

Construção de Compiladores

Bruno Ferreira Tomé, Gabriel Peres da Silva, Tainá Silva Santos

¹Faculdade de Computação – Universidade Federal de Uberlândia (UFU)
Uberlândia — MG – Brasil

{brunoferreira.tome, gabrielperessilva@ufu.br, taina.ssantos@ufu.br}

1. Projeto da Linguagem

1.1. Definição dos Padrões (Expressões Regulares) de cada token

- **letra** $\rightarrow [A-Za-z]$
- **digito** $\rightarrow [0-9]$
- **digitos** $\rightarrow \text{digito}^+$
- **ID** $\rightarrow \text{letra}[\text{letra digito}]^*$
- **relop** $\rightarrow = | < > | < | > | < = | > =$
- **ws** $\rightarrow (\" \" | \backslash t | \backslash n)$
- **CONST_CHAR** $\rightarrow '.'$
- **CONST_NUM** $\rightarrow \text{digitos}(. \text{digitos})?(E[+-]? \text{digitos})?$
- **COMM** $\rightarrow \% (. | \backslash n)^* \%$

1.2. Tabela com Identificação dos Tokens

Lexema	Token	Retorno
programa	programa	-
/*	/*	-
*/	*/	-
int	int	-
char	char	-
float	float	-
<-	<-	-
;	;	-
,	,	-
:	:	-
se	se	-
entao	entao	-
senao	senao	-
%	-	-
enquanto	enquanto	-
faca	faca	-
repita	repita	-
ate	ate	-
Identificador	ID	Posição na Tabela de Símbolos
=	relop	EQ
<>	relop	NE
<	relop	LT
>	relop	GT
<=	relop	LE
>=	relop	GE
+	+	-
-	-	-
*	*	-
/	/	-
^	^	-
->	->	-
Caracter entre apóstrofo	CONST_CHAR	Posição na Tabela de Símbolos
Constante numérica	CONST_NUM	Posição na Tabela de Símbolos
((-
))	-
qualquer ws	-	-

