1.6 I actually meant to assign 1.7! The examples in 1.6 don't especially lead themselves to more efficient nondeterministic solutions. Although it's not officially assigned you should definitely do 1.7.

1.60

$$Q = \{0, 1, \ldots, k\}$$
$$\Sigma = \{a, b\}$$
$$q_0 = 0$$
$$F = \{k\}$$
$$\delta(0, b) = \{0\}$$
$$\delta(0, a) = \{0, 1\}$$
$$\delta(i, a) = \delta(i, b) = \{i + 1\} \text{ for } 0 < i < k.$$
1.16

(a)

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

\( \emptyset \emptyset \emptyset \)

It's not obvious that the original NFA accepts all strings that start with a, but it does!

(b) e-closures:

1. \{1, 2\}
2. \{2\}
3. \{3\}

State transitions

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{3}</td>
<td>\emptyset</td>
</tr>
<tr>
<td>2</td>
<td>{4, 2}</td>
<td>\emptyset</td>
</tr>
<tr>
<td>3</td>
<td>{2}</td>
<td>{2, 3}</td>
</tr>
<tr>
<td>{4, 2}</td>
<td>{4, 2, 3}</td>
<td>\emptyset</td>
</tr>
<tr>
<td>{4, 3}</td>
<td>{4, 2, 3}</td>
<td>{2, 3}</td>
</tr>
<tr>
<td>{2, 3}</td>
<td>{1, 2}</td>
<td>{2, 3}</td>
</tr>
<tr>
<td>\emptyset</td>
<td>\emptyset</td>
<td>\emptyset</td>
</tr>
</tbody>
</table>

\( a \quad b \quad a,b \)
1.61 Suppose $M$ is a DFA that recognizes $C_k$. Let $v_1, v_2$ be two strings of length $k$ with $v_1 \neq v_2$. Let $q_1, q_2$ be the states of $M$ that you arrive at, starting from the initial state and reading $v_1, v_2$, respectively. Since the two strings are different, there is some position at which $v_1$ contains $a$ and $v_2$ contains $b$, or vice-versa:

$$v_1 = xay,$$
$$v_2 = x'y'y', \text{ with } 1x1 = 1x'y'1$$

Let $z$ be any string of length $k - 1y1 - 1$. Then $v_1 z \in C_k$, $v_2 z \notin C_k$. Thus $v_1 z$ and $v_2 z$ lead from the initial state to different states $p_1, p_2 \notin M$. You get to $p_1$ by starting in $q_1$ and reading $z$. You get to $p_2$ by starting in $q_2$ and reading $z$. 
So \( q_1 \neq q_2 \).

Thus the number of states must be at least as large as the number of strings of length \( k \), since we just proved that distinct strings of length \( k \) lead to distinct states from \( q_0 \).

So the number of states of \( M \geq 2^k \).

1.43

Start with an automaton \((Q, \Sigma, \delta, q_0, F)\) for \(A\). (Assume \( A \) is a DFA.)

We make a second copy of \(M\).

For every edge in \( M \) labeled by a letter \( \alpha \) from state \( q_1 \) to \( q_2 \), add an edge labeled \( \alpha \) from \( q_1 \) to the copy of \( q_2 \). The initial state of the new device is the original initial state. The accepting states are the copies of the original accepting state.
Formally:

\[ M' = (Q \times \{0,1,3\}, \Sigma, S', (q_0, 0), F \times \{0,1\}) \]

where

\[ S'(q, 0, a) = \{(s(q, a), 0)\} \]

\[ S'(q, 1, a) = \{(s(q, a), 1)\} \]

\[ S'(q, 0, \varepsilon) = \bigcup_{a \in \Sigma} \{(s(q, a), 1)\} \]

Do you see how this works? In order for a path to be successful in \( M' \), it must cross over from \( Q \times \{0\} \) to the copy \( Q \times \{1\} \). The cross-over point corresponds to a dropped-out letter in a successful path of the original device.

1.17 We did this in class. When constructing the NFA in (a), it's advisable to short-cut some of the steps in the construction outlined in the text, lest we get too many \( \varepsilon \)-transitions.
In class, I used something like

\[
\begin{array}{cccc}
& 0 & 1 & \\
\downarrow & \downarrow & \downarrow & \\
1 & \{2\} & \emptyset & \\
2 & \{3\} & \{4, 33\} & \\
3 & \{4, 43\} & \emptyset & \\
4 & \emptyset & \emptyset & \\
5 & \emptyset & \{1, 6\} & \\
6 & \emptyset & \emptyset & \\
\emptyset & \emptyset & \emptyset & \\
\{1, 3\} & \{2, 43\} & \emptyset & \\
\{2, 4\} & \emptyset & \{4, 63\} & \\
\{4, 3\} & \{2, 5\} & \{1, 3\} & \\
\{1, 6\} & \{2\} & \emptyset & \\
\{2, 5\} & \{5\} & \{1, 3, 6\} & \\
\{1, 3, 6\} & \{1, 2, 43\} & \emptyset & \\
\end{array}
\]
1.18 I left off f-h, as we worked these in class. f) seems to require brilliant insight or special techniques which were presented later. Note correct answers are not Unique!

(a) \(1(001)^*0\)
(b) \(0^*10^*10^*1(001)^*\)
(c) \((001)^*0101(001)^*\)
(d) \((001)(001)0(001)^*\)
(e) \(0((001)(001))^* U 1(001)(001)(001)^*\)
(f) \((1(001)^*)(001 U 3)\)
(g) \(00^*100^* U 000^*10^* U 0^*1000^* U 000^*\)
(h) \(\emptyset\)
(i) \((1^*01^*01^*)^* U 0^*10^*10^*\)
(j) \((001)(001)^*\)
Again, correct answers are not unique.

In
\[ a^*b^* \]
\[ a(ba)^*b \]
\[ a^*ub^* \]
\[ (aaa)^* \]
\[ abu^*b^* \]
\[ ababvbab \]
\[ (\epsilon^*a)^*b \]
\[ (\epsilon^*uvu^*b)^* \]

Out
\[ ba, bba \]
\[ b, a \]
\[ ab, ba \]
\[ a, \epsilon \]
\[ aa, ba \]
\[ \epsilon, b. \]