



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

Instituto de Ciências Exatas e de Informática

Departamento de Ciência da Computação

Disciplina: Compiladores

Compilador para a linguagem de programação L

*

Vinicius Francisco da Silva¹

*Trabalho apresentado para a disciplina de compiladores.

¹Aluno do Programa de Graduação em Ciência da Computação, Brasil – vinicius.silva.1046664@sga.pucminas.br.

1 ALFABETO Σ

- final
- else
- (
- <=
- ;
- write
- int
- &&
-)
- ,
- begin
- writeln
- byte
- ||
- <
- +
- endwhile
- TRUE
- string
- !
- >
- -
- endif
- FALSE
- while

- <=
- !=
- *
- endelse
- boolean
- if
- =
- >=
- /
- readln

2 LEXEMAS E PADRÃO DE FORMAÇÃO

3 ANALISADOR LÉXICO - AFD

4 GRAMÁTICA COM EXPRESSÕES REGULARES - GER

Tabela 1 – Lexema x Padrão de formação

Posição	Lexema	Padrão de Formação
1	final	$(f \cup F)(i \cup I)(n \cup N)(a \cup A)(l \cup L)$
2	else	$(e \cup E)(l \cup L)(s \cup S)(e \cup E)$
3	((
4	<=	<=
5	;	;
6	write	$(w \cup W)(r \cup R)(i \cup I)(t \cup T)(e \cup E)$
7	int	$(i \cup I)(n \cup N)(t \cup T)$
8	&&	&&
9))
10	,	,
11	begin	$(b \cup B)(e \cup E)(g \cup G)(i \cup I)(n \cup N)$
12	writeln	$(w \cup W)(r \cup R)(i \cup I)(t \cup T)(e \cup E)(l \cup L)(n \cup N)$
13	byte	$(b \cup B)(y \cup Y)(t \cup T)(e \cup E)$
14		
15	<	<
16	+	+
17	endwhile	$(e \cup E)(n \cup N)(d \cup D)(w \cup W)(h \cup H)(i \cup I)(l \cup L)(e \cup E)$
18	TRUE	TRUE
19	string	$(s \cup S)(t \cup T)(r \cup R)(i \cup I)(n \cup N)(g \cup G)$
20	!	!
21	>	>
22	-	-
23	endif	$(e \cup E)(n \cup N)(d \cup D)(i \cup I)(F \cup f)$
24	FALSE	FALSE
25	while	$(w \cup W)(h \cup H)(i \cup I)(l \cup L)(e \cup E)$
26	<-	<-
27	!=	!=
28	*	*
29	endelse	$(e \cup E)(n \cup N)(d \cup D)(e \cup E)(l \cup L)(s \cup S)(e \cup E)$
30	boolean	$(b \cup B)(o \cup O)(o \cup O)(l \cup L)(e \cup E)(a \cup A)(n \cup N)$
31	if	$(i \cup I)(F \cup f)$
32	=	=
33	>=	>=
34	/	/
35	readln	$(r \cup R)(e \cup E)(a \cup A)(d \cup D)(l \cup L)(n \cup N)$

Gramática com Expressões Regulares LL(1) da Linguagem L

S \rightarrow {**Declarar**}* {**Comando**}*

Declarar \rightarrow “final” “id” “<-” [“-”] const “;” | (int | boolean | byte | string) id
ListaIds “;”

ListaIds \rightarrow [**Atrib**] {“,” “id” [**Atrib**]}*

Atrib \rightarrow “<-” [“-”] const

Comando \rightarrow **Atribuicao** | **Repeticao** | **Teste** | **Nulo** | **Leitura** | **Escrita**

Atribuicao \rightarrow “id” “<-” **Expressao** “;”

Repeticao \rightarrow “while” (“**Expressao**”) **Blocowhile**

Blocowhile \rightarrow “begin” {**Comando**}* “endwhile” | **Comando**

Teste \rightarrow “if” (“**Expressao**”) (**Blocoif** | **Comando** [**BlocoElse**])

Blocoif \rightarrow “begin” {**Comando**}+ “endif” “else” “begin” {**Comando**}+ “endelse”

BlocoElse \rightarrow “else” **Comando**

Nulo \rightarrow “;”

Leitura \rightarrow “readln” (“id”) “;”

Escrita \rightarrow “write” (“**ListaExpressoes**”) “;” | “writeln” (“**ListaExpressoes**”) “;”

ListaExpressoes \rightarrow **Expressao** {“,” **Expressao**}*

Expressao \rightarrow **Exp** [(“=” | “!=” | “<” | “>” | “<=” | “>=”) **Exp**]

Exp \rightarrow [+ | -] **T** {(“+” | “-” | “||”) **T**}*

T \rightarrow **F** {(“*” | “&&” | “/”) **F**}*

F \rightarrow “!” **F** | (“**Expressao**”) | [“-”] const | id