

Primeira Etapa

Gramática livre de contexto

$\langle \text{Software} \rangle \rightarrow \text{programa id} \langle \text{bloco} \rangle$

$\langle \text{bloco} \rangle \rightarrow \{ \langle \text{conteudoBloco} \rangle \}$

$\langle \text{conteudoBloco} \rangle \rightarrow \langle \text{listaVariaveis} \rangle \langle \text{listaComandos} \rangle$

$\langle \text{listaVariaveis} \rangle \rightarrow \langle \text{declaracaoDeVariavel} \rangle \langle \text{listaVariaveis} \rangle | \epsilon$

$\langle \text{listaComandos} \rangle \rightarrow \langle \text{comando} \rangle \langle \text{listaComandos} \rangle | \epsilon$

$\langle \text{declaracaoDeVariavel} \rangle \rightarrow \text{TipoDeVariavel} : \langle \text{listaIds} \rangle ;$

$\langle \text{comando} \rangle \rightarrow \langle \text{cmdSelecao} \rangle | \langle \text{cmdRepeticaoWhile} \rangle | \langle \text{cmdRepeticaoDoUntil} \rangle | \langle \text{cmdAtribuicao} \rangle$

$\langle \text{listaIds} \rangle \rightarrow \text{id} \langle \text{maisDeUmID} \rangle$

$\langle \text{maisDeUmID} \rangle \rightarrow , \langle \text{listaIds} \rangle | \epsilon$

$\langle \text{cmdSelecao} \rangle \rightarrow \text{if} (\langle \text{condicao} \rangle) \text{then} \langle \text{bloco} \rangle \langle \text{comElse} \rangle$

$\langle \text{comElse} \rangle \rightarrow \text{else} \langle \text{bloco} \rangle | \epsilon$

$\langle \text{cmdRepeticaoWhile} \rangle \rightarrow \text{while} (\langle \text{condicao} \rangle) \langle \text{bloco} \rangle$

$\langle \text{cmdRepeticaoDoUntil} \rangle \rightarrow \text{do} \langle \text{bloco} \rangle \text{until} (\langle \text{condicao} \rangle) ;$

$\langle \text{cmdAtribuicao} \rangle \rightarrow \text{id} := \langle \text{expressaoAritmeticaOuConstAscii} \rangle ;$

$\langle \text{condicao} \rangle \rightarrow \langle \text{idOUConstante} \rangle \text{operadorLogico} \langle \text{idOUConstante} \rangle$

$\langle \text{idOUConstante} \rangle \rightarrow \text{id} \mid \text{ConstInt} \mid \text{ConstReal} \mid \text{ConstAscii}$

$\langle \text{expressaoAritmeticaOuConstAscii} \rangle \rightarrow \langle \text{expressaoAritmetica} \rangle \mid \text{ConstAscii}$

$\langle \text{expressaoAritmetica} \rangle \rightarrow \langle \text{termo} \rangle \langle \text{expressaoAritmetica}' \rangle$

$\langle \text{expressaoAritmetica}' \rangle \rightarrow \langle \text{addOUsub} \rangle \langle \text{termo} \rangle \langle \text{expressaoAritmetica}' \rangle | \epsilon$

$\langle \text{termo} \rangle \rightarrow \langle \text{fator} \rangle \langle \text{termo}' \rangle$

$\langle \text{termo}' \rangle \rightarrow \langle \text{multOUdiv} \rangle \langle \text{fator} \rangle \langle \text{termo}' \rangle | \epsilon$

$\langle \text{fator} \rangle \rightarrow \mathbf{id} \mid \mathbf{ConstInt} \mid \mathbf{ConstReal} \mid (\langle \text{expressaoAritmetica} \rangle)$

$\langle \text{addOUsub} \rangle \rightarrow \mathbf{add} \mid \mathbf{sub}$

$\langle \text{multOUdiv} \rangle \rightarrow \mathbf{mult} \mid \mathbf{div}$

Lexemas	Nome do Token	Valor do atributo
Qualquer ws	-	-
programa	Programa	-
if	If	-
then	Then	-
else	Else	-
while	While	-
do	Do	-
until	Until	-
Qualquer ID	Id	posição na tabela
numeros inteiros	ConstInt	posição na tabela
numeros reais	ConstReal	posição na tabela
letras e digitos da tabela ascii	ConstAscii	posição na tabela
ascii	TipoDeVariavel	Ascii
int	TipoDeVariavel	Inteiro
real	TipoDeVariavel	Real
;	PontoTerminação	-
,	Virgula	-
:	DoisPontos	-
{	AbreChaves	-
}	FechaChaves	-
(AbreParênteses	-
)	FechaParênteses	-
==	OperadorLógico	Igual
<>	OperadorLógico	Diferente
<	OperadorLógico	Menor
>	OperadorLógico	Maior
<=	OperadorLógico	Menor ou igual
>=	OperadorLógico	Maior ou igual
+	add	Soma
-	sub	Subtração
*	mult	Multiplicação
/	div	Divisão
:=	OperadorAtribuição	-

Table 1: Tabela de Tokens

Nome do Token	Expressão regular
programaPrograma	programa
If	if
Then	then
Else	else
While	while
Do	do
Until	until
PontoTerminação	;
Virgula	,
DoisPontos	:
AbreChaves	{
FechaChaves	}
AbreParênteses	(
FechaParênteses)
OperadorLógico	<> == > < <= >=
add	+
sub	-
mult	*
div	/
OperadorAtribuição	:=
TipoDeVariavel	real ascii int
Id	((a..z) (A..Z) (0..9)) ⁺
ConstInt	(+ - (0..9))(0..9) ⁺
ConstReal	(ConstInt.(0..9) ⁺)(((E e)(+ -)(0..9) ⁺) ϵ)
ConstAscii	'((a..z) (A..Z) (0..9))'
[comentário]	[[^]]

Table 2: Token e sua respectiva expressão regular

Segunda Etapa

Terceira Etapa

Simbolo	First	Follow
software	programa	\$
bloco	{	id, \$, }, if, else, while, do, until
conteudoBloco	id, TipoDeVariavel, if, while, do	}
listaVariaveis	TipoDeVariavel	id, }, if, while, do
listaComandos	id, if, while, do	}
declaracaoDeVariavel	TipoDeVariavel	id, }, TipoDeVariavel, if, while, do
comando	id if while do	id, }, if, while, do
listaIds	id	;
cmdSelecao	if	id, }, if, while, do
cmdAtribuicao	id	id, }, if, while, do
maisDeUmID	,	;
condicao	id)
condicao'	operadorLogico)
comElse	else	id, }, if, while, do
cmdRepeticaoWhile	while	id, }, if, while, do
cmdRepeticaoDoUntil	do	id, }, if, while, do
expressaoAritmeticaOuConstAscii	id, ConstInt, ConstReal, ConstAscii	;
expressaoAritmetica	id, ConstInt, ConstReal	;
expressaoAritmetica'	add, sub	;
addOUsub	add, sub	id, ConstInt, ConstReal
termo	id, ConstInt, ConstReal	add, sub
termo'	mult,div	add, sub
multOUdiv	mult,div	id, ConstInt, ConstReal
fator	id, ConstInt, ConstReal	add, sub, mult, div
idOUConstante	id, ConstInt, ConstReal, ConstAscii	; ,operadorAritimético

Table 3: **FIRST** e **FOLLOW** dos símbolos da gramática

Grafos sintáticos









