

# ENGR 3520: Foundations of Computer Science

## Professor Lynn Andrea Stein

### Fall 2010

## Assignment 2

See also: [Assignment2FAQ](#) with questions I've been asked and answers I've given.

Due Tuesday 28 February on paper in class *and* by email to LAS with the clearly labeled subject line **FOCS assignment 2**. If you could name any attachments so that they indicate they're \*yours\* (e.g. [SteinAssignment2](#)), I'd be grateful, and don't forget to put your name *in* the file and *on* the piece of paper as well.

This assignment is complete and officially released (as of 9:15am Friday 24 Sept).

1. Many programs require certain conditions of their passwords, e.g., at least 6 characters, at least one upper-case letter, at least one lower-case, at least one number, at least one symbol... In this question, you're going to build FSMs to recognize some of these properties. For the purpose of this question, **a lower case letter is one of {a, b, c}, an upper case letter is one of {A, B, C}, and a number is one of {1, 2, 3}**. A character is one of {a, b, c, A, B, C, 1, 2, 3}.
  - a. Design a FSM to recognize (accept) all and only passwords containing at least four characters including at least one upper case and at least one lower case letter. If you do this in JFLAP, please include the saved automaton in your electronic turn-in so that we can run it.
  - b. Now design a system to recognize (accept) all and only passwords containing at least six characters including at least one upper case and at least one lower case letter and at least one number. You may do this by actually building a single FSM *or* you may build two (or more) FSMs and explaining carefully and rigorously how you would construct a single machine from their combination. You may wish to include sample transitions from the combined machine to illustrate your method.
2. For each of the regular expressions below, list all strings of length 5 that match the expression.
  - a.  $a^*(bb \cup c)$
  - b.  $a \cup bb^*$
  - c.  $a^*b^*$
3. Consider an NDFSM that recognizes all strings (over the alphabet {a, b}) ending in the sequence abab: The start state,  $q_0$ , has self-transitions on both a and b, and there is also a trail leading out of the start state on a to  $q_1$ , then out of  $q_1$  to  $q_2$  on b, then to  $q_3$  on a, and finally to  $q_4$  (accepting) on b. (Picture coming...). Consider the result of converting this NDFSM to a DFSM. (DO NOT use the JFLAP auto-conversion and DO NOT have your friend do the conversion, please.

Note that you may not have to build the complete conversion in order to answer this question.) In the resulting (deterministic) machine, what are the transitions out of the following states on a and on b:

- a.  $\{q_0\}$
  - b.  $\{q_0, q_1\}$
  - c.  $\{q_0, q_2, q_4\}$
  - d. Is  $\{q_0, q_2, q_3\}$  reachable in this FSM? Why or why not?
4. Are the following languages regular? If so, give a regular expression or FSM that matches all and only the language specified. If not, use the pumping lemma to show why the language is not regular, i.e., show a string of at least the pumping length for which the pumping condition ( $xy^kz$  is in the language for all  $k \geq 0$ ,  $|xy| <$  the pumping length) does not hold. Each of the following languages is over the alphabet  $\{a,b\}$ .
- a. All strings of the form  $ww$ , where  $w$  may be any string in  $(a \cup b)^*$ , i.e., the language of doubled strings.
  - b. The set of strings having the property that, in every prefix (i.e., every substring starting all the way at the beginning) the number of as and the number of bs does not differ by more than 2.
  - c. All strings  $\{a^n \mid n \text{ is a power of } 2\}$
  - d. All strings that contain the substring  $aa$  exactly in the middle.