

# Documentação Compilador L

Alunos:

Gustavo Lopes Rodrigues

Lucas Fonseca Saliba

Thiago Henriques Nogueira

Professor:

Alexei Manso Correa Machado

## Alfabeto

N°	Token	Lexema
1	const	const
2	int	int
3	char	char
4	while	while
5	if	if
6	float	float
7	else	else
8	&&	&&
9		

10	!	!
11	:=	:=
12	=	=
13	(	(
14	)	)
15	<	<
16	>	>
17	!=	!=
18	>=	>=
19	<=	<=
20	,	,
21	+	+
22	-	-
23	*	*
24	/	/
25	;	;
26	{	{
27	}	}
28	readln	readln
29	div	div
30	string	string
31	write	write
32	writeln	writeln

33	mod	mod
34	[	[
35	]	]
36	true	true
37	false	false
38	boolean	boolean

basicConst: 'c'  $\cup$  0xDD  $\cup$  "c\*"  $\cup$  d+[.d\*]  $\cup$  .d+

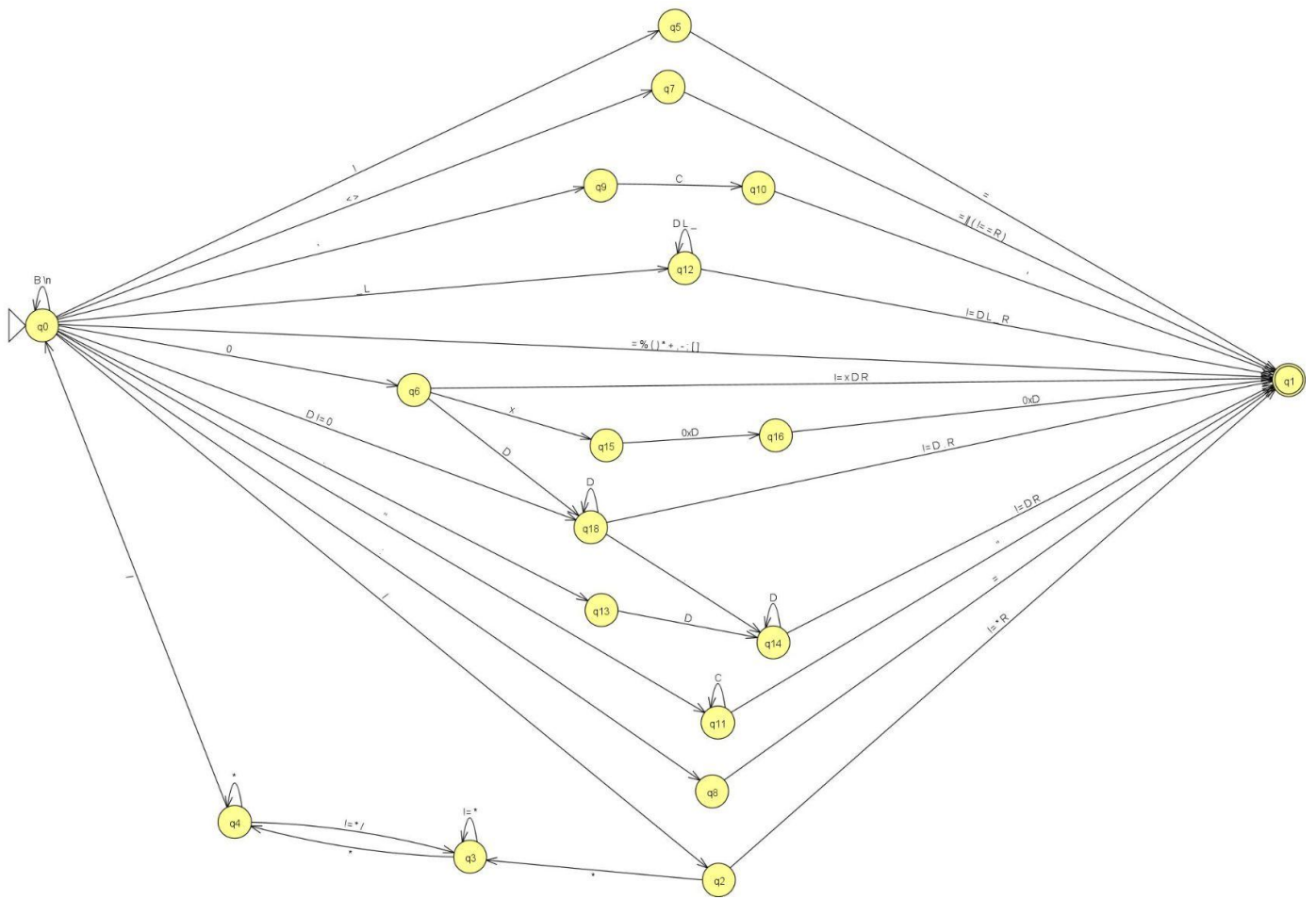
identifier: (l  $\cup$  \_)(d  $\cup$  l  $\cup$  \_)\*