Tabela 1. Tabela de tokens

2. Análise Léxica

2.1. Diagramas de Transição

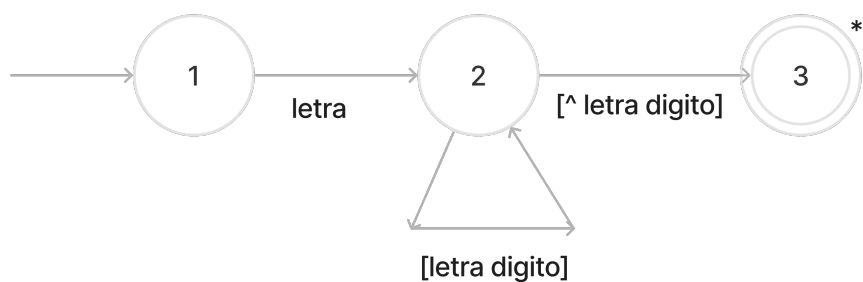


Figura 1. Diagrama ID

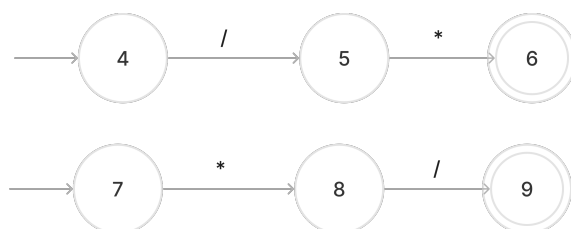


Figura 2. Diagramas Bloco

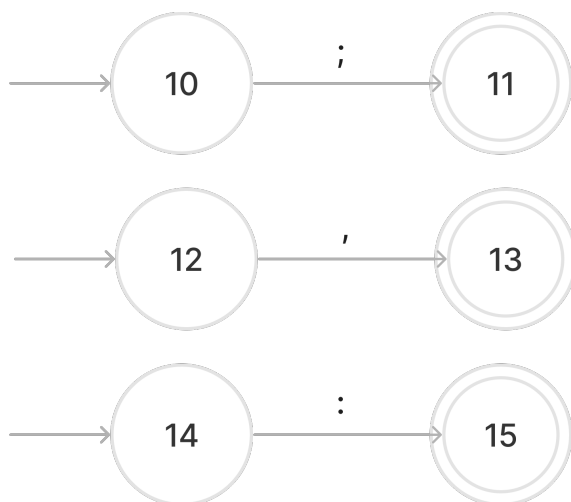


Figura 3. Diagramas Pontuações

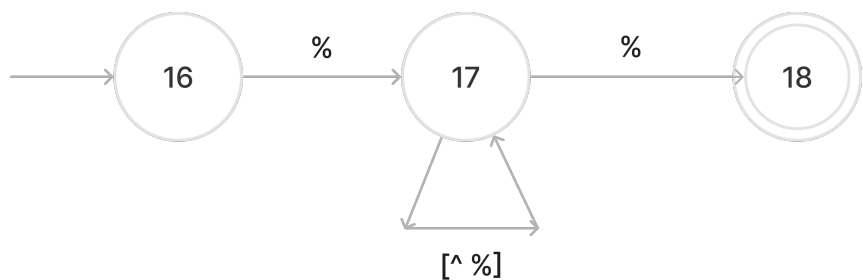


Figura 4. Diagrama Comentários

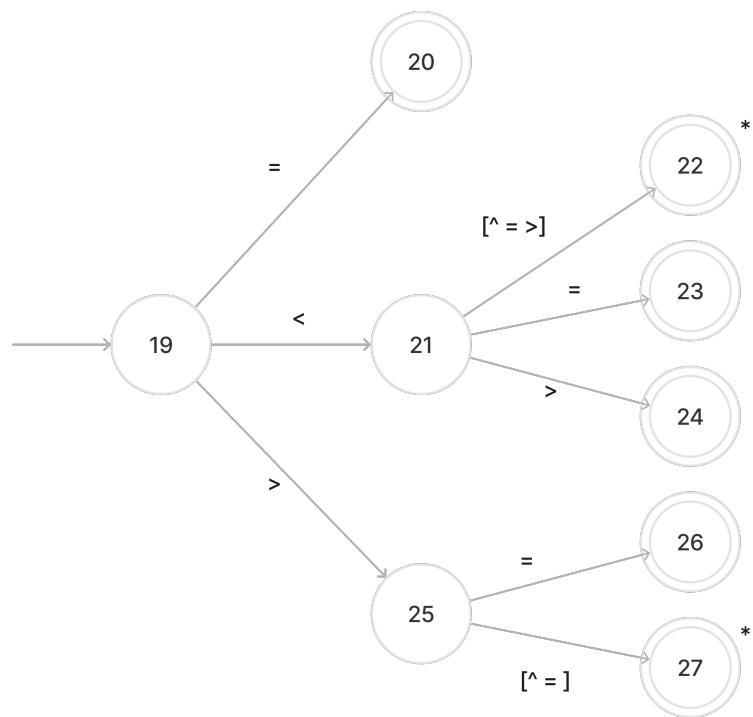


Figura 5. Diagrama Relop

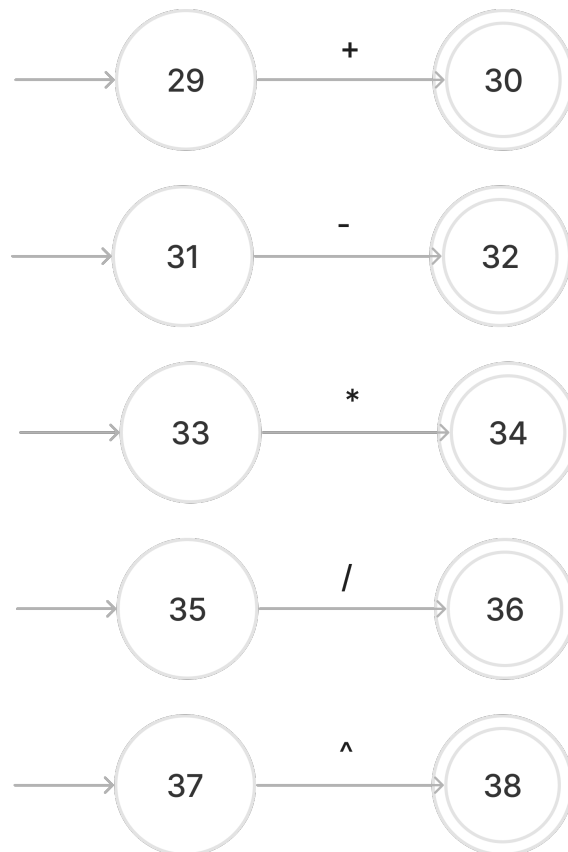


Figura 6. Diagramas Operações

