

COMS W4115

Programming Languages and Translators

Homework Assignment 2

Prof. Stephen A. Edwards Due June 27, 2016
Columbia University at 11:59 PM

CVN students: submit online.
Include your name and your Columbia ID (e.g., se2007).
Do this assignment alone. You may consult the instructor
or a TA, but not other students.

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

1. Using Ocamllex-like syntax, write a scanner for C's float-
ing point numbers following the definition in K&R 2ed.

A floating constant consists of an integer part,
a decimal part, a fraction part, an e or E, an
optionally signed integer exponent and an op-
tional type suffix, one of f, F, l, or L. The in-
teger and fraction parts both consist of a se-
quence 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. The type is deter-
mined by the suffix; F or f makes it float,
L or l makes it long double, otherwise it is
double.

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

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

```
chan chanin chanout width with if end endif  
elseif
```

3. Construct nondeterministic finite automata for the fol-
lowing regular expressions using Algorithm 3.23 (p. 159,
shown in class), then use the subset construction algo-
rithm to construct DFAs for them using Algorithm 3.20
(p. 153, also shown in class).

- (a) $(ab|b)^*$
- (b) $((\epsilon|a)b)^*$
- (c) $ab(a|b)^*$

4. Using the grammar

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

- (a) Construct a rightmost derivation for $((a, a), (a, a))$
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 con-
structed 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.

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

Check your work by running "ocamlyacc -v" on the gram-
mar below and looking through the ".output" file.

```
%token IF ELSE NULL  
%start s  
%type <int>s  
  
%%  
  
s : IF s t      { 0 }  
  | NULL       { 0 }  
  
t : /* empty */ { 0 }  
  | ELSE s      { 0 }
```