#### Legenda:

Caractere Azul- Parte de ou uma Palavra Reservada.

Caractere Vermelho - Símbolo Especial.

# Analizador Léxico



## Gramática

```
S -> {(A | E)}* fim_arquivo // estado inicial
A -> (B | D) // declaracao var/const
B -> C identifier [:= [-]basicConst]{, identifier [:= [-]basicConst]}*; //
declaracao var
C -> (int | float | string | boolean | char) // tipo var
D -> const identifier = [-]basicConst; // declaracao const
E -> [(F | G | I | K | L | S)]; // comando atribuicao / while / if /
readln / write / writeln / ;
F -> identifier[[M]] := M // atribuicao
G -> while (M) H // while
H -> (E | {{E}}*) // comando / sequencia de comandos
I -> if (M) J [else J] // if
J -> (E | {{E}}*) // comando / sequencia de comandos
K -> readln(identifier) // readln
L -> (write(M {, M}*) | writeln(M {, M}*)) // write / writeln
M -> N {(= | != | < | <= | > | >=) N}* // expressao precedencia 6
N -> [+ | -]O {(+ | - | ||) O}* // expressao precedencia 5
O -> P {( * | / | mod | div | &&) P}* // expressao precedencia 4
P -> (Q | !Q) // expressao precedencia 3
Q -> (R | float(M) | int(M)) // expressao precedencia 2
R -> (basicConst | identifier[[M]] | (M)) // expressao precedencia 1
```

### Legenda:

Caractere Azul - Parte de ou uma Palavra Reservada.

Caractere Vermelho - Símbolo Não Terminal.

## Esquema de Tradução

```
S -> {(A | E)}* fim_arquivo // estado inicial
A -> (B | D) // declaracao var/const
B -> C identifier[1][6] [:= [-[7]]basicConst[8][9]] {, identifier[1][6]
[:= [-[7]]basicConst[8][9]]}*; // declaracao var
C -> (int | float | string | boolean | char)[5] // tipo var
D -> const identifier[2] = [-[7]]basicConst[8][10]; // declaracao const
E -> [(F | G | I | K | L | S)]; // comando atribuicao / while / if /
readln / write / writeln / ;
F -> identifier[3][4][[[11]M[12]]] := M1[13] // atribuicao
G -> while (M[14]) H // while
H -> (E | {{E}}*) // comando / sequencia de comandos
I -> if (M[40]) J [else J] // if
J -> (E | {{E}}*) // comando / sequencia de comandos
K -> readln(identifier[3][4][15]) // readln
L -> (write(M[16] {, M1[17]}*) | writeln(M[16] {, M1[17]}*)) // write /
writeln
M -> N[33] {(= | != | < | <= | > | >=)[34] N1[35]}* // expressao
precedencia 6
N -> [+ [29] | - [29]] 0[30] {(+ | - | ||)[31] 01[32]}* // expressao
precedencia 5
O -> P[26] {(* | / | mod | div | &&)[27] P1[28]}* // expressao precedencia
P -> (Q[24] | !Q[25]) // expressao precedencia 3
Q -> (R[21] | float(M[22]) | int(M[23])) // expressao precedencia 2
R -> (basicConst[18] | identifier[3][[[11]M[12]]][19] | (M[20])) //
expressao precedencia 1
```

### Legenda:

Caractere Azul - Parte de ou uma Palavra Reservada.

