**Grammar**

prog -> type main ( ) { dec\_list stmt\_list }

dec\_list -> type id\_list ;

type -> int | float | void

id\_list -> id id\_list'

id\_list' -> , id id\_list' | ^

stmt\_list -> stmt stmt\_list'

stmt\_list' -> stmt stmt\_list' | ^

stmt -> assign | read\_stmt | write\_stmt | if\_stmt

assign -> id = exp ;

exp -> term exp'

exp' -> + term exp' | - term exp' | ^

term -> factor term'

term' -> \* factor term' | / factor term' | ^

factor -> id | integer | ( exp )

read\_stmt -> read ( id\_list ) ;

write\_stmt -> print ( id\_list ) ;

if\_stmt -> if ( cndtn ) stmt else stmt | if ( cndtn ) stmt

cndtn -> exp relop exp

relop -> < | > | <= | >= | ==