COMS W4115
Programming Languages and Translators
Homework Assignment 2

Prof. Stephen A. Edwards   Due Oct 11, 2018
Columbia University   at 11:59PM PM

Submit your assignment as a single PDF file on Courseworks. Do this assignment alone. You may consult the instructor or a TA, but not other students. You may not consult solutions from earlier semesters.

1. Write a regular expression for floating constants according to the following (after K&R):

   A floating constant consists of an integer part, a decimal point, a fraction part, an e or 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/E and the exponent (not both) may be missing.

   Hint: make sure your regular expression accepts floating constants such as `1. 0.5e-15 .3E+3 .2 1e5 3.5E-4` but not integer constants such as `42`

2. 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.

   ```
   deinit fileprivate func open operator private
   if init self Self Any as
   ```

3. Construct nondeterministic finite automata for the following regular expressions using Thompson's algorithm (Algorithm 3.23, p. 159, also 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) $a\,(a^* \mid b^*)\,a$

   (b) $(ba \mid ab)^* b$

   (c) $(a\,(b \mid \epsilon)\,b)^*$

   Number the NFA states; use the numbers to label DFA states while performing subset construction, e.g., like Figure 3.35 (p. 155).

4. Using this grammar, whose three terminals are a, b, and c,

   $$E \rightarrow a\,F\,b$$
   $$E \rightarrow c$$
   $$F \rightarrow E\,a\,F$$
   $$F \rightarrow E$$

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

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

   (c) Show the concrete parse tree that would be constructed during this shift-reduce parse.

5. Build the LR(0) automaton for the following ambiguous grammar. **if**, **else**, and **null** are terminals; the third rule indicates T may be the empty string. Indicate the state in which the shift/reduce conflict appears.

   $$S' \rightarrow S$$
   $$S \rightarrow \textbf{if}\ S\ T$$
   $$S \rightarrow \textbf{null}$$
   $$T \rightarrow$$
   $$T \rightarrow \textbf{else}\ S$$

   Check your work by running "ocamlyacc -v" on the grammar below and looking through the ".output" file. Include annotated snippets from the .output file that confirm your answer.

   ```
   %token IF ELSE NULL
   %start s
   %type <int>s

   %%

   s : IF s t      { 0 }
     | NULL        { 0 }

   t : /* empty */ { 0 }
     | ELSE s      { 0 }
   ```