

<b>Nomes:</b> <i>Gustavo Pioli Resende</i>	-	<i>12111BCC010</i>
<i>Natan Gonçalves de Lyra</i>	-	<i>12111BCC006</i>
<i>Vinicius Dias Martins Rocha</i>	-	<i>12111BCC039</i>

---

# Projeto Compiladores

## 1. Especificação Da Linguagem

### 1.1 Definição da Gramática Livre de Contexto (GLC)

Uma GLC  $G$  pode ser representada por uma quádrupla:

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

- $V$  = Conjunto dos Símbolos Não-Terminais
- $T$  = Conjunto dos Símbolos Terminais
- $P$  = Conjunto de Produções (regras de transição)
- $S$  = INICIO

**Símbolos Terminais:**

- **main**
- **ID**
- **begin**
- **end**
- **int**
- **char**
- **float**
- **if**
- **then**
- **else**
- **while**
- **do**
- **repeat**
- **until**
- **número**
- **caractere**

- letra
- relop
- ws
- {
- }
- (
- )
- :=
- ,
- ;
- :
- \*
- \*\*
- +
- -
- /

**Símbolos Não-Terminais ANTES da Fatoração e Remoção da Recursão à Esquerda:**

- <INICIO>
- <tipo>
- <bloco>
- <declaracao\_variaveis>
- <declaracao\_variavel>
- <lista\_ids>
- <sequencia\_comandos>
- <comando>
- <condição>
- <expressão>
- <comando\_selecao>
- <comando\_ou\_bloco>
- <comando\_repeticao>
- <comando\_atribuicao>
- <condicao>
- <expressao>

**Produções Gramaticais (regras de transição) ANTES da Fatoração e Remoção da Recursão à Esquerda:**

- <INICIO> ::= main ID <bloco>
- <bloco> ::= begin <declaracao\_variaveis> <sequencia\_comandos> end
- <declaracao\_variaveis> ::= <declaracao\_variavel> <declaracao\_variaveis> | ε

- **<declaracao\_variavel> ::= <tipo> : <lista\_ids> ;**
- **<tipo> ::= char | int | float**
- **<lista\_ids> ::= ID | ID, <lista\_ids>**
- **<sequencia\_comandos> ::= <comando> <sequencia\_comandos> | ε**
- **<comando> ::= <comando\_selecao> | <comando\_repeticao> | <comando\_atribuicao>**
- **<comando\_selecao> ::= if ( <condicao> ) then <comando\_ou\_bloco> | if ( <condicao> ) then <comando\_ou\_bloco> else <comando\_ou\_bloco>**
- **<comando\_ou\_bloco> ::= <comando> | <bloco>**
- **<comando\_repeticao> ::= while ( <condicao> ) do <comando\_ou\_bloco> | repeat <comando\_ou\_bloco> until ( <condicao> );**
- **<comando\_atribuicao> ::= ID := <expressao> ;**
- **<condicao> ::= <expressao> relop <expressao>**
- **<expressao> ::= num | caractere | ID | (<expressao>) | <expressao> operador <expressao>**

## 1.2 Definição da Gramática Livre de Contexto (GLC)

LEXEMA	TOKEN	ATRIBUTO
main	main	-
begin	begin	-
end	end	-
int	int	-
char	char	-
float	float	-
if	if	-
then	then	-
else	else	-
while	while	-
do	do	-
repeat	repeat	-
until	until	-
Qualquer número	NUM	Posição na tabela de símbolos
‘Qualquer caractere’	caractere	Posição na tabela de símbolos

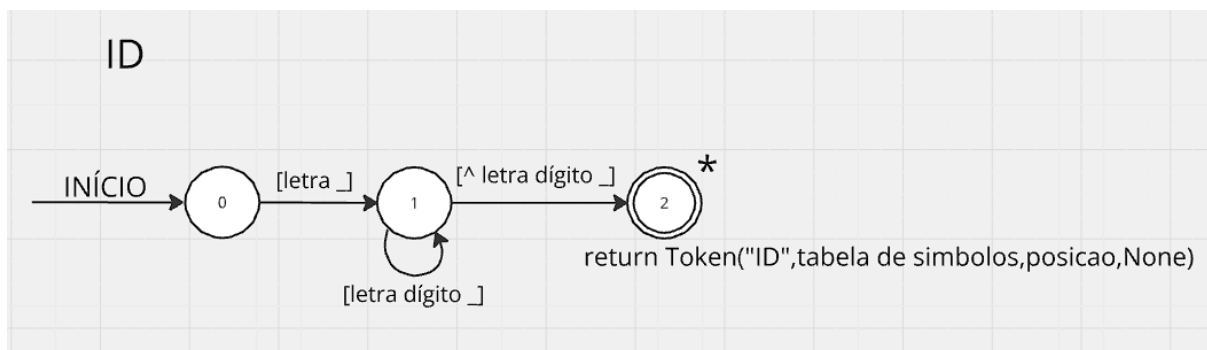
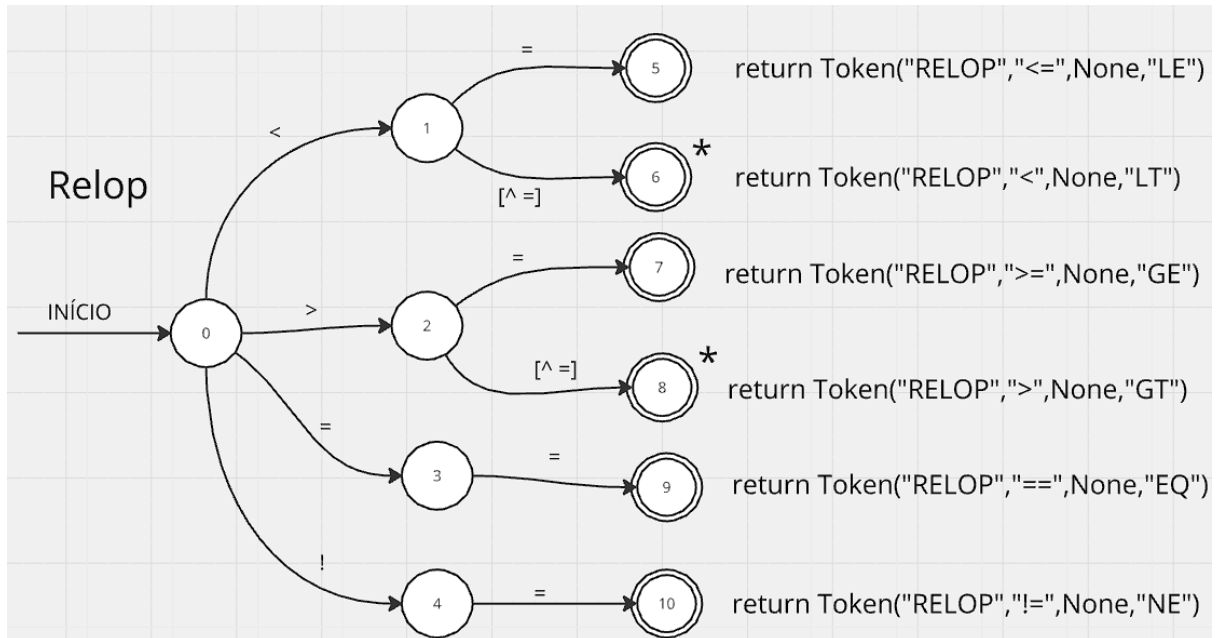
Qualquer ID	ID	Posição na tabela de símbolos
>	relop	GT
<	relop	LT
>=	relop	GE
<=	relop	LE
==	relop	EQ
!=	relop	NE
:=	:=	-
{	{	-
}	}	-
(	(	-
)	)	-
+	+	-
-	-	-
/	/	-
*	*	-
**	**	-
,	,	-
;	;	-
:	:	-
Texto entre chaves { }	-	-
ws	-	-

