Parsed Rules

```
 (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 → '{' EOL body '}'
(13) func_def_type → '(' func_def_ret
(14) func_def_ret → ')' '{' EOL body '}'
(15) func_def_ret → ret_list ')' '{' EOL body '}'
(16) ret_list → type next_ret
(17) next_ret → ε
(18) next_ret → ',' type next_ret
(19) body → ε
(20) body → command EOL body
(21) command → VAR_ID var_
(22) command → if
(23) command → cycle
(24) command → func_call
(25) var_ → var_def
(26) var_ → var_move
(27) var_def → ':=' EXPR
(28) var_move → next_id '=' expr_list
(29) next_id → ε
(30) next_id → ',' VAR_ID next_id
(31) if → IF COND '{' EOL body '}' ELSE '{' EOL body '}'
(32) cycle → FOR for_def ';' COND ';' for_move '{' EOL body '}'
(33) for_def → ε
(34) for_def → VAR_ID var_def
(35) for_move → ε
(36) for_move → VAR_ID var_move
(37) return → RETURN ret
(38) ret → ε
(39) ret → expr_list
(40) func_call → FUNC_ID '(' expr_list ')'
(41) expr_list → EXPR next_expr
(42) expr_list → func_call
(43) next_expr → ε
(44) next_expr → ',' EXPR next_expr
(45) type → INT
(46) type → FLOAT64
(47) type → STRING
```