Caractere Vermelho - Símbolo Não Terminal.

[Número] Roxo - Regra com Verificação de Compatibilidade entre Classes, Tipos e Unicidade.

## Esquema de Tradução

```
[1]
{
se id.classe != vazio
erro
senao
id.classe = classe-var
}
```

```
[2]
{
se id.classe != vazio
erro
senao
id.classe = classe-const
}
```

```
[3]
{
se id.classe == vazio
erro
}
```

```
[4]
{
se id.classe != classe-var
erro
}
```

```
[5]
{
C.tipo = token.lex
}
```

```
[6]
{
id.tipo = C.tipo
}
```

```
[7]
{
const.flag = 1
}
```

```
[8]
```

```

{
se const.flag == 1 e const.tipo != real e const.tipo != inteiro
erro
}

[9]
{
se id.tipo != const.tipo e !(id.tipo == real e const.tipo == inteiro)
erro
}

[10]
{
id.tipo = const.tipo
}

[11]
{
se id.tipo != string
    erro
senao id.flag = 1
}

[12]
{
se M.tipo != inteiro
    erro
}

[13]
{
se id.flag == 1 e M1.tipo != caractere
erro
senao
se id.flag != 1 e id.tipo != M1.tipo e !(id.tipo == real e M1.tipo ==
inteiro)
erro
}

[14]
{
se M.tipo != logico
erro
}

```



```
[15]
{
se id.tipo == logico
erro
}
```

```
[16]
{
se M.tipo == logico
erro
}
```

```
[17]
{
se M1.tipo == logico
erro
}
```

```
[18]
{
R.tipo = const.tipo
}
```

```
[19]
{
se id.flag == 1
R.tipo = caractere
}
```

```
[20]
{
R.tipo = M.tipo
R.end = M.end
}
```

```
[21]
{
Q.tipo = R.tipo
Q.end = R.end
}
```

```
[22]
{
se M.tipo != real e M.tipo != inteiro
erro
senao
```

```

Q.tipo = real
Q.end = M.end
}
[23]
{
se M.tipo != real e M.tipo != inteiro
erro
senao
Q.tipo = inteiro
Q.end = M.end
}

[24]
{
P.tipo = Q.tipo
P.end = Q.end
}

[25]
{
se Q.tipo != logico
erro
}

[26]
{
O.tipo = P.tipo
O.end = P.end
}

[27]
{
O.op = token.lex
}

[28]
{
se O.op == and
se O.tipo != logico ou P1.tipo != logico
erro

senao se O.op == * ou O.op == /
se (O.tipo != real e O.tipo != inteiro) ou (P1.tipo != real e P1.tipo !=
inteiro)
erro

senao

```

```
se O.tipo != P1.tipo
O.tipo = real
```

```
senao
se O.tipo == inteiro e O.op == /
erro
```

```
senao
se O.tipo != inteiro ou P1.tipo != inteiro
erro
```

```
senao
O.tipo = P.tipo
}
```

```
[29]
{
N.flag = 1
}
```

```
[30]
{
se N.flag == 1 e O.tipo != real e O.tipo != inteiro
erro
senao
N.tipo = O.tipo
N.end = O.end
}
```

```
[31]
{
N.op = token.lex
}
```

```
[32]
{
se N.op == or
    se N.tipo != logico ou O1.tipo != logico
    erro
senao
```

```
    se (N.tipo != real e N.tipo != inteiro) ou (O1.tipo != real e
O1.tipo != inteiro)
erro
senao
se N.tipo != O1.tipo
N.tipo = real
```

```
}
```

```
[33]
```

```
{
```

```
M.tipo = N.tipo
```

```
M.end = N.end
```

```
}
```

```
[34]
```

```
{
```

```
M.op = token.lex
```

```
}
```

```
[35]
```

```
{
```

```
se M.op == "==" e M.tipo == string e N1.tipo == string
```

```
M.tipo = logico
```

```
senao se M.tipo == caractere e N1.tipo == caractere
```

```
M.tipo = logico
```

```
senao se (M.tipo == real ou M.tipo == inteiro) e (N1.tipo == real ou  
N1.tipo == inteiro)
```

```
M.tipo = logico
```

```
senao
```

```
erro
```

```
}
```

```
[40]
```

```
{
```

```
se M.tipo != logico
```

```
erro
```

```
}
```