### 1.3 Definição dos padrões de cada token

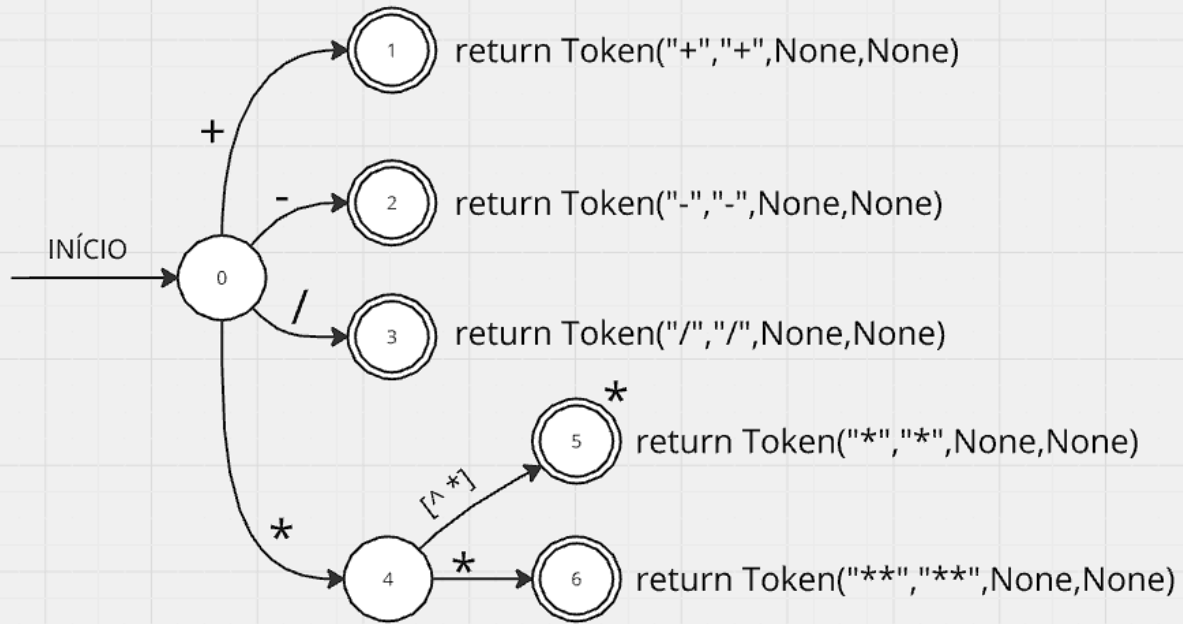
- main ➡ main
- begin ➡ begin
- end ➡ end
- int ➡ int
- char ➡ char
- float ➡ float
- if ➡ if
- then ➡ then
- else ➡ else
- while ➡ while
- do ➡ do
- repeat ➡ repeat
- until ➡ until
- { ➡ {
- } ➡ }
- ( ➡ (
- ) ➡ )
- := ➡ :=
- , ➡ ,
- ; ➡ ;
- : ➡ :
- + ➡ \+
- - ➡ -
- \* ➡ \\*
- \*\* ➡ \\*\\*
- / ➡ /
- digito ➡ [0-9]
- digitos ➡ digito+
- NUM ➡ digitos(\.digitos)?(E[\+-]?digitos)?
- letra ➡ [a-zA-Z]
- ID ➡ [letra \_][digito letra \_ ] \*
- caractere ➡ '.'
- relop ➡ < | <= | > | >= | == | !=
- comentário ➡ { [^ } ]\* }
- ws ➡ [ \t\n]

## 2. Análise Léxica

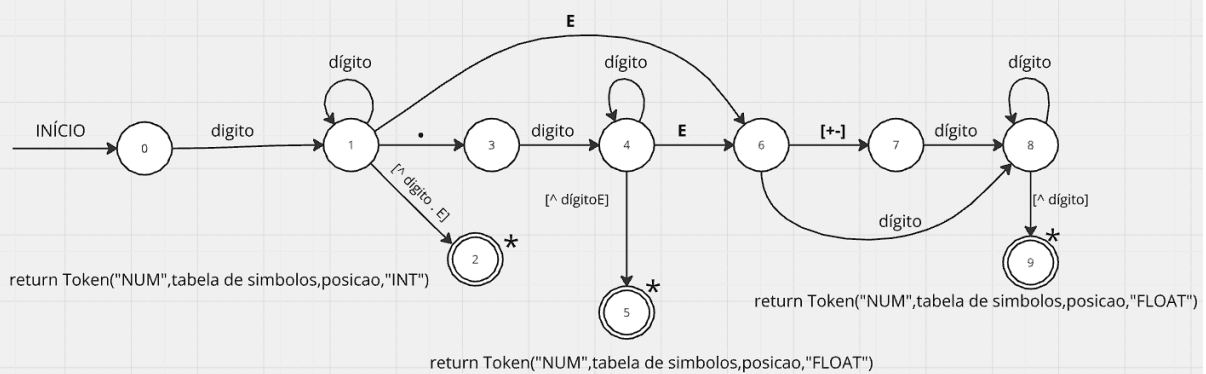
### 2.1 Diagramas de Transição para os Tokens

