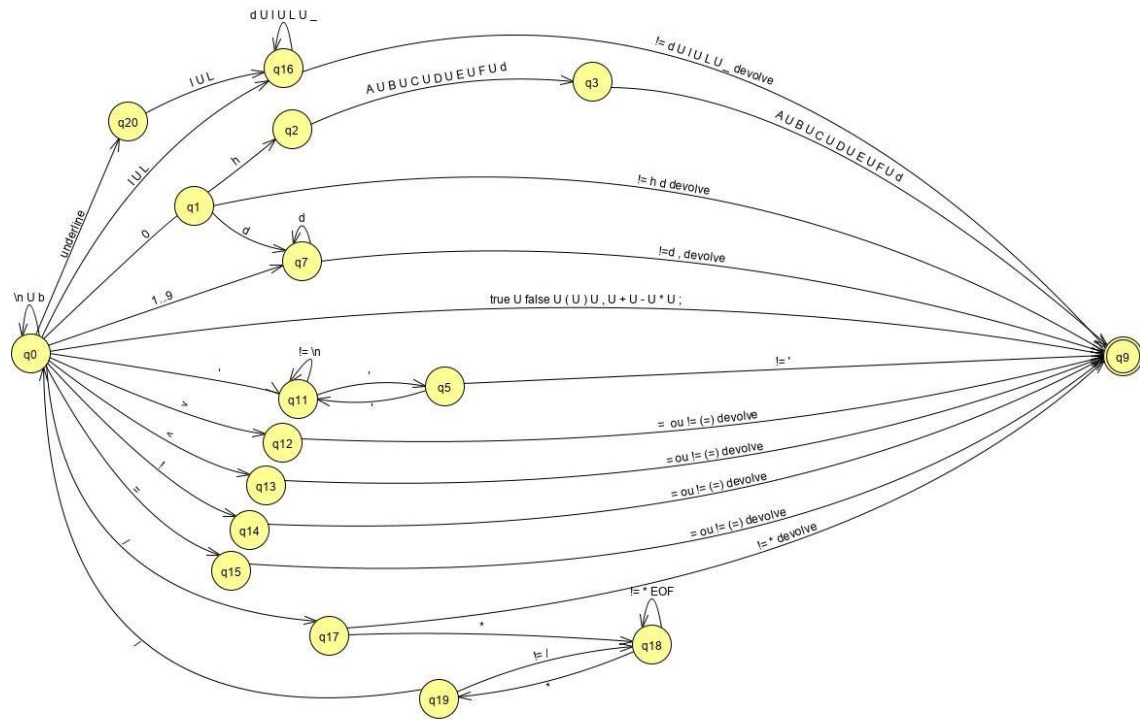
Belo Horizonte, Minas Gerais

2019

Alfabeto, padrão de formação dos lexemas:

Num	Token	Lex
1	<i>Byte</i>	0h(A U B U C U D U E U F)(A U B U C U D U E U F) U d
2	<i>Integer</i>	[-]d
3	<i>String</i>	'(!= \n)*'
4	<i>Boolean</i>	true U false
5	id	([_U _L U I U L]+ (d U L U I U _)*
6	while	while
7	if	if
8	else	else
9	and	and
10	or	or
11	not	not
12	=	=
13	==	==
14	(	(
15	)	)
16	<	<
17	>	>
18	!=	!=
19	>=	>=
20	<=	<=
21	,	,
22	+	+
23	-	-
24	*	*
25	/	/
26	;	;
27	begin	begin
28	end	end
29	then	then
30	readln	readln
31	main	main
32	write	write
33	writeln	writeln
34	true	true
35	false	false
36	boolean	boolean
37	string	string
38	const	const
39	byte	byte
40	integer	integer

Autômato:



Gramática:

**S** -> {**D**}+ main {**C**}+end

**D** -> (integer | boolean | byte | string) id [= [-]v\_const] {,id [= [-]v\_const]}; |  
const id = [-]v\_const;

**C** -> id = **EXP**; |  
while '(' **EXP** ')' **W** |  
if '(' **EXP** ')' then ( **C** [else (**W**) ] | begin {**C**} end [else (**W**) ] ) |  
; |  
readln(' id '); |  
(write | writeln)(' **EXP**{**EXP**} ');

**W** -> begin {**C**} end | **C**

**EXP** -> **EXPS** [(==|!=|<|>|<=|>=) **EXPS**]

**EXPS** -> [+|-] **T** {(+|-|or) **T**}

**T** -> **F** {( \* | and | / ) **F**}

**F** -> '(' **EXP** ')' | id | v\_const | not **F**