**Analizador Léxico**

INE5426 - Construção de Compiladores

Alunos:

Luis Henrique Mulinari

Rafaell Valle

Matheus Villela

1. **EBNF**

grammar grama;

**grama**

: 'main' ' {' bloco\* '}'

;

**bloco**

: definicao\_variavel ';'

| atribuicaoVariavel ';'

| chamadaFuncao ';'

| definicao\_if

| definicao\_while

;

**definicao\_while**

: 'while' '(' condicao ')' '{' bloco '}'

;

**definicao\_if**

: 'if' '(' condicao ')' '{' bloco '}' ('else' '{' bloco '}')?

;

**definicao\_variavel**

: type ID

;

**type**

: 'int'

| 'char'

| 'string'

| 'float'

;

**funcao**

: type declaracaoSimbolo '(' declaracaoSimbolo (',' declaracaoSimbolo)\* ')' '{' bloco '}'

;

**declaracaoSimbolo**

: ID

;

**condicao**

: 'true'

| 'false'

| '(' (argumento) (operacaoBooleana) (argumento) ')' (operacaoBooleana condicao)?

|(argumento) (operacaoBooleana) (argumento)

;

**operacaoBooleana**

: '!='

| '=='

| '||'

| '&&'

| '~'

| '>'

| '<'

| '>='

| '<='

;

**chamadaFuncao**

: ID '(' (argumento) ? (',' (argumento) )\* ')'

;

**atribuicaoVariavel**

: ID '=' expressao

;

**expressao**

:

| argumento (operacaoAritmetica expressao)?

| '(' expressao ')'

;

**argumento**

: ID

| INT

| FLOAT

| CHAR

| STRING

| chamadaFuncao

;

**operacaoAritmetica**

: '+'

| '-'

| '\*'

| '/'

;

**ID**

: ('a'..'z'|'A'..'Z'|'\_') ('a'..'z'|'A'..'Z'|'0'..'9'|'\_')\*

;

**INT** : '0'..'9'+

;

**FLOAT**

: INT+ '.' INT+

;

**COMMENT**

: '//' ~('\n'|'\r')\* '\r'? '\n' {$channel=HIDDEN;}

| '/\*' ( options {greedy=false;} : . )\* '\*/' {$channel=HIDDEN;}

;

**STRING**

: '"' ( ESC\_SEQ | ~('\\'|'"') )\* '"'

;

**CHAR**: '\'' ( ESC\_SEQ | ~('\''|'\\') ) '\''

;

fragment

**HEX\_DIGIT** : ('0'..'9'|'a'..'f'|'A'..'F') ;

fragment

**ESC\_SEQ**

: '\\' ('b'|'t'|'n'|'f'|'r'|'\"'|'\''|'\\')

| UNICODE\_ESC

| OCTAL\_ESC

;

fragment

**OCTAL\_ESC**

: '\\' ('0'..'3') ('0'..'7') ('0'..'7')

| '\\' ('0'..'7') ('0'..'7')

| '\\' ('0'..'7')

;

fragment

**UNICODE\_ESC**

: '\\' 'u' HEX\_DIGIT HEX\_DIGIT HEX\_DIGIT HEX\_DIGIT

;

**2. Input Correto**

main {

int abc;

if (abc == func1(10, abc) {

while (true) {

func2(func1());

}

}

}

VER: inputCorreto.png

**3. Input Incorreto**

main {

int 1abc; // Identificador começando com número

if (abc == func1(10, abc { // Falta de fechamento parenteses

while (true) {

func2(func1()) // Falta ;

}

}

}

VER: inputIncorreto.png

**4. Tokens**

T\_\_15=15

T\_\_16=16

T\_\_17=17

T\_\_18=18

T\_\_19=19

T\_\_20=20

T\_\_21=21

T\_\_22=22

T\_\_23=23

T\_\_24=24

T\_\_25=25

T\_\_26=26

T\_\_27=27

T\_\_28=28

T\_\_29=29

T\_\_30=30

T\_\_31=31

T\_\_32=32

T\_\_33=33

T\_\_34=34

T\_\_35=35

T\_\_36=36

T\_\_37=37

T\_\_38=38

T\_\_39=39

T\_\_40=40

T\_\_41=41

T\_\_42=42

T\_\_43=43

ID=4

INT=5

FLOAT=6

CHAR=7

STRING=8

COMMENT=9

ESC\_SEQ=10

HEX\_DIGIT=11

UNICODE\_ESC=12

OCTAL\_ESC=13

'main '=14

'{'=15

'}'=16

';'=17

'while'=18

'('=19

')'=20

'if'=21

'else'=22

'int'=23

'char'=24

'string'=25

'float'=26

','=27

'true'=28

'false'=29

'!='=30

'=='=31

'||'=32

'&&'=33

'~'=34

'>'=35

'<'=36

'>='=37

'<='=38

'='=39

'+'=40

'-'=41

'\*'=42

'/'=43