Sipser, Section 1.1/1.2: Regular Languages, NFAs Part 1

Reading: Chapter 1.2 up to page 54.

Note the following typo in the book. Lines 5–6 in p. 48 should read:
State $q_1$ has one exiting arrow for 0 [not “for 1”], but it has two for 1 [not “for 0”].

Homework Assignment 5
Due: November 2, in class

Homeworks are due at the beginning of class on the due date. Late homeworks will not be graded for credit, but I will give comments and feedback on them.

1. Give a formal description for the NFA of Figure 1.20 (p. 53). (Pay attention to how it differs from the formal description for deterministic automata!)

2. Do all parts of Sipser Exercise 1.5 (p. 84) except for part e.
   The expression $0^*$ in part f. means any string consisting of zero or more 0s, and nothing else: $\{ \varepsilon, 0, 00, 000, \ldots \}$.

3. This problem is based on concepts from the section The Regular Operations (pp. 44–47). All automata involved are deterministic (DFAs), not NFAs! The alphabet is $\{0, 1\}$. This material was also presented in class, so refer to your notes.
   a. Write a DFA that recognizes the language $L_1 = \{ w \mid w \text{ contains an odd number of } 0\}$.
   b. Write an automaton that recognizes the language $L_2 = \{ w \mid w \text{ contains exactly one } 1 \}$.
   c. Use the construction of Theorem 1.12 to write an automaton that recognizes the language $L_1 \cup L_2$ (any string that contains an odd number of zeros or exactly one 1). (You must use the construction—do not try to write an automaton from scratch!)
   d. Footnote 3 on p. 46 of Sipser explains (rather succinctly!) how the construction of Theorem 1.12 can be modified to construct the intersection of two regular languages. Use the modified construction (refer to your class notes also) to construct an automaton that recognizes $L_1 \cap L_2$ (any string that contains an odd number of 0s and exactly one 1).