

Primeira Etapa

Gramática livre de contexto

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

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

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

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

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

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

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

$\langle \text{tipo} \rangle \rightarrow \text{int} | \text{real} | \text{ascii}$

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

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

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

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

$\langle \text{cmdRepeticao} \rangle \rightarrow \langle \text{cmdRepeticaoWhile} \rangle | \langle \text{cmdRepeticaoDoUntil} \rangle$

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

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

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

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

$\langle \text{condicao}' \rangle \rightarrow \langle \text{operadorLogico} \rangle \langle \text{condicao} \rangle | \epsilon$

$\langle \text{operadorLogico} \rangle \rightarrow == | < > | < | > | < = | > =$

$\langle \text{expressao} \rangle \rightarrow \langle \text{idOUConstante} \rangle \langle \text{expressao}' \rangle$

$\langle \text{expressao}' \rangle \rightarrow \langle \text{operadorAritimético} \rangle \langle \text{expressao} \rangle | \epsilon$

$\langle \text{operadorAritimético} \rangle \rightarrow + | - | * | /$

$\langle \text{idOUConstante} \rangle \rightarrow \mathbf{id} | \langle \text{constante} \rangle$

$\langle \text{constante} \rangle \rightarrow \mathbf{ConstInt} | \mathbf{ConstReal} | \mathbf{ConstAscii}$

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	TipoDeVariável	Ascii
int	TipoDeVariável	Inteiro
real	TipoDeVariável	Real
;	PontoTerminação	-
,	Virgula	-
:	DoisPontos	-
{	AbreChaves	-
}	FechaChaves	-
(AbreParênteses	-
)	FechaParênteses	-
==	OperadorLógico	Igual
<>	OperadorLógico	Diferente
<	OperadorLógico	Menor
>	OperadorLógico	Maior
<=	OperadorLógico	Menor ou igual
>=	OperadorLógico	Maior ou igual
+	OperadorAritmético	Soma
-	OperadorAritmético	Subtração
*	OperadorAritmético	Multiplicação
/	OperadorAritmético	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	<> == > < <= >=
OperadorAritmético	+ - * /
OperadorAtribuição	:=
TipoDeVariável	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

Simbolo	First	Follow
software	programa	\$
bloco	{	id, \$, }, int, real, ascii, if, else, while, do, until
conteudoBloco	id, int, real, ascii, if, while, do	}
listaVariaveis	int, real, ascii	id, }, int, real, ascii, if, while, do
listaComandos	id, if, while, do	id }, int, real, ascii, if, while, do
declaracaoDeVariavel	int, real, ascii	id, }, int, real, ascii, if, while, do
comando	id if while do	id, }, int, real, ascii, if, while, do
tipo	int, real, ascii	:
listaIds	id	;
cmdSelecao	if	id, }, int, real, ascii, if, while, do
cmdRepeticao	while, do	id, }, int, real, ascii, if, while, do
cmdAtribuicao	id	id, }, int, real, ascii, if, while, do
maisDeUmID	,	;
condicao	id)
condicao'	==, <> , <, >, <= , >=)
comElse	else	id, }, int, real, ascii, if, while, do
cmdRepeticaoWhile	while	id, }, int, real, ascii, if, while, do
cmdRepeticaoDoUntil	do	id, }, int, real, ascii, if, while, do
expressao	id, ConstInt, ConstReal, ConstAscii	;
expressao'	+, -, *, /	;
operadorLogico	==, <> , <, >, <= , >=	id
operadorAritimetrico	+, -, *, /	id, ConstInt, ConstReal, ConstAscii
idOUConstante	id, ConstInt, ConstReal, ConstAscii	;;, +, -, *, /
constante	ConstInt, ConstReal, ConstAscii	;;, +, -, *, /

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