There are three problems, labeled A–C, to be handed in on paper. There are also two optional problems D–E, which you are not required to hand in, but which you might want to solve as part of your learning/studying process.

A. Working over $\Sigma = \{a, b, c, d\}$, draw an NFA that recognizes the same language as the regular expression

$$(ad \cup b \cup c)^*(dda)^* \cup ac.$$

Use the algorithm described in class and in our textbook — rather than, say, guessing.

B. Exercise 1.21b (which is about converting a three-state DFA to a regular expression). Again, use the algorithm.

Whenever you store data on a computer or transmit it between computers, there is a chance that disturbances in the hardware (electrical faults, defects in fiber optic cables, etc.) will introduce errors into your data. To handle this issue, there are error-correction protocols operating invisibly everywhere. Hamming distance is an important concept in error correction. For any two bit strings w and x, the *Hamming distance* $H(w, x)$ is defined as follows. If $|w| \neq |x|$, then $H(w, x) = \infty$. If $|w| = |x|$, then $H(w, x)$ is the number of bits in which w and x differ. For example, $H(00010, 10111) = 3$. For any set A of bit strings, define $N_2(A)$ to be the set of bit strings within Hamming distance 2 of A:

$$N_2(A) = \{w : \exists x \in A \text{ such that } H(w, x) \leq 2\}.$$

C. Prove that if $A \subseteq \{0, 1\}^*$ is regular, then so is $N_2(A)$. (Hint: If $A = L(M)$, where M is a DFA with state set Q, then construct an NFA with state set $Q \times \{0, 1, 2\}$.)

You are not required to hand in solutions to the two problems below.

D. Prove that if A is a regular language, then the reverse $\text{rev}(A)$ is also regular.

E. Exercise 1.23 (about BB^+).