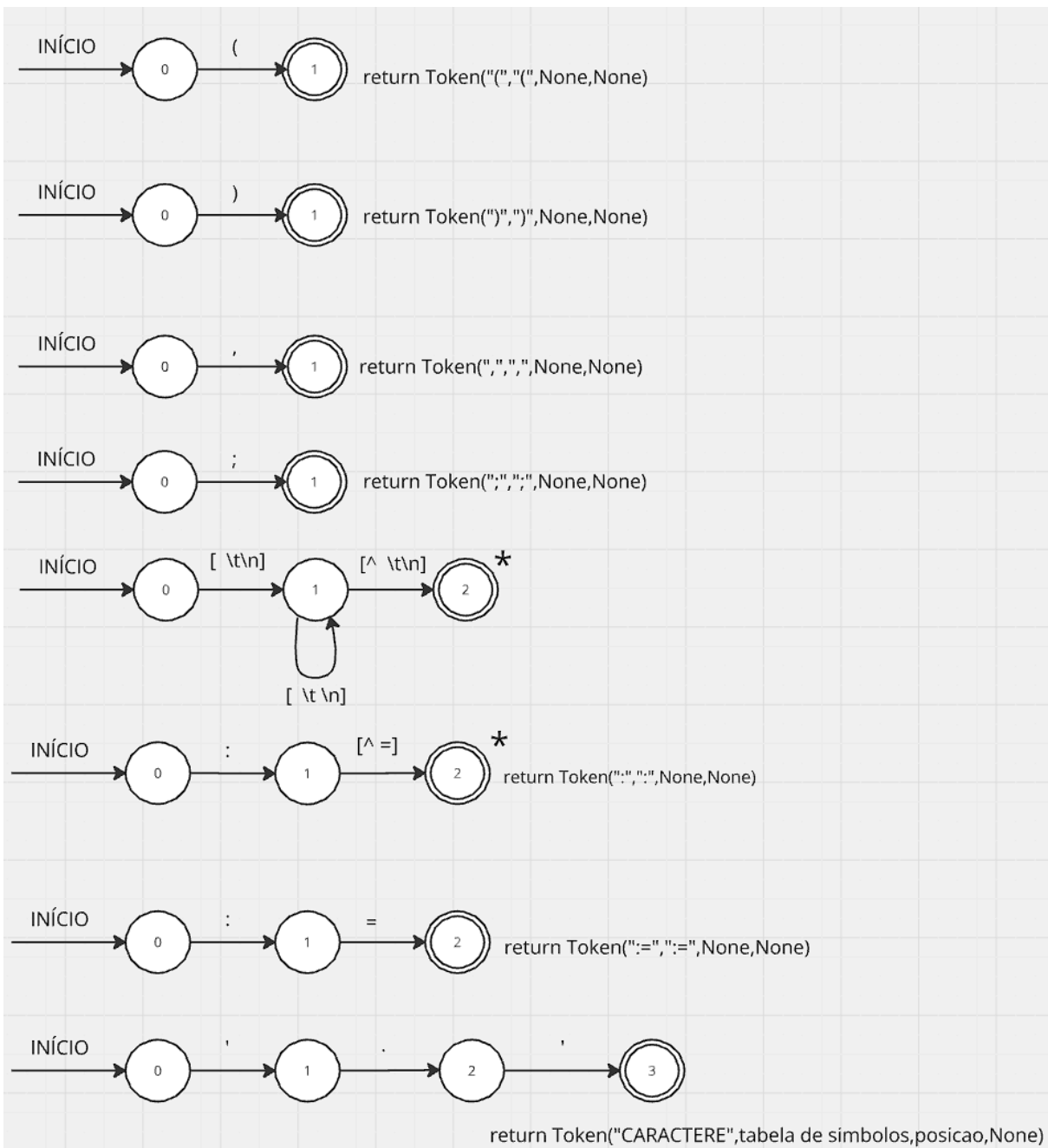


## Operador

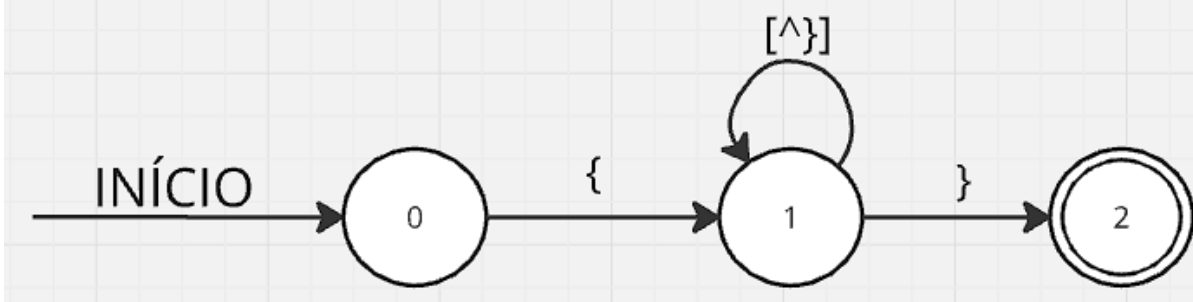


## NÚMERO

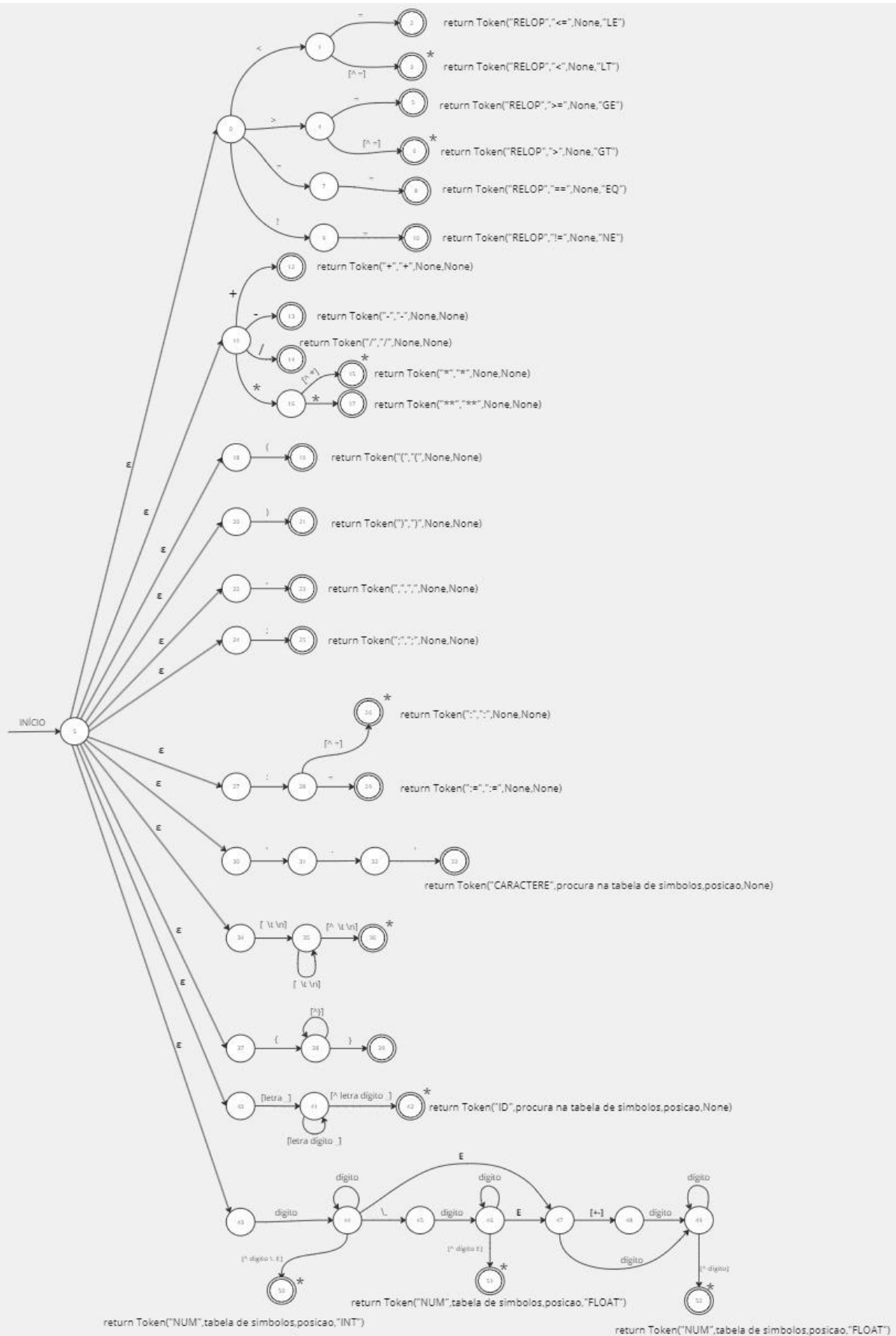


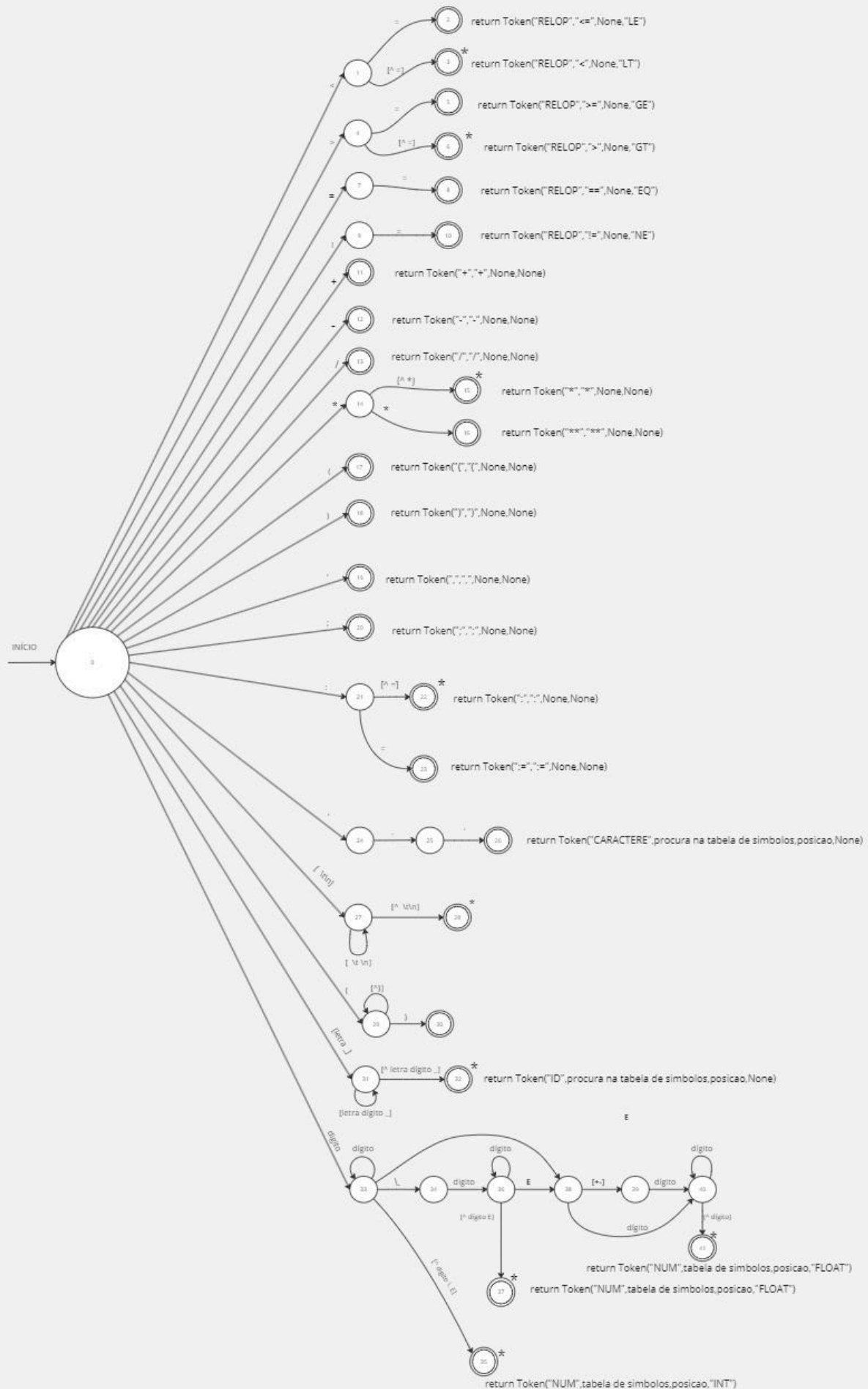


## Comentário









### 3. Análise Sintática

#### 3.1 Ajustes da GLC para que a linguagem seja LL(1)

A gramática de produção <lista\_ids> possuía ambiguidade no ID:

- <lista\_ids> ::= ID | ID, <lista\_ids>

Para remover a ambiguidade da gramática de produção <lista\_ids> foram feitos os seguintes ajustes:

- <lista\_ids> ::= ID<lista\_ids'>
- <lista\_ids'> ::= ,ID<lista\_ids'> | ε

A gramática de produção <comando\_selecao> possuía ambiguidade no if:

- <comando\_selecao> ::= if ( <condicao> ) then <comando\_ou\_bloco> | if ( <condicao> ) then <comando\_ou\_bloco> else <comando\_ou\_bloco>

Para remover a ambiguidade da gramática de produção <comando\_selecao> foram feitos os seguintes ajustes:

- <comando\_selecao> ::= if ( <condicao> ) then <comando\_ou\_bloco> <comando\_selecao'>
- <comando\_selecao'> ::= else <comando\_ou\_bloco> | ε

A gramática de produção <expressao> possuía recursão à esquerda:

- <expressao> ::= num | caractere | ID | (<expressao>) | <expressao> operador <expressao>

Para remover a recursão à esquerda da gramática de produção <expressao> foram feitos os seguintes ajustes:

- <expressao> ::= <termo><expressao'>
- <expressao'> ::= + <termo> <expressao'> | - <termo> <expressao'> | ε
- <termo> ::= <exp><termo'>
- <termo'> ::= \* <exp> <termo'> | / <exp> <termo'> | ε
- <exp> ::= <fator><exp'>
- <exp'> ::= \*\* <fator> <exp'> | ε
- <fator> ::= num | caractere | ID | (<expressao>)

**Símbolos Não-Terminais DEPOIS da Fatoração e Remoção da Recursão à Esquerda:**

- <INICIO>
- <tipo>
- <bloco>

- <declaracao\_variaveis>
- <declaracao\_variavel>
- <lista\_ids>
- <lista\_ids'>
- <sequencia\_comandos>
- <comando>
- <condição>
- <expressão>
- <comando\_selecao>
- <comando\_selecao'>
- <comando\_ou\_bloco>
- <comando\_repeticao>
- <comando\_atribuicao>
- <condicao>
- <expressao>
- <expressao'>
- <termo>
- <termo'>
- <exp>
- <exp'>
- <fator>

### Produções Gramaticais (regras de transição) DEPOIS da Fatoração e Remoção da Recursão à Esquerda:

- <INICIO> ::= main ID <bloco>
- <bloco> ::= begin <declaracao\_variaveis> <sequencia\_comandos> end
- <declaração\_variaveis> ::= <declaracao\_variavel> <declaracao\_variaveis> | ε
- <declaração\_variavel> ::= <tipo> : <lista\_ids> ;
- <tipo> ::= char | int | float
- <lista\_ids> ::= ID<lista\_ids'>
- <lista\_ids'> ::= ,ID<lista\_ids'> | ε
- <sequencia\_comandos> ::= <comando> <sequencia\_comandos> | ε
- <comando> ::= <comando\_selecao> | <comando\_repeticao> | <comando\_atribuicao>
- <comando\_selecao> ::= if ( <condicao> ) then <comando\_ou\_bloco> <comando\_selecao'>
- <comando\_selecao'> ::= else <comando\_ou\_bloco> | ε
- <comando\_ou\_bloco> ::= <comando> | <bloco>
- <comando\_repeticao> ::= while ( <condicao> ) do <comando\_ou\_bloco> | repeat <comando\_ou\_bloco> until ( <condicao> );
- <comando\_atribuicao> ::= ID := <expressao> ;
- <condicao> ::= <expressão> relop <expressão>
- <expressao> ::= <termo><expressao'>

- $\langle \text{expressao}' \rangle ::= + \langle \text{termo} \rangle \langle \text{expressao}' \rangle \mid - \langle \text{termo} \rangle \langle \text{expressao}' \rangle \mid \epsilon$
- $\langle \text{termo} \rangle ::= \langle \text{exp} \rangle \langle \text{termo}' \rangle$
- $\langle \text{termo}' \rangle ::= * \langle \text{exp} \rangle \langle \text{termo}' \rangle \mid / \langle \text{exp} \rangle \langle \text{termo}' \rangle \mid \epsilon$
- $\langle \text{exp} \rangle ::= \langle \text{fator} \rangle \langle \text{exp}' \rangle$
- $\langle \text{exp}' \rangle ::= ** \langle \text{fator} \rangle \langle \text{exp}' \rangle \mid \epsilon$
- $\langle \text{fator} \rangle ::= \text{num} \mid \text{caractere} \mid \text{ID} \mid (\langle \text{expressao} \rangle)$

### 3.2 Cálculo e Tabela de FIRST

- $\text{FIRST}(\langle \text{INICIO} \rangle) ::= \text{FIRST}(\text{main ID } \langle \text{bloco} \rangle) = \{\text{main}\}$ 
  - $\langle \text{INICIO} \rangle ::= \text{main ID } \langle \text{bloco} \rangle$
- $\text{FIRST}(\langle \text{bloco} \rangle) ::= \text{FIRST}(\text{begin } \langle \text{declaracao\_variaveis} \rangle \langle \text{sequencia\_comandos} \rangle \text{end}) = \{\text{begin}\}$
- $\text{FIRST}(\langle \text{declaração\_variaveis} \rangle) ::= \text{FIRST}(\langle \text{declaracao\_variavel} \rangle \langle \text{declaracao\_variaveis} \rangle) \cup \{\epsilon\} = \text{FIRST}(\langle \text{declaracao\_variavel} \rangle) \cup \{\epsilon\} = \{\text{char, int, float, } \epsilon\}$ 
  - $\langle \text{declaração\_variaveis} \rangle ::= \langle \text{declaracao\_variavel} \rangle \langle \text{declaracao\_variaveis} \rangle \mid \epsilon$
- $\text{FIRST}(\langle \text{declaração\_variavel} \rangle) ::= \text{FIRST}(\langle \text{tipo} \rangle : \langle \text{lista\_ids} \rangle ;) = \text{FIRST}(\langle \text{tipo} \rangle) = \{\text{char, int, float}\}$ 
  - $\langle \text{declaração\_variavel} \rangle ::= \langle \text{tipo} \rangle : \langle \text{lista\_ids} \rangle ;$
- $\text{FIRST}(\langle \text{tipo} \rangle) ::= \text{FIRST}(\text{char}) \cup \text{FIRST}(\text{int}) \cup \text{FIRST}(\text{float}) = \{\text{char, int, float}\}$
- $\text{FIRST}(\langle \text{lista\_ids} \rangle) ::= \text{FIRST}(\text{ID} \langle \text{lista\_ids}' \rangle) = \{\text{ID}\}$ 
  - $\langle \text{lista\_ids} \rangle ::= \text{ID} \langle \text{lista\_ids}' \rangle$
- $\text{FIRST}(\langle \text{lista\_ids}' \rangle) ::= \text{FIRST}(, \text{ID} \langle \text{lista\_ids}' \rangle) \cup \{\epsilon\} = \{, , \epsilon\}$
- $\text{FIRST}(\langle \text{sequencia\_comandos} \rangle) ::= \text{FIRST}(\langle \text{comando} \rangle \langle \text{sequencia\_comandos} \rangle) \cup \{\epsilon\} = \text{FIRST}(\langle \text{comando} \rangle) \cup \{\epsilon\} = \{\text{if, while, repeat, ID}\}$
- $\text{FIRST}(\langle \text{comando} \rangle) ::= \text{FIRST}(\langle \text{comando\_selecao} \rangle) \cup \text{FIRST}(\langle \text{comando\_repeticao} \rangle) \cup \text{FIRST}(\langle \text{comando\_atribuicao} \rangle) = \{\text{if, while, repeat, ID}\}$
- $\text{FIRST}(\langle \text{comando\_selecao} \rangle) ::= \text{FIRST}(\text{if } ( \langle \text{condicao} \rangle ) \text{ then } \langle \text{comando\_ou\_bloco} \rangle \langle \text{comando\_selecao}' \rangle) = \{\text{if}\}$
- $\text{FIRST}(\langle \text{comando\_selecao}' \rangle) ::= \text{FIRST}(\text{else } \langle \text{comando\_ou\_bloco} \rangle) \cup \{\epsilon\} = \{\text{else}, \epsilon\}$