Commits on Nov 22, 2022		
hfix: trocado um compatibility por incompatibility GustavoLR548 committed 1 hour ago		d6bcd2e
[feat] adicionado id nulo GustavoLR548 committed 2 hours ago		4dbabe7
[feat] Case && added in lexical and parser files ThiagoHN committed 2 hours ago		9bb00e3
fix: corrigido erro de atribuicao de tipo na arvore do parser (#12) GustavoLR548 committed 3 hours ago	Verified	305e9b9

Commits on Nov 22, 2022		
Error types saliba65 committed 13 minutes ago		550c1b1
Error messages saliba65 committed 13 minutes ago		1fcb972
Add work documentation saliba65 committed 2 hours ago		a1ea218
update precompiler.py saliba65 committed 2 hours ago		cd8963b
Merge branch 'main' of github.com:Trees-Over-The-Lake/Lcompiler saliba65 committed 2 hours ago		0a7516a
Esquema de traducao .txt saliba65 committed 2 hours ago		0431977
Generator (#10) GustavoLR548 and ThiagoHN committed 2 hours ago	Verified	30284e4
Comments saliba65 committed 2 hours ago		19b2e4e
Semantic2.0 (#8) GustavoLR548 committed 2 hours ago	Verified	7322e97
Commits on Nov 20, 2022		
Implementing code generator (#6) saliba65 committed 2 days ago	Verified	b68fe97

Commits on Nov 17, 2022		
[feat] Semantic Analyzer fully implemented (#5) ThiagoHN committed 5 days ago	Verified	04dda59
Commits on Nov 12, 2022		
refc: adicionado diretiva do analisador semantico GustavoLR548 committed 10 days ago		083ec87
hfix: corrigido insercao de tokens na tabela de simbolos GustavoLR548 committed 10 days ago		d030b5d
[refactor] Archives have been modified ThiagoHN committed 10 days ago		ba6fe86
Commits on Nov 9, 2022		
[feat] Initial Semantic Analyzer and Code Generator ThiagoHN committed 13 days ago		480365a
Commits on Nov 7, 2022		
Merge branch 'main' of github.com:Trees-Over-The-Lake/Lcompiler saliba65 committed 15 days ago		5ab0476
Add tp 03 and 04 saliba65 committed 15 days ago		83c4912
Refactor and otimize code (#4) saliba65 committed 15 days ago	Verified	f0095d1

Refactor (#3) 100%

GustavoLR548

committed 27 days ago

Verified

3efa39f

<>

Commits on Oct 12, 2022

Merge pull request #2 from Trees-Over-The-Lake/sint\_analyzer

...

GustavoLR548

committed on Oct 12

Verified

e7bb69

<>

feat: adicionado ignore para o unit testing

GustavoLR548

committed on Oct 12

a23389c

<>

feat: analisador sintatico feito

GustavoLR548

committed on Oct 12

d750995

<>

implementing parser

saliba65

committed on Oct 12

511742b

<>

[feat] Implementação parcial do Parser

ThiagoHN

committed on Oct 12

3b9d79e

<>

Commits on Oct 11, 2022

hfix: removido entrada e saidas

GustavoLR548

committed on Oct 11

c1e52f3

<>

feat: tp02 ate tp05 feito

GustavoLR548

committed on Oct 11

cbe8892

<>

implemmenting grammar

saliba65

committed on Oct 11

423473b

<>

implementing grammar

saliba65

committed on Oct 11

7c68738

<>

implementing grammar

saliba65

committed on Oct 11

7c68738

<>

create parser

saliba65

committed on Oct 11

4072511

<>

Commits on Oct 9, 2022

Merge pull request #1 from Trees-Over-The-Lake/lex\_analyzer

...

GustavoLR548

committed on Oct 9

Verified

3b99fd7

<>

[feat] TP2 finalizado

ThiagoHN

committed on Oct 9

82db242

<>

fix: state 1 loop in lexical analyzer,feat: adicionado leitura direta...