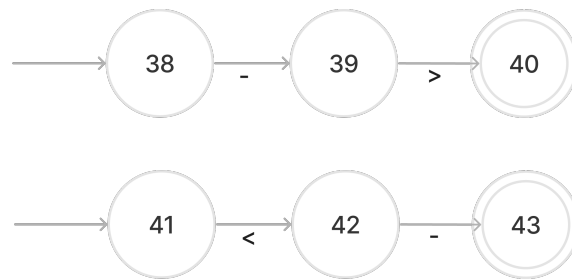


Figura 7. Diagramas Atribuições

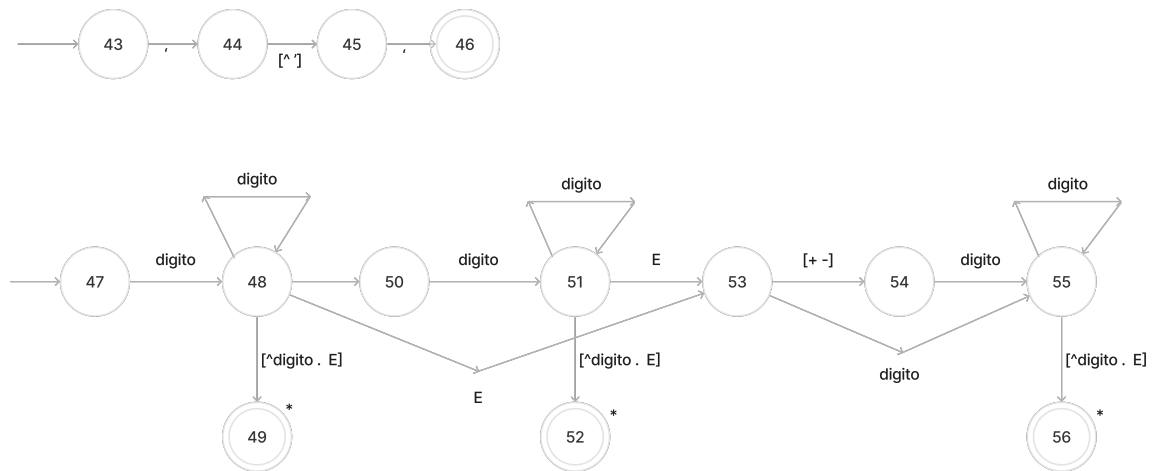


Figura 8. Diagramas Tipos de Dados

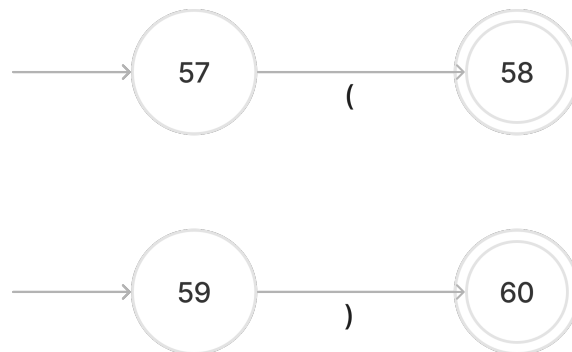


Figura 9. Diagramas Parênteses

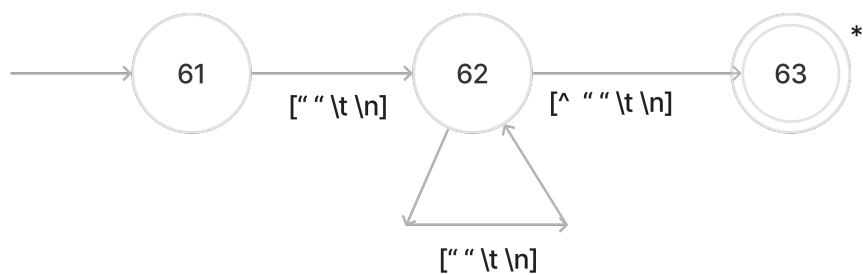


Figura 10. Diagrama ws

2.2. Diagrama Não Determinístico

Imagem (afnd.png) enviada em arquivo separado. Motivo: pesada.

Figura 11. Diagrama AFND

2.3. Diagrama de Transição Determinístico

Imagem (afd.png) enviada em arquivo separado. Motivo: pesada.