- $\text{FIRST}(\langle \text{comando\_ou\_bloco} \rangle) ::= \text{FIRST}(\langle \text{comando} \rangle) \cup \text{FIRST}(\langle \text{bloco} \rangle) = \{\mathbf{if, while, repeat, ID, begin}\}$ 
  - $\langle \text{comando\_ou\_bloco} \rangle ::= \langle \text{comando} \rangle \mid \langle \text{bloco} \rangle$
- $\text{FIRST}(\langle \text{comando\_repeticao} \rangle) ::= \text{FIRST}(\text{while}(\langle \text{condicao} \rangle) \text{ do } \langle \text{comando\_ou\_bloco} \rangle) \cup \text{FIRST}(\text{repeat } \langle \text{comando\_ou\_bloco} \rangle \text{ until } (\langle \text{condicao} \rangle);)$   
 $= \{\mathbf{while, repeat}\}$
- $\text{FIRST}(\langle \text{comando\_atribuicao} \rangle) ::= \text{FIRST}(\text{ID} := \langle \text{expressao} \rangle;) = \{\mathbf{ID}\}$
- $\text{FIRST}(\langle \text{condicao} \rangle) ::= \text{FIRST}(\langle \text{expressao} \rangle \text{ relop } \langle \text{expressao} \rangle) = \text{FIRST}(\langle \text{expressao} \rangle) = \{\mathbf{num, caractere, ID, \{}}\}$ 
  - $\langle \text{condicao} \rangle ::= \langle \text{expressao} \rangle \text{ relop } \langle \text{expressao} \rangle$
- $\text{FIRST}(\langle \text{expressao} \rangle) ::= \text{FIRST}(\langle \text{termo} \rangle \langle \text{expressao}' \rangle) = \text{FIRST}(\langle \text{termo} \rangle) = \{\mathbf{num, caractere, ID, \{}}\}$ 
  - $\langle \text{expressao} \rangle ::= \langle \text{termo} \rangle \langle \text{expressao}' \rangle$
- $\text{FIRST}(\langle \text{expressao}' \rangle) ::= \text{FIRST}(+ \langle \text{termo} \rangle \langle \text{expressao}' \rangle) \cup \text{FIRST}(- \langle \text{termo} \rangle \langle \text{expressao}' \rangle) \cup \{\epsilon\} = \{+, -, \epsilon\}$
- $\text{FIRST}(\langle \text{termo} \rangle) ::= \text{FIRST}(\langle \text{exp} \rangle \langle \text{termo}' \rangle) = \text{FIRST}(\langle \text{exp} \rangle) = \{\mathbf{num, caractere, ID, \{}}\}$
- $\text{FIRST}(\langle \text{termo}' \rangle) ::= \text{FIRST}(* \langle \text{exp} \rangle \langle \text{termo}' \rangle) \cup \text{FIRST}(/ \langle \text{exp} \rangle \langle \text{termo}' \rangle) \cup \{\epsilon\} = \{*, /, \epsilon\}$
- $\text{FIRST}(\langle \text{exp} \rangle) ::= \text{FIRST}(\langle \text{fator} \rangle \langle \text{exp}' \rangle) = \text{FIRST}(\langle \text{fator} \rangle) = \{\mathbf{num, caractere, ID, \{}}\}$
- $\text{FIRST}(\langle \text{exp}' \rangle) ::= \text{FIRST}(** \langle \text{fator} \rangle \langle \text{exp}' \rangle) \cup \{\epsilon\} = \{**, \epsilon\}$
- $\text{FIRST}(\langle \text{fator} \rangle) ::= \text{FIRST}(\text{num}) \cup \text{FIRST}(\text{caractere}) \cup \text{FIRST}(\text{ID}) \cup \text{FIRST}(\langle \text{expressao} \rangle) = \{\mathbf{num, caractere, ID, \{}}\}$

Símbolos (Terminais e Não-Terminais)	FIRST
main	main
ID	ID
begin	begin
end	end

int	int
char	char
float	float
:	:
(	(
)	)
;	;
if	if
then	then
else	else
while	while
do	do
repeat	repeat
until	until
relop	relop
:=	:=
,	,
+	+
-	-
*	*
**	**
/	/
num	num
caractere	caractere
<fator>	{num, ID, (, caractere}
<exp'>	{**, ε }
<exp>	{num, ID, (, caractere}

<termo'>	{*, /, ε }
<termo>	{num, ID, (, caractere}
<expressao'>	{+, -, ε }
<expressao>	{num, ID, (, caractere}
<condicao>	{num, ID, (, caractere}
<comando_atribuicao>	{ID}
<comando_repeticao>	{while, repeat}
<comando_selecao'>	{else, ε }
<comando_selecao>	{if}
<comando>	{while, repeat, if, ID}
<bloco>	{begin}
<comando_ou_bloco>	{while, repeat, if, ID, begin}
<sequencia de comandos>	{while, repeat, if, ID, ε}
<lista_ids'>	{, , ε}
<lista_ids>	{ID}
<tipo>	{char, int, float}
<declaracao_variavel>	{char, int, float}
<declaracao_variaveis>	{char, int, float, ε}
<INICIO>	{main}

### 3.3 Cálculo e Tabela de FOLLOW

- FOLLOW(<INICIO>) ::= FIRST(\$) = {\$}
  - <INICIO> ::= **main ID** <bloco>
- FOLLOW(<bloco>) ::= FOLLOW(<comando\_ou\_bloco>) U FOLLOW(<bloco>) = {else,while, repeat, if, ID, end, until}
- FOLLOW(<declaracao\_variaveis>) ::= FIRST(<sequencia\_comandos>end) - {ε} U FOLLOW(<declaracao\_variaveis>) = {while,repeat,if,ID,end}



- FOLLOW(<declaracao\_variavel>) ::= FIRST(<declaracao\_variaveis>) - { $\epsilon$ } U  
FOLLOW(<declaracao\_variaveis>) = {char, int, float, while, repeat, if, ID, end}
- FOLLOW(<tipo>) ::= FIRST( : <lista\_ids> ;) = {;}
- FOLLOW(<lista\_ids>) ::= FIRST(;) = {;}
- FOLLOW(<lista\_ids'>) ::= FOLLOW(<lista\_ids>) U FOLLOW(<lista\_ids'>) = {;}
- FOLLOW(<sequencia\_comandos>) ::= FIRST(end) U  
FOLLOW(<sequencia\_comandos>) = {end}
  - <sequencia\_comandos> ::= <comando> <sequencia\_comandos> |  $\epsilon$
  - <bloco> ::= begin <declaracao\_variaveis> <sequencia\_comandos> end
- FOLLOW(<comando>) ::= FIRST(<sequencia\_comandos>) - { $\epsilon$ } U  
FOLLOW(<comando\_ou\_bloco>) = {while, repeat, if, ID, end, else, until}
  - <comando> ::= <comando\_selecao> | <comando\_repeticao> |  
<comando\_atribuicao>
  - <sequencia\_comandos> ::= <comando> <sequencia\_comandos> |  $\epsilon$
  - <comando\_ou\_bloco> ::= <comando> | <bloco>
- FOLLOW(<comando\_selecao>) ::= FOLLOW(<comando>) = {while, repeat, if, ID, end}
- FOLLOW(<comando\_selecao'>) ::= FOLLOW(<comando\_selecao>) = {while, repeat, if, ID, end}
- FOLLOW(<comando\_ou\_bloco>) ::= FIRST(<comando\_selecao'>) - { $\epsilon$ } U  
FOLLOW(<comando\_selecao'>) U FOLLOW(<comando\_repeticao>) U FIRST(until  
( <condicao> );) = {while, repeat, if, ID, end, else, until}
- FOLLOW(<comando\_repeticao>) ::= FOLLOW(<comando>) = {while, repeat, if, ID, end}
- FOLLOW(<comando\_atribuicao>) ::= FOLLOW(<comando>) = {while, repeat, if, ID, end}
- FOLLOW(<condicao>) ::= FIRST( ) then <comando\_ou\_bloco>) U FIRST( ) do  
<comando\_ou\_bloco>) U FIRST( );) = {}
  - <condicao> ::= <expressão> relop <expressão>
  - <comando\_selecao> ::= if ( <condicao> ) then <comando\_ou\_bloco>  
<comando\_selecao'>

