<! compound_statement é um mau nome >

```
block                   : const_def_part type_def_part var_dec_part
proc_func_dec_part compound_statement

lcont                   :
                        | ',' INT lcont


const_def_part          :
                        | CONST const_def ';' ccont

ccont                   :
                        | const_def ';' ccont

type_def_part           :
                        | TYPE type_def ';' tcont

tcont                   :
                        | type_def ';' tcont

var_dec_part            :
                        | VAR var_dec ';' vcont

vcont                   :
                        | var_dec ';' vcont

proc_func_dec_part      :
                        | proc_dec ';'
                        | func_dec ';'

const_def               : ID '=' constant

sign                    : '+'
                        | '-'

constant                : sign INT
                        | INT
                        | sign REAL
                        | REAL
                        | ID
                        | sign ID
                        | CHAR
                        | STR

type_def                : ID = tipo <! o livro usa type_denoter em vez de tipo >

tipo                    : ID
                        | new_type

new_type                : enumerated_type
                        | subrange_type
```

```
                        | PACKED array_type
                        | array_type

enumerated_type         : '(' id_list ')'

id_list                 : ID
                        | ID ',' id_list

subrange_type           : constant DOTDOT constant

array_type              : ARRAY '['  ordinal_type acont ']' OF tipo

acont                   :
                        | ',' ordinal_type acont

ordinal_type            : enumerated_type
                        | subrange_type
                        | ID

var_dec                 : id_list ':' tipo

var_access              : ID
                        | indexed_var

indexed_var             : var_access '[' expr, expr_sequence ']'

expr_sequence           :
                        | , expr expr_sequence

expr                    : simple_expr
                        | simple_expr relation_op simple_expr

simple_expr             : term term_sequence
                        | sign term term_sequence

term                    : factor factor_sequence

term_sequence           :
                        | add_op term term_sequence

factor                  : ID
                        | var_access
                        | INT
                        | REAL
                        | CHAR
                        | STR
                        | ID actual_param_list
                        | '(' expr ')'
                        | NOT factor

factor_sequence         :
                        | mul_op factor factor_sequence

add_op                  : '+'
```

```
                        | '-'
                        | OR

mul_op                  : '*'
                        | '/'
                        | DIV
                        | MOD
                        | AND

relation_op             : '='
                        | NE
                        | '<'
                        | '>'
                        | LE
                        | GE
                        | IN

compound_statement      : BEGIN statement statement_sequence END

statement_sequence      :
                        | ';' statement statement_sequence

statement               : simple_statement
                        | structured_statement
                        | INT ':' structured_statement
                        | INT ':' simple_statement

simple_statement        :
                        | assignment_statement
                        | proc_statement
                        | goto_statement

assignment_statement    : variable_access ASSIGN expr
                        | ID ASSIGN expr

proc_statement          : ID proc_id_cont

proc_id_cont            :
                        | actual_param_list
                        | read_param_list
                        | readln_param_list
                        | write_param_list
                        | writeln_param_list


actual_param_list       : '(' actual_param actual_param_cont ')'

actual_param_cont       :
                        | ',' actual_param

actual_param            : expr
                        | variable_access
                        | ID
```

```
read_param_list         : '(' var_access var_access_sequence ')'

var_access_sequence     :
                        | ',' var_access var_access_sequence


readln_param_list       :
                        | read_param_list

write_param_list        : '(' var_access write_param_sequence ')'
                        | '(' write_param write_param_sequence ')'

write_param_sequence    :
                        | ',' write_param write_param_sequence


write_param             : expr
                        | expr ':' expr
                        | expr ':' expr ':' expr

writeln_param_list      :
                        | write_param_list

structured_statement    : compound_statement
                        | if_statement
                        | while_statement
                        | for_statement


if_statement            : IF expr THEN statement
                        | IF expr THEN statement ELSE statement

while_statement         : WHILE expr DO statement

for_statement           : FOR ID ASSIGN expr TO expr DO statement
                        | FOR ID ASSIGN expr DOWNTO expr DO statement

proc_dec                : proc_heading ';' ID
                        | PROCEDURE ID ';' block
                        | proc_heading ';' block

proc_heading            : PROCEDURE ID
                        | PROCEDURE ID formal_param_list

func_dec                : func_heading ';' ID
                        | FUNCTION ID ';' block
                        | func_heading ';' block

func_heading            : FUNCTION ID ':' ID
                        | FUNCTION ID formal_param_list ':' ID

formal_param_list       : '(' formal_param_section fcont ')'

fcont                   :
```

```
                          | ';' formal_param_section fcont

formal_param_section    : id_list ':' ID
                          | VAR id_list ':' ID
                          | proc_heading
                          | func_heading
                          | id_list ':' conformant_array_schema
                          | VAR id_list ':' conformant_array_schema

conformant_array_schema : PACKED ARRAY '[ index_type_specification ']' OF ID
                          | ARRAY '[' index_type_specification icont ']' OF ID
                          | ARRAY '[' index_type_specification icont ']' OF
conformant_array_schema

index_type_specification: ID DOTDOT ID ':' ID

program                 : program_heading ';' block '.'

program_heading         : PROGRAM ID '(' id_list ')'
                          | PROGRAM ID
```