Homework 1

(Due Sept. 9, Tuesday, in class for on-campus students, email by midnight for remote students)

NOTE: On-campus students must submit a hard copy with a cover page that has the student's name on it, which allows the grader to write the grade on the next page.

For remote students, the grader will print the emailed work and send the graded hard copy through the distance-learning staff. Please identify yourself clearly on the cover page.

Problem 1 (25 pts). Consider identifiers defined as follows:

An identifier may begin with a letter, denoted by **a** , or with an underscore **_**.

If the identifier begins with a letter then it may or may not be followed by an arbitrarily long combination of **a**, **_**, and a digit, denoted by **d**. If the identifier begins with an _, then it must be followed an arbitrarily long combination that contains at least an occurrence of **a** or d.

(1.1) Please write a regular expression to represent all identifiers defined above.

(1.2) Please directly draw a DFA to represent all identifiers defined above.

Problem 2 (25 pts). Consider both kinds of comments described below. One kind starts with two slashes **//** and ends with a newline, denoted by **#**. In between it can contain any characters except another newline.

The second kind is in C style. It is an arbitrary string (possibly empty) that is enclosed by a pair of **/*** and */, with no other substring of */ in the string.

For convenience, any character that is not **/**, *****, or **#**, will be denoted by a letter **a**.

(2.1) Please directly draw a DFA for the comments defined above.

(2.2) Please write a regular expression for the comments defined above.

Problem 3 (50 pts).Example 3.25 in the textbook shows a regular expression for which any equivalent DFA with contain the number of states growing at an exponential rate with the increasing length of the expression (i.e. the size of n).

We now limit ourselves to a much shorter case, (a|b)*a(a|b).

(3.1) Please follow the Thompson's construction steps to build an NFA for this regular expression. Note: this NFA will contain many more states than the graph in Figure 3.47 (with n = 2) in the textbook, because our NFA has ε-edges.

(3.2) "Simulate" the NFA in Figure 3.47 of the textbook (with n = 2) on input **aaab** . Please show the ε-closures at each step of input scanning, including the starting state.

(3.3) "Simulate" the NFA you drew for problem 3.1 above on input **aaab**. Please show the ε-closures at each step of input scanning, including the starting state. Note: this may be a little more tedious and error prone than 3.2.

(3.4) Please convert the NFA in Figure 3.47 of the textbook (with n = 2) to an equivalent DFA. Show how the state table for this DFA is built based on sets of NFA states.