Appendix A

The SIMPLE Grammar

```
all_stmts : stmtlist stop_stmt
          | stmtlist

stmtlist : stmtlist stmt
         | stmt

stmt : compstmt
     | expr ‘;’
     | IF ‘(’ condexpr ‘)’ stmt
     | IF ‘(’ condexpr ‘)’ stmt ELSE stmt
     | WHILE ‘(’ condexpr ‘)’ stmt
     | DO stmt WHILE ‘(’ condexpr ‘)’
     | FOR ‘(’exprseq ‘;’ condexpr ‘;’exprseq ‘)’ stmt
     | SWITCH ‘(’ val ‘)’ casestmts
     | ‘;’

compsmt : ‘{’ all_stmts ‘}’
        | ‘{’ ‘}’
        | ‘{’ decls all_stmts ‘}’
        | ‘{’ decls ‘}’

/** decls denotes all possible C declarations. The only difference is that**/
/** the declarations are not allowed to have initializations in them.     **/

condexpr : val
```

```
         | val relop val

exprseq : exprseq ‘,’ expr
        | expr

casestmts : ‘{’ cases default‘}’
          | ‘;’
          | ‘{’ ‘}’

cases : cases  case
      | case

case  : CASE INT_CONST‘:’ stmtlist stop_stmt

default : DEFAULT ‘:’ stmtlist stop_stmt

stop_stmt : BREAK ‘;’
          | CONTINUE ‘;’
          | RETURN ‘;’
          | RETURN val ‘;’
          | RETURN ‘(’ val ‘)’ ‘;’

expr : modify_expr
     | rhs

modify_expr : varname ‘=’ rhs
            | ‘*’ ID ‘=’ rhs

rhs : unary_expr
    | binary_expr

unary_expr : simp_expr
           | ‘*’ ID
           | ‘&’ varname
           |call_expr
           | unop val
           | ‘(’ cast ‘)’ varname
```

```
/** cast here stands for all valid C typecasts **/

simp_expr : varname
          | INT_CONST
          | FLOAT_CONST
          | STRING_CONST

call_expr : ID '(' arglist ')'

arglist : arglist ',' val
        | val
        |

unop : '+'
     | '-'
     | '!'
     | '~'

binary_expr : val binop val

binop :  relop
      | '-' | '+' | '/' | '*' | '%'
      | '&' | '|' | '<<' | '>>' | '^'

relop : '<' | '<=' | '>' | '>=' | '==' | '!='

varname : arrayref
        | compref
        | ID

arrayref : ID reflist

reflist : '[' val ']'
        | reflist '[' val ']'

val : ID
    | CONST
```

```
compref : '(' '*' ID ')' '.' idlist
        | idlist

idlist : idlist '.' ID
       | ID
```