Homework 3: NFAs and regular expressions

CSE 30151 Spring 2016

Due 2016/02/02 at 11:55pm

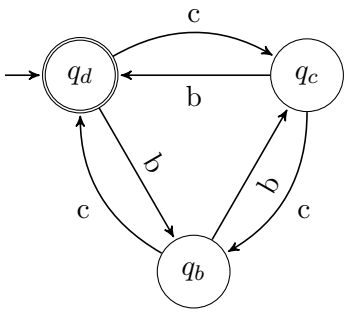
Instructions

- You can prepare your solutions however you like, but you must submit them as a single PDF file.
- Please name your PDF as follows:
  - If you're making a complete submission, name your PDF file `netid-hw3.pdf`, where `netid` is replaced with your NetID.
  - If you're submitting some problems now and want to submit other problems later, name your PDF file `netid-hw3-1234.pdf`, where `1234` is replaced with the problems you are submitting at this time.
  - If you use the same name twice, only the most recent version will be graded!
- Submit your PDF file in Sakai. Don't forget to click the Submit (or Resubmit) button!

Problems

1. **Regular expressions to NFAs.** Convert the regular expression $(- \cup \varepsilon)\mathtt{1}(\mathtt{0} \cup \mathtt{1})^*$ to an NFA using the construction of Lemma 1.55, showing each step.
2. **Complementation.**
   (a) In class we saw that if L is recognized by a DFA, then by flipping accept and reject states, we obtain a DFA that recognizes L^C. Explain why this construction doesn't work on NFAs [Exercise 1.14b].
   (b) In class we discussed how a regular expression n symbols long converts to a NFA with $O(n)$ states (the details of this proof are not important). Imagine adding a complement operator to regular expressions, and converting a regular expression (with complement) n symbols long. How many states would the resulting finite automaton have? Use big-O notation and briefly explain.

3. **The state elimination algorithm.** Convert the following finite automaton into an equivalent regular expression using the construction in Lemma 1.60, showing each step. Please eliminate states in the following order: q_b, q_c, q_d.



4. **The regular operations.** In this question we will show that all three of the regular operations are necessary. Prove that there is a language that is regular but can *not* be described by a regular expression...

   (a) that uses only union and concatenation (no Kleene star)

   (b) that uses only union and Kleene star (no concatenation)

   (c) that uses only concatenation and Kleene star (no union)