## EECS 337 Compiler Design 2009 Fall Semester

## Homework 3

Due on Tuesday, October 13, 2009

**Objectives:** The objective of this assignment is to develop a parser. In the next assignments, you will reuse the parser that you will develop in this assignment.

**Parser generator:** In this assignment, the instructions assume the use of yacc for consistency with the book. You are encouraged to transition to bison (<u>http://www.gnu.org/software/bison/</u>).

**Parser**: Write a parser in yacc (or bison) for the source language defined by the grammar in Appendix A.1. The parser should implement the following actions.

*Printout:* The parser prints on standard output the rule that is being applied. For example, whenever the parser uses the rule  $loc \rightarrow id$ , it should print "loc->id".

*Error handling:* Design and implement a strategy for error handling. Follow the principles outlined in sections 4.1.3 and 4.1.4. Hand in a document with general principles and the design of your error handling strategy. Implement error handling in yacc.

*Symbol table:* The parser will update the symbol table with the type of each identifier. To represent the identifier type, define:

```
typedef struct id_type id_type_t;
struct id_type {
    void *type; /* The type of this variable, as per Hw2 */
    unsigned int dimension; /* Vector dimension */
    unsigned int size; /* Size along the last dimension */
    id_type_t *subsize; /* Type of the previous dimensions */
};
```

For example, a float[10][50] would have {type=&float\_var, dimension=2, size=50, subsize=address of {type=&float\_var, dimension=1, size=10, subsize=NULL} }. A scalar type uses dimension=0, size is 0, and subsize=NULL.

In assignment 2, the lexical analyzer created an entry in the symbol table for each identifier found in the source code. The key was the identifier and the value was NULL. The parser should replace the NULL value with a pointer (cast to void \*) to an  $id_type_t$  that describes the type of the identifier.

In the present assignment, the parser does not distinguish (yet) the scope of identifiers. In particular, identifiers can be inserted in the symbol table multiple times.

**Hand in**: in addition to source code, documentation, e-mails, and other material described on the class Web page, hand in the program's output on the program at the bottom of page 986.