CSE 302: Compiler Design — Lab 5

BX3 Grammar

⟨program⟩ ::= (⟨globalvar⟩ | ⟨function⟩ | ⟨procedure⟩ | ⟨type-abbrev⟩)*

⟨type-abbrev⟩ ::= "type" ⟨variable⟩ "=" ⟨type⟩ ";"

⟨type⟩ ::= "int64" | "bool" | ⟨type⟩ "*" | ⟨type⟩ "[" ⟨number⟩ "]" | ⟨struct-type⟩
⟨struct-type⟩ ::= "struct" "{" (⟨struct-field⟩ ("," ⟨struct-field⟩)* ","?)? "}"
⟨struct-field⟩ ::= ⟨variable⟩ ":" ⟨type⟩

⟨globalvar⟩ ::= "var" ⟨globalvar-init⟩ ("," ⟨globalvar-init⟩)* ":" ⟨type⟩ ";"
⟨globalvar-init⟩ ::= ⟨variable⟩ "=" (⟨number⟩ | ⟨bool⟩)

⟨function⟩ ::= "fun" ⟨variable⟩ "(" ⟨parameter-groups⟩? ")" ":" ⟨type⟩ ⟨block⟩

⟨procedure⟩ ::= "proc" ⟨variable⟩ "(" ⟨parameter-groups⟩? ")" ⟨block⟩

⟨parameter-groups⟩ ::= ⟨parameter-group⟩ ("," ⟨parameter-groups⟩)?
⟨parameter-group⟩ ::= ⟨variable⟩ ("," ⟨variable⟩) * ":" ⟨type⟩

⟨stmt⟩ ::= ⟨vardecl⟩ | ⟨block⟩ | ⟨expr⟩ ";" | ⟨assign⟩ | ⟨print⟩ | ⟨ifelse⟩ | ⟨while⟩
    | "return" ⟨expr⟩? ";"

⟨vardecl⟩ ::= "var" ⟨varinit⟩ ("," ⟨varinit⟩)* ":" ⟨type⟩ ";"
⟨varinit⟩ ::= ⟨variable⟩ "=" ⟨expr⟩

⟨block⟩ ::= "{" ⟨stmt⟩* "}"

⟨assign⟩ ::= ⟨assignable⟩ "=" ⟨expr⟩ ";"

⟨print⟩ ::= "print" ⟨expression⟩ ";"

⟨ifelse⟩ ::= "if" "(" ⟨expr⟩ ")" ⟨block⟩ ("else" (⟨ifelse⟩ | ⟨block⟩))?

⟨while⟩ ::= "while" "(" ⟨expr⟩ ")" ⟨block⟩

⟨expr⟩ ::= ⟨assignable⟩ | ⟨number⟩ | ⟨bool⟩ | ⟨unop⟩ ⟨expr⟩ | ⟨expr⟩ ⟨binop⟩ ⟨expr⟩
    | ⟨variable⟩ "(" (⟨expr⟩ ("," ⟨expr⟩) *)? ")"
    | "alloc" ⟨type⟩ "[" ⟨expr⟩ "]" | "null" | "&" ⟨assignable⟩
    | "(" ⟨expr⟩ ")"

⟨assignable⟩ ::= ⟨variable⟩ | "*" ⟨expr⟩ | ⟨expr⟩ "[" ⟨expr⟩ "]"
    | ⟨expr⟩ "." ⟨variable⟩ | ⟨expr⟩ "->" ⟨variable⟩

⟨variable⟩ ::= [A-Za-z_][A-Za-z0-9_]* (except keywords)
⟨number⟩ ::= -?[0-9]+ (must be in range $[-2^{63}, 2^{63})$)
⟨bool⟩ ::= "true" | "false"
⟨unop⟩ ::= "-" | "~" | "!"
⟨binop⟩ ::= "+" | "-" | "*" | "/" | "%" | "<<" | ">>" | "&" | "|" | "^"
    | "==" | "!=" | "<" | "<=" | ">" | ">=" | "&&" | "||"