- **<comando\_repeticao> ::= while ( <condicao> ) do <comando\_ou\_bloco> | repeat <comando\_ou\_bloco> until ( <condicao> );**
- FOLLOW(<expressao>) ::= FIRST(;) U FIRST(relop <expressão>) U FIRST( ) = {  
; , relop, )}
- **<expressao> ::= <termo><expressao'>**
- **<comando\_atribuicao> ::= ID := <expressao> ;**
- **<condicao> ::= <expressao> relop <expressao>**
- **<fator> ::= num | caractere | ID | (<expressao>)**
- FOLLOW(<expressao'>) ::= FOLLOW(<expressao>) U FOLLOW(<expressao'>) = {  
; , relop, )}
- FOLLOW(<termo>) ::= FIRST(<expressao'>) - {ε} = {+, -}
- **<termo> ::= <exp><termo'>**
- **<expressao> ::= <termo><expressao'>**
- **<expressao'> ::= + <termo> <expressao'> | - <termo> <expressao'> | ε**
- FOLLOW(<termo'>) ::= FOLLOW(<termo>) U FOLLOW(<termo'>) = {+, -}
- FOLLOW(<exp>) ::= FIRST(<termo'>) - {ε} = {\*, /}
- FOLLOW(<exp'>) ::= FOLLOW(<exp>) U FOLLOW(<exp'>) = {\*, /}
- FOLLOW(<fator>) ::= FIRST(<exp'>) - {ε} = {\*\*}

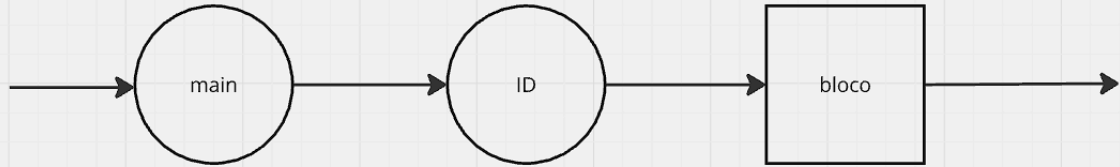
Símbolos Não-Terminais	FOLLOW
<INICIO>	{ \$ }
<bloco>	{ while, repeat, if, ID, end, else, \$, until }
<declaracao_variaveis>	{ while, repeat, if, ID, end }
<declaracao_variavel>	{ char, int, float, while, repeat, if, ID, end }
<tipo>	{ : }
<lista_ids>	{ ; }
<lista_ids'>	{ ; }
<sequencia_comandos>	{ end }
<comando>	{ while, repeat, if, ID, end, else, until }

<comando_ou_bloco>	{while, repeat, if, ID, end, else, until}
<comando_selecao>	{while, repeat, if, ID, end, else, until}
<comando_selecao’>	{while, repeat, if, ID, end, else, until}
<comando_atribuicao>	{while, repeat, if, ID, end, else, until}
<comando_repeticao>	{while, repeat, if, ID, end, else, until}
<condicao>	{}
<expressao>	{;, ), relop}
<expressao’>	{;, ), relop}
<termo>	{+, -, ;, ), relop}
<termo’>	{+, -, ;, ), relop}
<exp>	{*, /, +, -, ;, ), relop}
<exp’>	{*, /, +, -, ;, ), relop}
<fator>	{**, *, /, +, -, ;, ), relop}

### 3.4 Construção dos Grafos Sintáticos

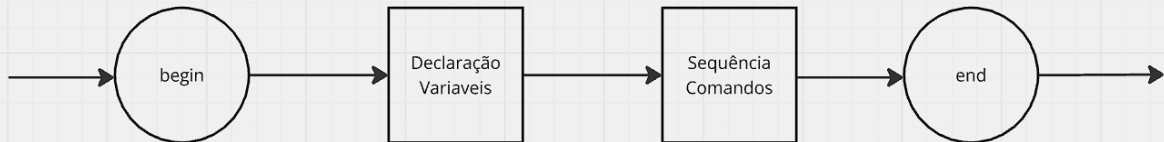
- **<INICIO> ::= main ID <bloco>**

Início



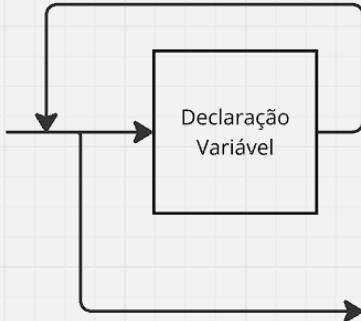
- **<bloco> ::= begin <declaracao\_variaveis> <sequencia\_comandos> end**

Bloco



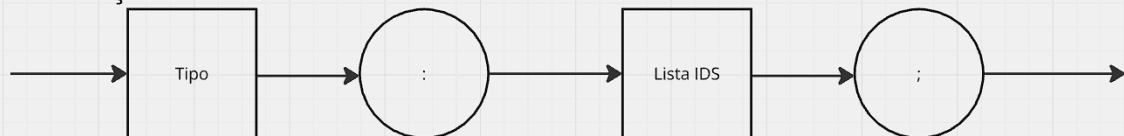
- **<declaração\_variaveis> ::= <declaracao\_variavel> <declaracao\_variaveis> | ε**

Declaração Variáveis

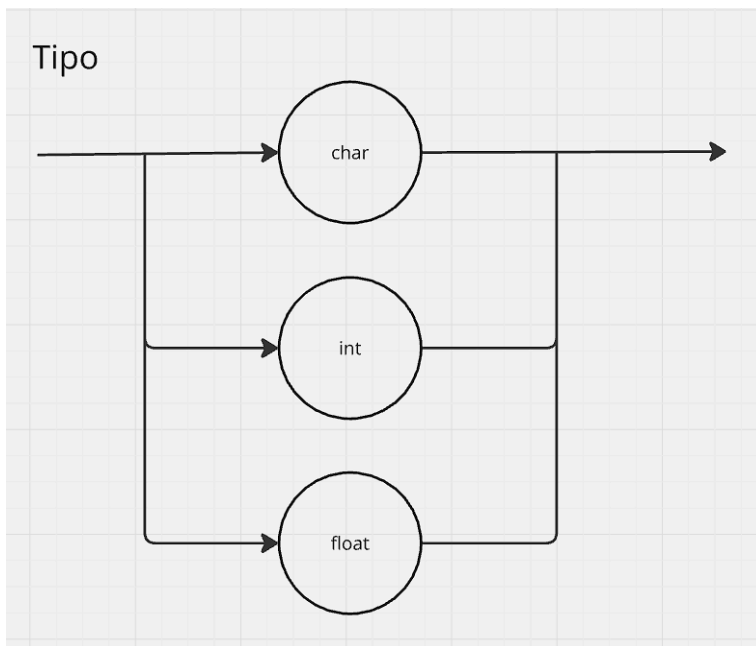


- **<declaração\_variavel> ::= <tipo> : <lista\_ids> ;**

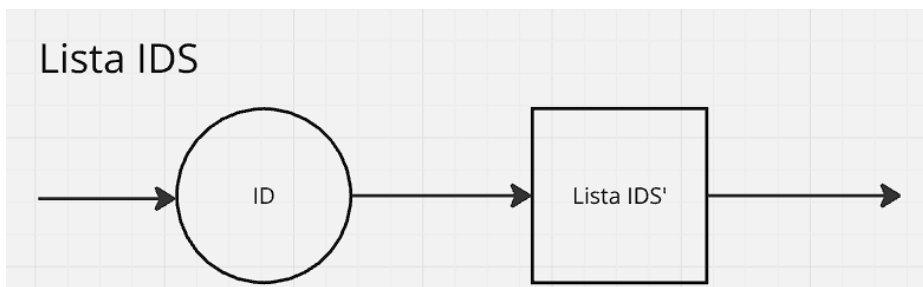
Declaração Variável



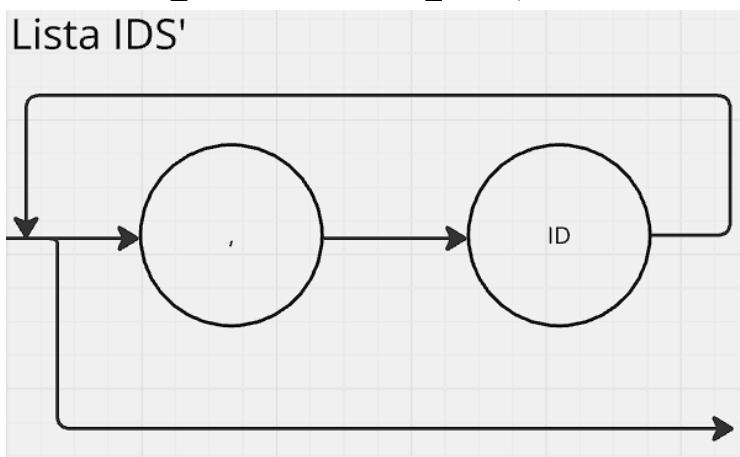
- $\langle \text{tipo} \rangle ::= \text{char} \mid \text{int} \mid \text{float}$



