```
(1)  program → prolog FUNC func_list_pre MAIN '(' ')' '{' EOL body '}' EOL func_list EOF
(2)  prolog → PACKAGE MAIN EOL
(3)  func_list_pre → ε
(4)  func_list_pre → FUNC_ID '(' param_list ')' func_def_type EOL FUNC func_list_pre
(5)  func_list → ε
(6)  func_list → func_def func_list
(7)  func_def → FUNC FUNC_ID '(' param_list ')' func_def_type EOL
(8)  param_list → ε
(9)  param_list → VAR_ID type next_param
(10) next_param → ε
(11) next_param → ',' VAR_ID type next_param
(12) func_def_type → func_def_void
(13) func_def_type → func_def_ret
(14) func_def_void → '{' EOL body '}'
(15) func_def_void → '(' func_def_ret
(16) func_def_ret → ')' '{' EOL body '}'
(17) func_def_ret → ret_list ')' '{' EOL body RETURN expr_list '}'
(18) ret_list → type next_ret
(19) next_ret → ε
(20) next_ret → ',' type next_ret
(21) body → ε
(22) body → command EOL body
(23) command → VAR_ID var_
(24) command → if
(25) command → cycle
(26) command → func_call
(27) var_ → var_def
(28) var_ → var_move
(29) var_def → ':=' EXPR
(30) var_move → next_id '=' expr_list
(31) next_id → ε
(32) next_id → ',' VAR_ID next_id
(33) if → IF COND '{' EOL body '}' ELSE '{' EOL body '}'
(34) cycle → FOR for_def ';' COND ';' for_move '{' EOL body '}'
(35) for_def → ε
(36) for_def → VAR_ID var_def
(37) for_move → ε
(38) for_move → VAR_ID var_move
(39) return → RETURN ret
(40) ret → ε
(41) ret → expr_list
(42) func_call → FUNC_ID '(' expr_list ')'
(43) expr_list → EXPR next_expr
(44) expr_list → func_call
(45) next_expr → ε
(46) next_expr → ',' EXPR next_expr
(47) type → INT
(48) type → FLOAT64
(49) type → STRING
```