CS/Math 387 Homework for Wednesday, Week 2

Problem 1. Fix the alphabet be $\Sigma = \{0, 1\}$. Construct DFAs A and B such that $L(A)$ is the language of all words with length divisible by 2 and $L(B)$ is the language of all words with length divisible by 3. Use the Cartesian product construction from the text to create a DFA C such that $L(C) = L(A) \cup L(B)$, labeling states using elements from the relevant Cartesian product.

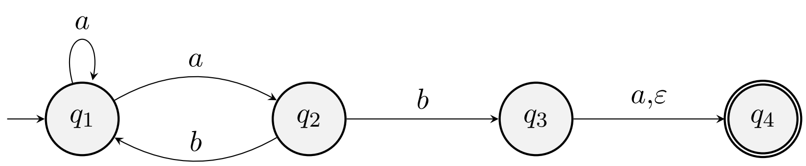
Problem 2. Create DFAs for the listed languages on the alphabet $\Sigma = \{0, 1\}$.

(a) Words with length divisible by either 2 or 3 but not both.

(b) Nonempty words whose beginning and ending characters differ.

Problem 3. Create an NFA accepting the language A of words ending in an odd number of 1s and an NFA accepting exactly the language B of words ending in 0. Combine these using the technique from the lecture to create an NFA recognizing the concatenation AB.

Problem 4. Convert the following NFA into a DFA using the method from our text:



Label the states with states describing subsets of the original set of states, omitting states corresponding to unreachable states. Each state in your DFA should have arrows for a and for b.

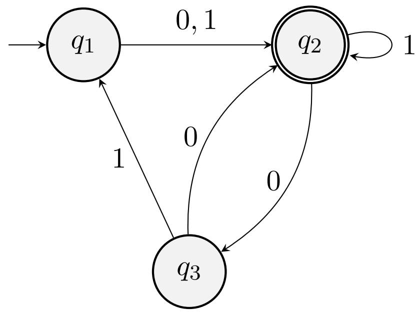
Problem 5. Find regular expressions for the following languages on $\Sigma = \{0, 1\}$:

(a) Words of even length with exactly two 1s. (Provide an explanation.)

(b) Words not containing the substring 011. (Hint: $0^*(10^+)^*1^*$ gives words not containing the substring 110.) (No explanation needed if your solution is a slight modification of the hint's solution.)

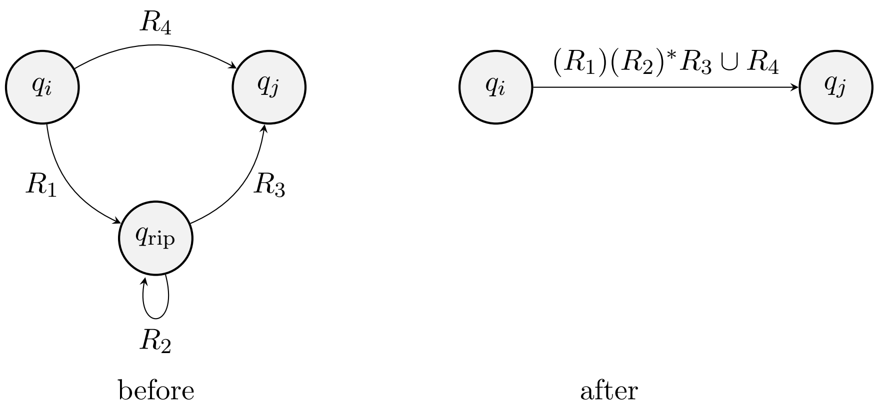
Problem 6. Create a DFA with no more than five states that excepts the language consisting of all words not containing the substring 011.

Problem 7. Let A be the following DFA:



(a) Convert A into an NFA having (i) an initial states with no transitions leading into it, and (ii) a single accept state having no transitions of it. The resulting NFA should have five states, including the initial state q_0 and the unique accepting state q_4.

(b) Recall the general prescription for ripping out a state given in the text:



Letting $q_i = q_2$, $q_j = q_1$, and $q_{\text{rip}} = q_3$, identify R_1, R_2, R_3 and R_4. What regular expression labels the transition from q_1 to q_2 after ripping out q_3 (and making any trivial simplification of the regular expression)?

(c) Repeat the previous part with $q_i = q_j = q_2$, and $q_{\text{rip}} = q_3$

(d) Rip out state q_3, and draw the resulting GNFA. The transitions will be labeled by regular expressions.

(e) Rip out state q_2, and draw the resulting GNFA.

(f) Finally, rip out state q_1 and draw the resulting GNFA having two states.

(g) What is the resulting regular expression for the language accepted by A?

PROBLEM 8. (Bonus problem for glory and fame!) The text describes a method for converting an NFA with n states into a DFA with 2^n states. Show that this bound is roughly tight. Specifically, show that for every n there exists a language that can be recognized with an $(n+1)$-state NFA but cannot be recognized by a DFA with fewer than 2^n states.