Figura 12. Diagrama AFD

3. Análise Sintática

3.1. Definição da Gramática Livre de Contexto

A gramática da linguagem é representada da seguinte maneira:

$$G = (V, T, P, S)$$

- Em que:
 - T : conjunto dos símbolos terminais
 - V : conjunto dos símbolos não terminais
 - P : conjunto de produções
 - S : símbolo inicial

Símbolos terminais:

- programa
- /*
- */
- int
- char
- float
- <-
- se
- (
-)
- entao
- senao
- %
- enquanto
- faca
- repita
- ate
- ID
- ->
- ;
- ,

- :
- relop
- +
- -
- *
- ^
- CONST_NUM
- CONST_CHAR

Símbolos não terminais:

- inicio
- bloco
- declaracao_vars
- declaracao_var
- tipo
- lista_ids
- comandos
- comando
- comando_atributo
- comando_repeticao1
- comando_repeticao2
- comando_selecao
- condicao
- expressao
- termo
- fator

3.2. Derivações

`<inicio> ::= programa ID () <bloco>`

`<bloco> ::= /* <declaracao_vars><comandos> */`

`<declaracao_vars> ::= <declacarao_var><declaracao_vars> | ;`

`<declaracao_var> ::= <tipo> -> <lista_ids>`

`<tipo> ::= int | char | float`

`<lista_ids> ::= ID <lista_ids'>`

`<lista_ids> ::= , ID <lista_ids'>`

`<comandos> ::= <comando><comandos>`

`<comando> ::= <comando_atributo><comando_repeticao1>
<comando_repeticao2><comando_selecao>`

```

<comando_atributo> ::= ID <- <expressao> ;

<comando_repeticao1> ::= enquanto <condicao> faca
<comando_ou_bloco>

<comando_repeticao2> ::= repita <comando_ou_bloco> ate <condicao> ;

<comando_selecao> ::= se <condicao> entao <comando_ou_bloco>
| se <condicao> entao <comando_ou_bloco> senao <comando_ou_bloco>

<comando_ou_bloco> ::= <comando> | <bloco>

<condicao> ::= <expressao> relop <expressao>

<expressao> ::= <expressao> <+> <termo>
| <expressao> <-> <termo> | <termo>

<termo> ::= <termo> <*> <expo> | <termo> </> <expo>
| <expo>

<expo> ::= <expo> <^> <fator> | <fator>

<fator> (<expressao>) | const_char | const_num | ID

```

3.3. Gramática na forma LL

Removendo recursão à esquerda imediata para `|expressao|`

```

<expressao> ::= <termo><expressao'>

<expressao'> ::= <+> <termo><expressao'> | <-> <termo>
<expressao'> | ε

<termo> ::= <expo><termo'>

<termo'> ::= <+> <expo><termo'> | <-> <expo>
<termo'> | ε

<expo> ::= <fator><expo'>

<expo'> ::= <^> <fator><expo'> | ε

<fator> (<expressao>) | const_char | const_num | ID

```


Removendo ambiguidade para problema do "else vazio", ou seja, fatoração a esquerda:

```

<comando_selecao> ::= se <condicao> entao
<comando_ou_bloco> <comando_selecao'>
<comando_selecao'> ::= senao <comando_ou_bloco> | ε

```

3.4. FIRST e FOLLOW dos símbolos da gramática

Símbolo	Conjunto do FISRT
programa	programa
/*	/*
*/	*/
int	int
float	float
char	char
<-	<-
se	se
entao	entao
senao	senao
%	%
enquanto	enquanto
faca	faca
ate	ate
ID	ID
->	
;	;
,	,
:	:
relop	relop
+	+
-	-
*	*
^	^
CONST_NUM	CONST_NUM
CONST_CHAR	CONST_CHAR
inicio	bloco
bloco	/*
declaracao_vars	ε, int, float, char
comandos	ε, ID, enquanto, repita, se
declaracao_var	int, float, chat
tipo	int, float, chat
lista_ids	ID
lista_ids'	ε, ,
comando	ε, ID, enquanto, repita, se

comando_atributo	ID
comando_repeticao1	enquanto
comando_repeticao2	repita
comando_selecao	se
comando_selecao'	senao
comando_ou_bloco'	ID, enquanto, repita, se,
condicao	(, CONST_CHAR, CONST_NUM, ID
expressao	(, CONST_CHAR, CONST_NUM, ID
expressao'	+, -, ϵ
termo	CONST_CHAR, CONST_NUM, ID
termo'	*, /, ϵ
expo	CONST_CHAR, CONST_NUM, ID
expo'	\wedge , ϵ
fator	CONST_CHAR, CONST_NUM, ID

