Do Exercises 1.4(c,e), 1.5(d,f), 1.6(c,l), 1.7(b,c), 1.16(b).

Here are the exercises written out for the purpose of comparing your book version’s exercises with mine. (Note: “state diagram” is the same as “transition diagram.”) You must do the exercises as worded below.

Exercise 1.4: Each of the following languages is the intersection of two simpler languages. In each part, construct DFAs for the simpler languages, then combine them using the construction discussed in footnote 3 (page 46) to give the state diagram of a DFA for the language given. In all parts, \( \Sigma = \{a, b\} \).

c. \( \{w \mid w \text{ has an even number of } a \text{'s and one or two } b \text{'s}\} \)
e. \( \{w \mid w \text{ starts with an } a \text{ and has at most one } b\} \)

Exercise 1.5: Each of the following languages is the complement of a simpler language. In each part, construct a DFA for the simpler language, then use it to give the state diagram of a DFA for the language given. In all parts, \( \Sigma = \{a, b\} \).

d. \( \{w \mid w \text{ is any string not in } a^*b^* \} \)
f. \( \{w \mid w \text{ is any string not in } a^* \cup b^* \} \)

Exercise 1.6: Give state diagrams of DFAs recognizing the following languages. In all parts, the alphabet is \( \{0, 1\} \).

c. \( \{w \mid w \text{ contains the substring } 0101 \text{ (i.e., } w = x0101y \text{ for some } x \text{ and } y\} \)
l. \( \{w \mid w \text{ contains an even number of } 0 \text{'s or contains exactly two } 1 \text{'s}\} \)

Exercise 1.7: Give state diagrams of NFAs with the specified number of states recognizing each of the following languages. In all parts, the alphabet is \( \{0, 1\} \).

b. The language of Exercise 1.6c with five states
c. The language of Exercise 1.6l with six states
Exercise 1.16: Use the construction given in Theorem 1.39 to convert the following two nondeterministic finite automata to equivalent deterministic finite automata.

b. [Given in tabular form:]

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>→ 1</td>
<td>3</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>→ 2</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>→ 3</td>
<td>2</td>
<td>{2,3}</td>
<td>0</td>
</tr>
</tbody>
</table>