CS 310 Homework 4
Fall 2006
DUE: November 8, 1 pm

For this homework you can use lex and yacc on zeus. Or install lex & yacc (or flex & bison) on your home computer (see the back page for more information).

1. Build a lexer to recognize the tokens in the grammar on page 101 in Sipser. The lexer needs to print the part of speech and the token to the screen for each token it reads.

       File: PUNetID_cs310Hmwk4_1.l

2. Build a lexer to recognize tokens in the grammar on page 101 of Sipser, pass the tokens to yacc to check the syntax of the input string. The parser should print out the terminals and nonterminals as the string is processed. Print an error if the string is not a proper sentence. Make sure your parser will accept more than one sentence.

Build a JFLAP file for this grammar and demonstrate, using screen shots of the parse tree JFLAP creates and the output from your parser, that your parser tool builds the same tree as JFLAP. Show at least three screenshots from JFLAP as the parse tree is built as well as the total output of your parser. Remember, yacc does a bottom up parse!

Warning: This grammar contains a conflict. You may need to change the rules slightly and specify precedence in your yacc file.

       Files: PUNetID_cs310Hmwk4_2.l
              PUNetID_cs310Hmwk4_2.y
              PUNetID_cs310Hmwk4_2.jff

****EXTRA CREDIT****
3. Build a simple integer math calculator that will produce a value from a mathematical expression. Be sure to implement precedence and grouping correctly. The operations you need to implement are +, -, and *.

Build a grammar in JFLAP to test the calculator grammar you develop. Show, using screen shots of the parse tree being built, that precedence and grouping is working correctly by parsing:
      8 + 15 * 9
      (8 + 15) * 9

        Files: PUNetID_cs310Hmwk4_3.l
               PUNetID_cs310Hmwk4_3.y
               PUNetID_cs310Hmwk4_3.jff

What do you turn in: To turn in the assignment, build a single .zip or .tar.gz file (PUNetID_cs310Hmwk4.{tar.gz|zip}) that contains all the files described above as well as a Word document (PUNetID_cs310Hmwk4.doc) that contains and explains the necessary screenshots. Also, provide a Makefile that will build the following four targets: Hmwk4_1, Hmwk4_2, Hmwk4_3, clean. Email me this file (subject: CS310Hmwk4) before 1pm on Nov 8. There is no hard copy of anything to turn in.

FLEX/BISON   LEX/YACC

You can use lex and yacc on Zeus.

Flex/Bison and/or Lex/Yacc are available for most platforms.

Mac OS X:
Flex/Bison and Lex/Yacc are part of the dev tools for Mac OS X.
http://developer.apple.com/tools/index.html

Windows:
Various ports are available for Windows, some that integrate with Visual Studio. As a standalone, Unix-like environment, Cygwin works well and contains everything you need, though you need to specify (in the devel packages) that you want gcc, flex, and bison installed.
http://cygwin.com/

Linux: Here are links for the source code for the Linux port:

ftp://prep.ai.mit.edu/pub/gnu/bison/bison-2.3.tar.gz
ftp://ftp.gnu.org/non-gnu/flex/flex-2.5.4a.tar.gz

FreeBSD:
Lex and Yacc are part of the base system and Bison is available in the devel ports collection.