AtomC - reguli sintactice

unit: (**structDef** | **fnDef** | **varDef**)* END

structDef: STRUCT ID LACC **varDef*** RACC SEMICOLON

varDef: **typeBase** ID **arrayDecl**? SEMICOLON

typeBase: TYPE_INT | TYPE_DOUBLE | TYPE_CHAR | STRUCT ID

arrayDecl: LBRACKET INT? RBRACKET

fnDef: (**typeBase** | VOID) ID
          LPAR (**fnParam** (COMMA **fnParam**)*)? RPAR
          stmCompound

fnParam: **typeBase** ID **arrayDecl**?

stm: **stmCompound**
    | IF LPAR **expr** RPAR **stm** (ELSE **stm**)?
    | WHILE LPAR **expr** RPAR **stm**
    | RETURN **expr**? SEMICOLON
    | **expr**? SEMICOLON

stmCompound: LACC (**varDef** | **stm**)* RACC

expr: **exprAssign**

exprAssign: **exprUnary** ASSIGN **exprAssign** | **exprOr**

exprOr: **exprOr** OR **exprAnd** | **exprAnd**

exprAnd: **exprAnd** AND **exprEq** | **exprEq**

exprEq: **exprEq** (EQUAL | NOTEQ) **exprRel** | **exprRel**

exprRel: **exprRel** (LESS | LESSEQ | GREATER | GREATEREQ) **exprAdd** | **exprAdd**

exprAdd: **exprAdd** (ADD | SUB) **exprMul** | **exprMul**

exprMul: **exprMul** (MUL | DIV) **exprCast** | **exprCast**

exprCast: LPAR **typeBase arrayDecl**? RPAR **exprCast** | **exprUnary**

exprUnary: (SUB | NOT) **exprUnary** | **exprPostfix**

exprPostfix: **exprPostfix** LBRACKET **expr** RBRACKET
    | **exprPostfix** DOT ID
    | **exprPrimary**

exprPrimary: ID (LPAR (**expr** (COMMA **expr**)*)? RPAR)?
    | INT | DOUBLE | CHAR | STRING | LPAR expr RPAR