Compiler Theory: Assignment 6

Marc Moreno Maza
University of Western Ontario
CS447b – Winter 2004

Posted: Wednesday, March 17, 2004
Due: Tuesday, April 6, 2004

1 Guidelines

Date of postage. Wednesday, March 17, 2004.

Normal date for submission. Tuesday, April 6, 2004 (no penalties).

Last date for submission. Sunday, March 11, 2004 (late penalties apply from Wednesday 7).

Due format. The assignment should be returned to the instructor by email. Your submission will have the form of a UNIX *archive* created with the UNIX command `tar`. This archive will have a structure similar to those of the example archives available on the course web site. More precisely, a `src` directory will contain your `flex/lex` and `yacc` source programs, a `test` directory will contain your test files, a `Makefile` will define the targets and a `README` will explain these targets. Moreover, your answers to the assignment questions (that are not computer programs) will be made in a file

- which may be a text file, or a `postscript` file or a `pdf` file (No other formats are allowed. Moreover, I recommend that you generate your postscript and pdf files under a recent UNIX/Linux distribution.)

- and which will be placed in a `doc` directory.

In addition, you could have a `bin` directory for binaries (as in the course example archives) and a `ps` directory for postscript versions of computer programs (this is convenient to print them!). The computer programs of this assignment are all input files for the `flex/lex` and `yacc` program, together with C program or header files. The preferred version of `flex/lex` is `flex version 2.5.4`.

Your `Makefile` must work under Linux (with a recent distribution) or under a UNIX system of the GAUL network. You can reuse the Makefile of the archive `yacc.tar.gz` from the course web site. Note that this Makefile was developed with `bash` under Red Hat 7.3 (Linux) and using GCC version 2.96.

Your `README` must specify which UNIX system you are using (Linux or SunOS) and which shell (command language interpreter) you are using (`tcsh` or `bash`). For the compilation options of `flex/lex` and `yacc` please refer to the Makefile of the archive `yacc.tar.gz`.

Late penalties. See the course outlines.

Collaboration. You are expected to do this assignment on your own without assistance from anyone else in the class.

Marking. This assignment will be marked out of 40. The allocation of marks is 5 for the first and the last activities, and 30 for the second one.

2 The Assignment

The objective is to write a compiler for the ALLCOT language, with C as the target language. Please, refer to the document titled *The* ALLCOT *Language* and to the last year assignment 3 and available on the course web site.

- The ALLCOT language has been revised and augmented. In particular, it discusses the semantics of the ALLCOT language. Also, it suggests some simplifications to the language in order to keep things simple.

- The assignment 3 of 2003 discusses translation to C for the MOOL language. The remarks and hints there apply here, to theAllcot language.

2.1 Status of your Allcot

Discuss which restrictions or extensions to the Allcot language you consider. Please, refer to the Allcot language document. Remember that for domain inheritance, you can restrict to the case where the parent and the descendant have the same attributes.

2.2 A compiler for Allcot

Using `flex/lex` and `yacc`, write a compiler for your version of Allcot.

- From every valid input Allcot file, your compiler must generate a valid and corresponding C file.
- From every non-valid input Allcot file, your compiler must report an error with a friendly error message.

2.3 Test Programs for Allcot

Update the test files of your previous assignment according to your revised version of Allcot. Therefore, you must have three *significant* test files that are valid input Allcot file. You must check that the generated C files compile and run properly.