1. Apply the state elimination procedure described in class to convert the finite automata (c) from Problem 2.3.7 on Page 84 of Lewis and Papadimitriou into a regular expression. Simplify the resulting regular expression as much as you can.

2. Write down regular expressions for each of the following languages over the alphabet \{0, 1\}.
   (a) \(L_1 = \{w \mid w\) starts with 0 and has odd length, or starts with 1 and has even length\}.
   (b) \(L_2 = \{w \mid w\) is any string except 00 or 000\}.
   (c) \(L_3 = \{w \mid w\) does not contain the substring 110\}.

3. Lewis and Papadimitriou, Problem 1.8.4.
   (You must give a regular expressions based proof to receive credit, i.e., do not resort to a DFA/NFA construction to prove regularity of \(L'\).)

4. Let \(r\) and \(s\) be regular expressions where the language represented by \(r\) does not contain the empty string \(\epsilon\). Consider the equation \(X = r \circ X \cup s\) (where \(\circ\) stands for concatenation of regular expressions, and \(\cup\) for union) with unknown variable \(X\). Find a solution (namely, a regular expression) for \(X\) that satisfies the above equation and prove that this solution is unique. (Comment: This question is harder than the others!)

5. (a) By first constructing an NFA and then adding the necessary backward transitions, construct the DFA useful for determining whether the pattern \(aababaabaaa\) occurs in a string over the alphabet \{a, b\}.
   (b) Lewis and Papadimitriou, Problem 2.6.3, Part (c).
   (c) (Just for fun; no need to turn anything in for this part) Think about why such an NFA as in (b) above is good to have; this is actually Part (d) of Problem 2.6.3. (Again, you don’t have to turn in a solution to this part.)