Tabela 2. Tabela com FISRT de todos os símbolos da gramática

Símbolo	Conjunto do FOLLOW
inicio	{ $\$$ }
bloco	{ $\$, ID, enquanto, faca, repita, ate, se, senao, ate$ }
declaracao_vars	{ $ID, enquanto, repita, se$ }
declaracao_var	{ $int, char, float, ID, enquanto, repita, se$ }
tipo	{ $- >$ }
lista_ids	{ $;$ }
lista_ids'	{ $;$ }
comandos	{ $/$ }
comando	{ $ID, enquanto, repita, se, /, ate, senao$ }
comando_atributo	{ $ID, enquanto, repita, se, /, ate, senao$ }
comando_repeticao1	{ $ID, enquanto, repita, se, /, ate, senao$ }
comando_repeticao2	{ $ID, enquanto, repita, se, /, ate, senao$ }
comando_selecao	{ $ID, enquanto, repita, se, /, ate, senao$ }
comando_selecao'	{ $ID, enquanto, repita, se, /, ate, senao$ }
comando_ou_bloco	{ $ID, enquanto, repita, se, /, ate, senao$ }
expressao	{ $relop, ;$ }
expressao'	{ $relop, ;$ }
termo	{ $+, -, relop, ;$ }
termo'	{ $+, -, relop, ;$ }
expo	{ $*, /, +, -, relop, ;$ }
expo'	{ $*, /, +, -, relop, ;$ }
fator	{ $\wedge, *, /, +, -, relop, ;$ }

Tabela 3. Tabela com FOLLOW de todos os símbolos não terminais

3.5. Análise Preditiva

Enumeração das produções

```
1.      €
2.      programa ID () <bloco>
3.      /* <declaracao_vars><comandos> */
4.      <declacarao_var><declaracao_vars>
5.      <tipo> -> <lista_ids> ;
6.      int
7.      char
8.      float
9.      ID <lista_ids'>
10.     , ID <lista_ids'>
11.     <comando><comandos>
12.     <comando_atributo>
13.     <comando_repeticao1>
14.     <comando_repeticao2>
15.     <comando_selecao>
16.     ID <- <expressao> ;
17.     enquanto (<condicao>) faca <comando_ou_bloco>
18.     repita <comando_ou_bloco> ate (<condicao>) ;
19.     se (<condicao>) entao <comando_ou_bloco> <comando_selecao'>
20.     senao <comando_ou_bloco>
21.     <comando>
22.     <bloco>
23.     <expressao> relop <expressao>
24.     <termo> <expressao'>
25.     <+> <termo> <expressao'>
26.     <-> <termo> <expressao'>
27.     <expo> <termo'>
28.     <*> <expo> <termo'>
29.     </> <expo> <termo'>
30.     <fator> <expo'>
31.     <^> <fator> <expo'>
32.     (<expressao>)
33.     CONST_CHAR
34.     CONST_NUM
35.     ID
```

Não Terminal	Terminal	Produção
inicio	programa	P2
Bloco	/*	P3
declaracao_vars	int	P4
	float	P4
	char	P4
	ID	P1
	enquanto	P1
	repita	P1
	se	P1
	*/	P1
declaracao_var	int	P5
	float	P5
	char	P5
tipo	int	P6
	float	P7
	char	P8
lista_ids	ID	P9
lista_ids'	,	P10
	;	P1
comandos	ID	P11
	enquanto	P11
	repita	P11
	se	P11
	*/	P1
comando	ID	P12
	enquanto	P13
	repita	P14
	se	P15
comando_atribuicao	ID	P16
comando_repeticao1	enquanto	P17
comando_repeticao2	repita	P18
comando_selecao	se	P19
comando_selecao'	senao	P20
	ID	P1
	enquanto	P1
	repita	P1
	se	P1
	*/	P1
	ate	P1
comando_ou_bloco'	ID	P21
	enquanto	P21
	repita	P21
	se	P21
	/*	P22

condicao	(P23
	CONST_CHAR	P23
	CONST_NUM	P23
	ID	P23
expressao	(P24
	CONST_CHAR	P24
	CONST_NUM	P24
	ID	P24
expressao'	+	P25
	-	P26
	relop	P1
)	P1
	;	P1
termo	(P27
	CONST_CHAR	P27
	CONST_NUM	P27
	ID	P27
termo'	*	P28
	/	P29
	+	P1
	-	P1
	relop	P1
)	P1
	;	P1
expo	(P30
	CONST_CHAR	P30
	CONST_NUM	P30
	ID	P30
expo'	^	P31
	*	P1
	/	P1
	+	P1
	-	P1
	relop	P1
)	P1
	;	P1
fator	(P32
	CONST_CHAR	P33
	CONST_NUM	P34
	ID	P35

Tabela 4. Tabela de Análise Sintática Preditiva