

Apêndice A

Gramática da linguagem G-Portugol

A.1 Termos léxicos

A seguir é apresentado as convenções léxicas usadas em G-Portugol.

Regras para identificar literais numéricos

T_INT_LIT : T_OCTAL_LIT | T_HEX_LIT | T_BIN_LIT | T_DEC_LIT

T_DEC_LIT : [0-9]+

T_OCTAL_LIT : '0' ('c'|'C') [0-8]+

T_HEX_LIT : '0' ('x'|'X') [0-9a-fA-F]+

T_BIN_LIT : '0' ('b'|'B') [01]+

T_REAL_LIT : T_DEC_LIT+ '.' T_DEC_LIT+

Regras para identificar caracteres e cadeias de caracteres

T_CARAC_LIT : ''' (~(''' | '\ ') | '\ ' .)? '''

T_STRING_LIT : '"' (~('"' | '\ ' | CR | LF) | '\ ' .)* '"'

Regras para identificar comentários

SL_COMMENT : "//" [^LF]* ('\n')?

ML_COMMENT : "/*" (~('*') | '*' ~ '/')* "*/"

Regra para identificar nomes de variáveis, funções, etc.

T_IDENTIFICADOR : [a-zA-Z_] [a-zA-Z0-9_]*

A tabela [A.1](#) contém as palavras-chave padrão da linguagem G-Portugol.

A.2 Gramática

A seguir é apresentado a gramática da linguagem G-Portugol.

Palavras-chave de G-Portugol				
fim-variáveis	algoritmo	variáveis	inteiro	real
caractere	literal	lógico	início	verdadeiro
falso	fim	ou	e	não
se	senão	então	fim-se	enquanto
faça	fim-enquanto	para	de	até
fim-para	matriz	inteiros	reais	caracteres
literais	lógicos	função	retorne	passo

Tabela A.1: Palavras-chave de G-Portugol

```

algoritmo
: declaracao_algoritmo (var_decl_block)? stm_block (func_decls)* EOF
;

declaracao_algoritmo
: "algoritmo" T_IDENTIFICADOR ";"
;

var_decl_block
: "variáveis" (var_decl ";")+ "fim-variáveis"
;

var_decl
: T_IDENTIFICADOR ("," T_IDENTIFICADOR)* ":" (tp_primitivo | tp_matriz)
;

tp_primitivo
: "inteiro"
| "real"
| "caractere"
| "literal"
| "lógico"
;

tp_matriz
: "matriz" ("[" T_INT_LIT "]" )+ "de" tp_prim_pl
;

tp_prim_pl
: "inteiros"
| "reais"
| "caracteres"
| "literais"
| "lógicos"
;

stm_block
: "início" (stm_list)* "fim"
;

stm_list
: stm_attr

```

```

    | fcall ";"
    | stm_ret
    | stm_se
    | stm_enquanto
    | stm_para
    ;
stm_ret
: "retorne" expr? ";"
;

lvalue
: T_IDENTIFICADOR ("[" expr "]" ) *
;

stm_attr
: lvalue "!=" expr ";"
;

stm_se
: "se" expr "então" stm_list ("senão" stm_list)? "fim-se"
;

stm_enquanto
: "enquanto" expr "faça" stm_list "fim-enquanto"
;

stm_para
: "para" lvalue "de" expr "até" expr passo? "faça" stm_list "fim-para"
;

passo
: "passo" ("+" | "-" )? T_INT_LIT
;

expr
: expr ("ou" | "||" ) expr
| expr ("e" | "&&" ) expr
| expr "|" expr
| expr "^" expr
| expr "&" expr
| expr ("=" | "<>" ) expr
| expr (">" | ">=" | "<" | "<=" ) expr
| expr ("+" | "-" ) expr
| expr ("/" | "*" | "%" ) expr
| ("+" | "-" | "~" | "não" )? termo
;

termo
: fcall
| lvalue
| literal
| "(" expr ")"
;

```

```
fcall
: T_IDENTIFICADOR "(" fargs? ")"
;

fargs
: expr ("," expr)*
;

literal
: T_STRING_LIT
| T_INT_LIT
| T_REAL_LIT
| T_CARAC_LIT
| T_KW_VERDADEIRO
| T_KW_FALSO
;

func_decls
: "função" T_IDENTIFICADOR "(" fparams? ")" (":" tb_primitivo)?
  fvar_decl
  stm_block
;

fvar_decl
: (var_decl ";")*
;

fparams
: fparam ("," fparam)*
;

fparam
: T_IDENTIFICADOR ":" (tp_primitivo | tp_matriz)
;
```