COMS W4115
Programming Languages and Translators
Homework Assignment 1

Prof. Stephen A. Edwards
Columbia University

Due October 10th, 2006
at 11:00AM (beginning of class)

On-campus students: submit solution on paper; do not email.

CVN students: FAX the solutions to CVN.

Write both your name and your Columbia ID (e.g., se2007) on your solutions.

Do this assignment alone. You may consult the instructor or a TA, but not other students.

1. Scanners

   (a) Using ANTLR-like syntax, write a scanner for C's floating point numbers, as defined by Ritchie.

   > A floating constant consists of an integer part, a decimal point, a fraction part, an `e`, and an optionally signed integer exponent. The integer and fraction parts both consist of a sequence of digits. Either the integer part or the fraction part (not both) may be missing; either the decimal point or the `e` and the exponent (not both) may be missing.

   Hint: make sure your scanner accepts constants such as `1. 0.5e-15 .3e+3 .2 1e5` but not integer constants such as `42`

   (b) Draw a DFA for a scanner that recognizes and distinguishes the following set of keywords. Draw accepting states with double lines and label them with the name of the keyword they accept. Follow the definition of a DFA given in class.

   ```
   auto case char const continue
   default do double else enum if
   ifelse
   ```

   (c) Draw an NFA for a scanner that recognizes C-style multiline comments, i.e., that begin with `/*` and end at the next `*/`. Make sure your DFA correctly recognizes comments such as `/***/`, `/* * */`, and `/*/**/`. Feel free to label transitions with groups of characters (e.g., "any character except *").

   (d) Convert your comment-recognizing NFA to a DFA using the subset construction algorithm.

2. Dragon book 2ed, Exercise 3.7.3, p. 166:

   Construct nondeterministic finite automata for the following regular expressions using Algorithm 3.23 (p. 159, shown in class), then use the subset construction algorithm to construct DFAs for them using Algorithm 3.20 (p. 153, also shown in class).

   (a) $(a\,|\,b)^*$

   (b) $((\varepsilon\,|\,a)b^*)^*$

   (c) $(a\,|\,b)^*abb(a\,|\,b)^*$

   I suggest you use text to label the states of the DFA while running the subset construction algorithm. You do not have to use the graphical style in the lecture notes.

3. Using the grammar

   $$S \to (L)\,|\,a$$
   $$L \to L,S\,|\,S$$

   (a) Construct a rightmost derivation for $(a,(a,a))$ and show the handle of each right-sentinel form.

   (b) Show the steps of a shift-reduce (bottom-up) parser corresponding to this rightmost derivation.

   (c) Show the steps in the bottom-up construction of a parse tree during this shift-reduce parse.

4. Disambiguate and remove left recursion from the following grammar (i.e., show an equivalent grammar):

   $e \to e$ `>>` $e\,|\,e$ `?` e `:` $e\,|\,e$ `->` $e\,|\,$**id**

   Use C's precedence rules, i.e., the precedence of `->` is higher than that of `>>` is higher than that of `?:`. Hint: make sure your grammar can accept an expression such as **id?id:id->id?id:id** with the right precedence rules, e.g., as if it had been written **id?id:(id->id)?id:id**.