- $\langle \text{lista\_ids} \rangle ::= \text{ID} \langle \text{lista\_ids}' \rangle$

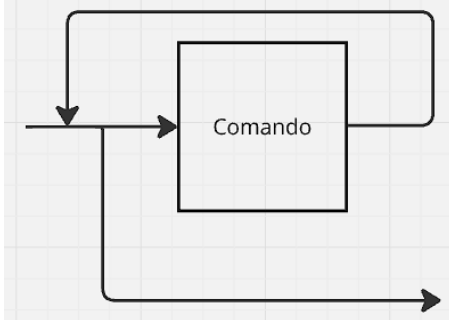


- $\langle \text{lista\_ids}' \rangle ::= , \text{ID} \langle \text{lista\_ids}' \rangle \mid \epsilon$



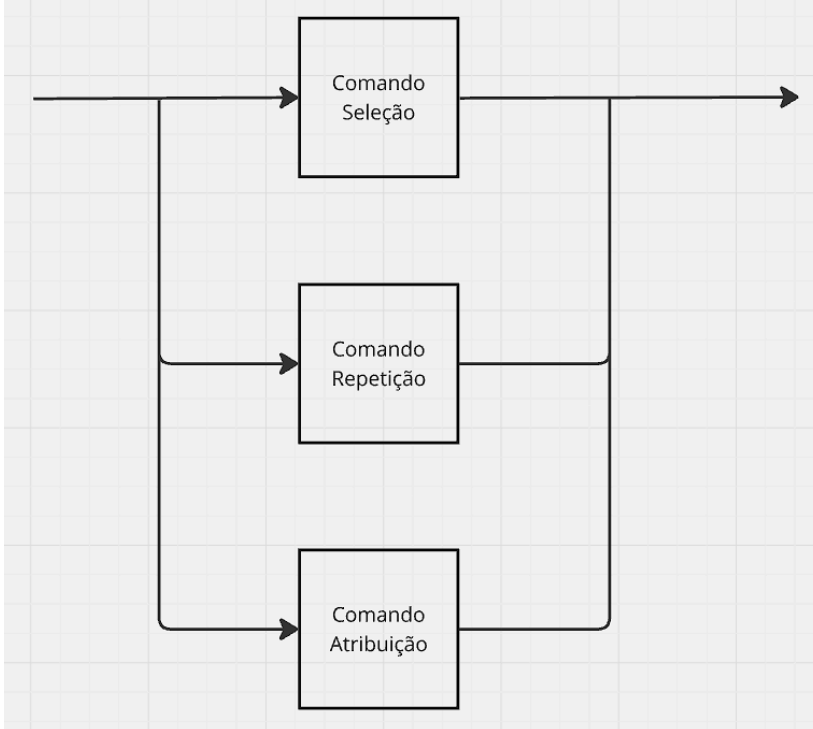
- $\langle \text{sequencia\_comandos} \rangle ::= \langle \text{comando} \rangle \langle \text{sequencia\_comandos} \rangle \mid \varepsilon$

### Sequência Comandos



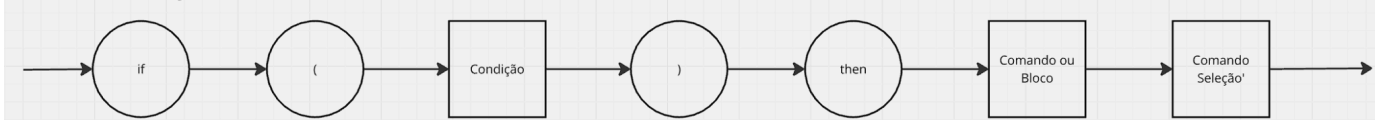
- $\langle \text{comando} \rangle ::= \langle \text{comando\_selecao} \rangle \mid \langle \text{comando\_repeticao} \rangle \mid \langle \text{comando\_atribuicao} \rangle$

### Comando



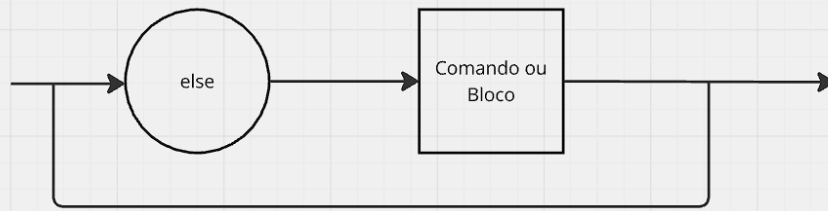
- $\langle \text{comando\_selecao} \rangle ::= \text{if } ( \langle \text{condicao} \rangle ) \text{ then } \langle \text{comando\_ou\_bloco} \rangle \langle \text{comando\_selecao}' \rangle$

### Comando Seleção



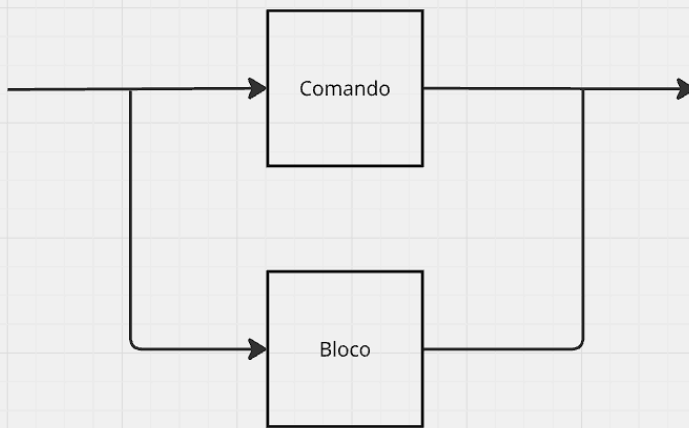
- **<comando\_selecao'> ::= else <comando\_ou\_bloco> |  $\epsilon$**

Comando Seleção'



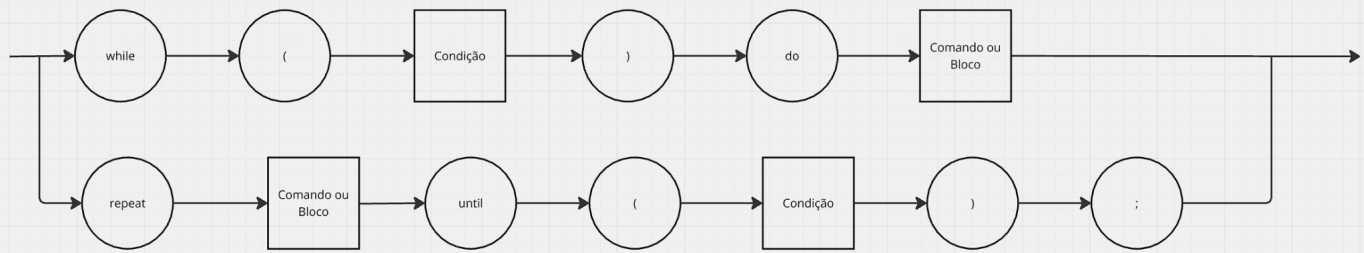
- **<comando\_ou\_bloco> ::= <comando> | <bloco>**