...

GustavoLR548

committed on Oct 9

b331885

<>

Commits on Oct 8, 2022

[fix] compilerError corrigido

ThiagoHN

committed on Oct 8

f46ecff

<>

[feat] Analisador Lexico implementado

ThiagoHN

committed on Oct 8

ec67aba

<>

Commits on Oct 5, 2022

[feat] Implementação parcial do analisador lexico

ThiagoHN

committed on Oct 5

9739e21

<>

Commits on Oct 5, 2022

Fix Lexical Analyser

saliba65 committed on Oct 5

4714f80<>

Commits on Oct 4, 2022

feat: added error manager logic

GustavoLR548 committed on Oct 4

7462482<>

Merge branch 'main' of <https://github.com/Trees-Over-The-Lake/Lcompiler>

saliba65 committed on Oct 4

f132370<>

Create analisador léxico

saliba65 committed on Oct 4

c921b1d<>

Add enunciados

saliba65 committed on Oct 4

695F24c<>

Commits on Oct 2, 2022

feat: separado a primeira entrega

GustavoLR548 committed on Oct 2

59da8a5<>

feat: primeira entrega feita

GustavoLR548 committed on Oct 2

6877023<>

feat: melhorado precompilador para revitar import ciclicos

GustavoLR548 committed on Oct 2

601d7df<>

valid char and valid hexa

saliba65 committed on Oct 2

3a86b5b<>

hfix: algumas correcoes ao analisador lexico

GustavoLR548 committed on Oct 2

72379cf<>

import analisador lexico

saliba65 committed on Oct 2

c-f51568<>

read file and count lines

saliba65 committed on Oct 2

e316f0e<>

Merge branch 'main' of <https://github.com/Trees-Over-The-Lake/Lcompiler>

saliba65 committed on Oct 2

f538e85<>

iniciate analisador lexico

saliba65 committed on Oct 2

9afab3c<>

adicionado entrada e saida do verde

GustavoLR548 committed on Oct 2

f9059b0<>

feat: adicionado tab no print da tabela de simbolos

GustavoLR548 committed on Oct 2

6abb984<>

Commits on Sep 30, 2022

Update README.md

GustavoLR548 committed on Sep 30

Verifiedd7ed455<>

Update README.md

GustavoLR548 committed on Sep 30

Verified461c718<>

adicionado opcoes de debug ao makefile, adicionado sistema de leitura...

GustavoLR548 committed on Sep 30

13d12bb<>

Commits on Sep 26, 2022

Create README.md

GustavoLR548 committed on Sep 26

Verified2e9d1b8<>

feat: feito estrutura basica da tabela de simbolos

GustavoLR548 committed on Sep 26

c905eb5<>

feat: implementacao da tabela de simbolos

GustavoLR548 committed on Sep 26

39546cc<>

[feat] adicionando funcionalidade lexema

ThiagoHN committed on Sep 26

7ab59b2<>

hfix: fixed types, feat: added lexemes

GustavoLR548 committed on Sep 26

5c75d68<>

Commits on Sep 25, 2022

[feat] adicionando funcionalidade token

ThiagoHN committed on Sep 25

018271d<>


feat: adicionado arquivo exemplo e entrega exemplo

GustavoLR548 committed on Sep 25

5c0fed6<>

Commits on Sep 24, 2022

refc: pastas modificadas

 GustavoLR548 committed on Sep 24



0906325



feat: adicionado precompilador e arquivos iniciais do projeto

 GustavoLR548 committed on Sep 24



a5ca4d9



Commits on Aug 13, 2022

Update README.md

 GustavoLR548 committed on Aug 13

Verified



2f5faa5



Update README.md

 GustavoLR548 committed on Aug 13

Verified



a9de9bf



hfix: removed build

 GustavoLR548 committed on Aug 13

Unverified



b34f77a



feat: added FMT, added cmake

 GustavoLR548 committed on Aug 13


Unverified



e877447



feat: added gitignore for CMake

 GustavoLR548 committed on Aug 13

Unverified



69adbab



Initial commit

 GustavoLR548 committed on Aug 13

Verified



4f8c6c5

