```
program           -> extern_list globals_list decl_list
                      | globals_list decl_list
globals_list      -> global globals_list | eps
global            -> param ";"
extern_list       -> extern extern_list'
extern_list'      -> extern extern_list' | eps
extern            -> "extern" prototype ";"
prototype         -> type_spec IDENT "(" params ")"
decl_list         -> decl decl_list'
decl_list'        -> decl decl_list' | eps
decl              -> prototype block
type_spec         -> "void" | var_type
params            -> param_list | "void" | eps
param_list        -> param param_list'
param_list'       -> "," param param_list' | eps
param             -> var_type IDENT
var_type          -> "int" | "float" | "bool"
stmt              -> expr_stmt | block | if_stmt
                      | while_stmt | return_stmt
return_stmt       -> "return" return_stmt_tail
return_stmt_tail  -> ";" | expr ";"
if_stmt           -> "if" "(" expr ")" block else_stmt
else_stmt         -> "else" block | eps
while_stmt        -> "while" "(" expr ")" stmt
block             -> "{" local_decls stmt_list "}"
local_decls       -> local_decl local_decls | eps
local_decl        -> param ";"
stmt_list         -> stmt stmt_list | eps
expr_stmt         -> expr ";" | ";"
expr              -> IDENT "=" expr | rval
rval              -> rval1
rval1             -> rval2 rval1'
rval1'            -> "||" rval2 rval1' | eps
rval2             -> rval3 rval2'
rval2'            -> "&&" rval3 rval2' | eps
rval3             -> rval4 rval3'
rval3'            -> "==" rval4 rval3' | "!=" rval4 rval3' | eps
rval4             -> rval5 rval4'
rval4'            -> "<=" rval5 rval4' | "<" rval5 rval4'
                      | ">=" rval5 rval4' | ">" rval5 rval4' | eps
rval5             -> rval6 rval5'
rval5'            -> "+" rval6 rval5' | "-" rval6 rval5 | eps
rval6             -> rval7 rval6'
rval6'            -> "*" rval7 rval6' | "/" rval7 rval6'
                      | "%" rval7 rval6' | eps
rval7             -> "-" rval7 | "!" rval7 | "(" expr ")"
                      | IDENT | IDENT "(" args ")" | INT_LIT
                      | FLOAT_LIT | BOOL_LIT
args              -> arg_list | eps
arg_list          -> expr arg_list'
arg_list'         -> "," expr arg_list' | eps
```