Comando ou Bloco



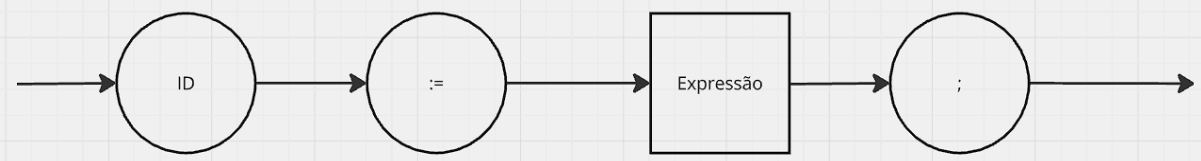
- **<comando\_repeticao> ::= while ( <condicao> ) do <comando\_ou\_bloco> | repeat <comando\_ou\_bloco> until ( <condicao> );**

Comando Repetição



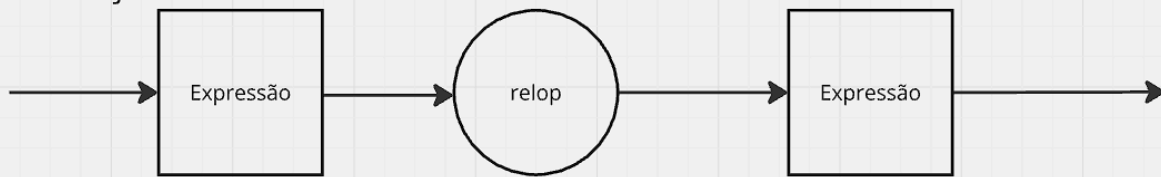
- **<comando\_atribuicao> ::= ID := <expressao> ;**

Comando Atribuição



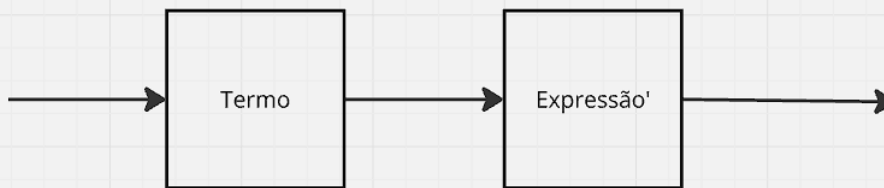
- $\langle \text{condicao} \rangle ::= \langle \text{expressão} \rangle \text{ relop } \langle \text{expressão} \rangle$

Condição



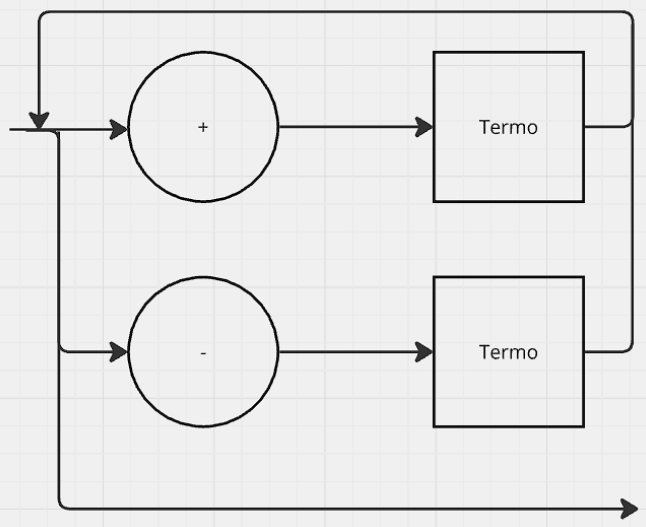
- $\langle \text{expressao} \rangle ::= \langle \text{termo} \rangle \langle \text{expressao}' \rangle$

Expressão



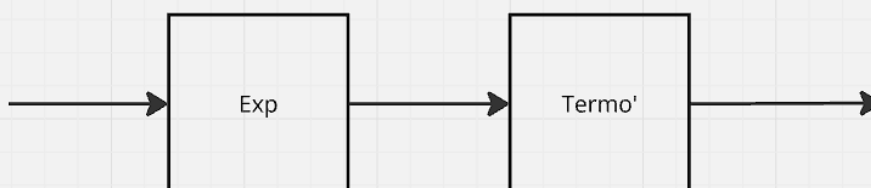
- $\langle \text{expressao}' \rangle ::= + \langle \text{termo} \rangle \langle \text{expressao}' \rangle \mid - \langle \text{termo} \rangle \langle \text{expressao}' \rangle \mid \epsilon$

Expressão'



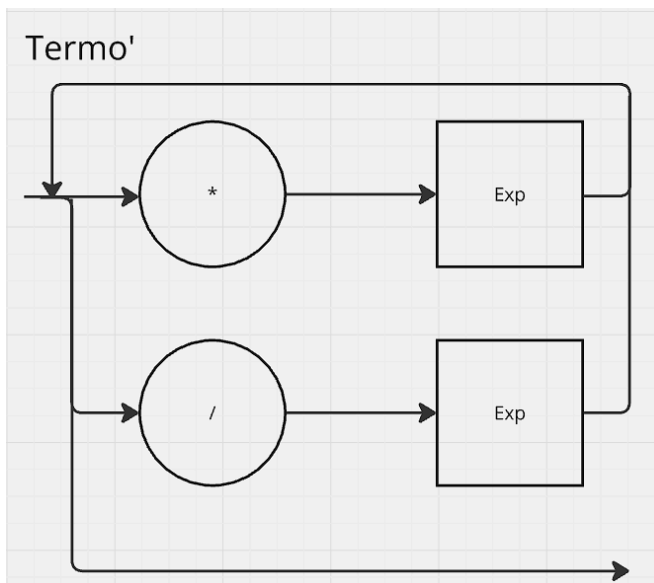
- $\langle \text{termo} \rangle ::= \langle \text{exp} \rangle \langle \text{termo}' \rangle$

Termo

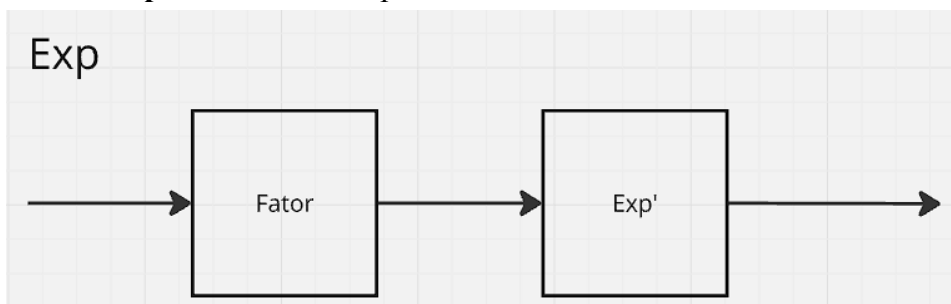




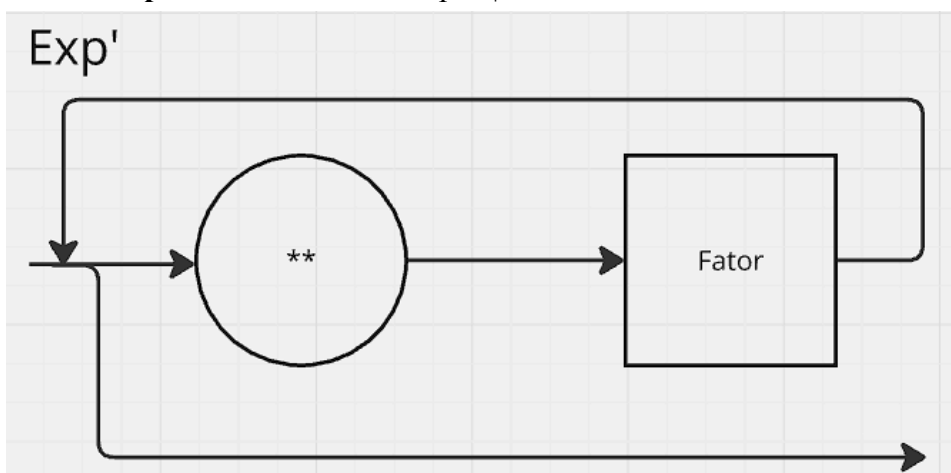
- $\langle \text{termo}' \rangle ::= * \langle \text{exp} \rangle \langle \text{termo}' \rangle \mid / \langle \text{exp} \rangle \langle \text{termo}' \rangle \mid \varepsilon$



- $\langle \text{exp} \rangle ::= \langle \text{fator} \rangle \langle \text{exp}' \rangle$



- $\langle \text{exp}' \rangle ::= ** \langle \text{fator} \rangle \langle \text{exp}' \rangle \mid \varepsilon$



- **<fator> ::= num | caractere | ID | (<expressao>)**

Fator

