



Projeto de Compiladores

CompilaTicX

Aluno: Mateus Costa
Professor: Roberto Fontes

Projeto final apresentado e entregue
em 10 de junho de 2019.

Taguatinga – DF

1 de junho de 19

1. Analisador Léxico (Lexer)

A próxima subseção detalha os tipos de tokens que deverão ser reconhecidos e retornados pela linguagem CompilaTicX.

Tokens

WHITE=[\b\t\r]

ASPAS = \"

ASPASIMPLES = \'

- Operadores Lógicos
 - ✓ and
 - ✓ or
 - ✓ not

- Operadores Relacionais
 - ✓ <
 - ✓ >
 - ✓ <=
 - ✓ >=
 - ✓ =
 - ✓ <>
 - ✓ :=

- Comentários
 - ✓ (">>"(.)*(.)"<<")
 - ✓ *Qualquer coisa*

- Operadores Aritméticos
 - ✓ +
 - ✓ -
 - ✓ *
 - ✓ /
 - ✓ %

- Símbolos Especiais

- ✓ (
- ✓)
- ✓ ;
- ✓ {
- ✓ }
- ✓ ,
- ✓ “ “
- ✓ ‘ ’

- /* Palavras-chave reservadas */

- ✓ if
- ✓ else
- ✓ while
- ✓ return
- ✓ float
- ✓ char
- ✓ void
- ✓ main
- ✓ prnt
- ✓ int
- ✓ proc
- ✓ var

- /* ID */

- ✓ IDENTIFICADOR = [a-z][a-zA-Z_0-9]*

- NUMERO

- ✓ [0-9]

- /* Numero Decimal */

- ✓ NUM_DEC = [0-9]*,[0-9]+ | [0-9]+,[0-9]*

2. Analisador Sintático (Parser)

terminal VAR, MINUS, VIRG, PVIRG, ID, INT, CHAR, FLOAT, PROC, LPAR, RPAR, LCHA, RCHA, SETOP, WHILE, IF, ELSE, PRINT, RETURN, PLUS, TIMES, DIV, MOD, AND, OR, NOT, EQ, DIFF, LEQ, MEQ, LTH, MTH, INTL, CHARL, FLOATL, VOID;

non terminal programa, decl_global, decl_var, decl_func, lista_idents, tipo, nome_args, bloco, lista_comandos, comando, atribuicao, iteracao, decisao, escrita, retorno, chamada_func_cmd, expressao, lista_exprs, chamada_func, expr_basica, param_formais;

precedence nonassoc ELSE;

precedence left OR, AND, EQ, DIFF, LTH, MTH, LEQ, MEQ, PLUS, MINUS, TIMES, DIV, MOD, NOT;

start with programa;

programa ::= decl_global;

decl_global ::= decl_var | decl_func | decl_global decl_var | decl_var decl_func;

decl_var ::= VAR lista_idents MINUS tipo PVIRG;

lista_idents ::= ID | lista_idents VIRG ID ;

tipo ::= INT | CHAR | FLOAT;

decl_func ::= PROC nome_args MINUS tipo bloco | PROC nome_args bloco;

nome_args ::= ID LPAR param_formais RPAR | nome_args ID LPAR param_formais RPAR |

ID LPAR RPAR | nome_args ID LPAR RPAR;

param_formais ::= ID MINUS tipo | param_formais VIRG ID MINUS tipo;

bloco ::= LCHA lista_comandos RCHA;

lista_comandos ::= comando | lista_comandos comando;

comando ::= decl_var | atribuicao | iteracao | decisao | escrita | retorno | bloco |
chamada_func_cmd;

atribuicao ::= ID SETOP expressao PVIRG;

iteracao ::= WHILE LPAR expressao RPAR comando;

decisao ::= IF LPAR expressao RPAR comando | IF LPAR expressao RPAR comando
ELSE comando;

escrita ::= PRINT LPAR lista_exprs RPAR PVIRG | PRINT LPAR RPAR PVIRG;

chamada_func_cmd ::= chamada_func PVIRG;

retorno ::= RETURN expressao PVIRG;

chamada_func ::= ID LPAR lista_exprs RPAR | chamada_func ID LPAR lista_exprs RPAR;

lista_exprs ::= expressao | lista_exprs VIRG expressao;

expressao ::= expressao PLUS expressao | expressao MINUS expressao |

expressao TIMES expressao | expressao DIV expressao | expressao MOD expressao |

expressao AND expressao | expressao OR expressao | expressao EQ expressao |

expressao DIFF expressao | expressao LEQ expressao | expressao LTH expressao |

expressao MEQ expressao | expressao MTH expressao | expr_basica;

expr_basica ::= LPAR expressao RPAR | NOT expr_basica | INTL | CHARL | FLOATL
| ID |

chamada_func;