Exercise 2.1.1

A deterministic finite automaton $M$ accepts the empty string (i.e., $e \in L(M)$) if and only if its initial state is a final state.

Exercise 2.1.3

(a) The following state diagram represents an automaton that accepts those strings $w \in \{a, b\}^*$ in which each $a$ is immediately preceded by $a\ b$.

(c) The following automaton accepts those strings $w \in \{a, b\}^*$ that contain neither $aa$ nor $bb$ as a substring.
Exercise 2.1.5

(a) (iii) The following 2-tape finite automaton accepts the set \{ (a^n b, a^n b^m) : n, m \geq 0 \}.

(a) (iv) The following 2-tape finite automaton accepts the set \{ (a^n b, a^m b^n) : n, m \geq 0 \}. (Not all transitions to a dead state are shown.)
Exercise 2.2.3

(a) A possible automaton for accepting the language $(ab)^*(ba)^* \cup aa^*$ is:

Exercise 2.3.3

Let $M_1 = (K_1, \Sigma, \delta_1, s_1, F_1)$ and $M_2 = (K_2, \Sigma, \delta_2, s_2, F_2)$ be deterministic finite automata. We construct a finite automaton $M$ that accepts the intersection $L(M_1) \cap L(M_2)$ of the languages accepted by $M_1$ and $M_2$, respectively.

The automaton $M$ is designed to simulate the operation of both $M_1$ and $M_2$. More specifically, its states are pairs, where the first component reflects $M_1$ and the second component $M_2$.

Formally, let $K$ be $K_1 \times K_2$; let $\Delta$ be the set of all triples $((q_1, q_2), \sigma, (q'_1, q'_2))$ in $K \times \Sigma \times K$, such that $\delta_1(q_1, \sigma) = q'_1$ and $\delta_2(q_2, \sigma) = q'_2$; and let $F$ be the set $F_1 \times F_2$. The finite automaton $M = (K, \Sigma, \Delta, (s_1, s_2), F)$ accepts $L(M_1) \cap L(M_2)$. 
Exercise 2.3.6

Let $M = (K, \Sigma, \delta, s, F)$ be a deterministic finite automaton that accepts the language $L$.

(a) Let $M'$ be the finite automaton $(K, \Sigma, \delta, s, F')$, where

$$ F' = \{ p \in K : (p, w) \vdash_M^* (f, e) \text{ for some } f \in F \text{ and } w \in \Sigma^* \}. $$

That is, $M'$ differs from $M$ only in that its final states are all those states from which a final state of $M$ can be reached. The automaton $M'$ accepts the set of prefixes of (strings in) $L$.

(b) Let $M'$ be the finite automaton $(K, \Sigma, \Delta, s, F)$, where

$$ \Delta = \{ (p, \sigma, q) : \delta(p, \sigma) = q \} \cup \{ (s, e, q) : q \in K \text{ and } (s, w) \vdash_M^* (q, e) \text{ for some } w \in \Sigma^* \}. $$

Intuitively, $M'$ works as $M$, but computations may begin at any state that can be reached from $s$ by $M$. The automaton $M'$ accepts the set of suffixes of (strings in) $L$.

(d,e) Let $M'$ be the finite automaton $(K, \Sigma, \delta, s, F')$, where

$$ F' = \{ q : (q, w) \vdash_M^* (q', e) \text{ for some } w \in L' \text{ and some } q' \in F \}. $$

The automaton $M'$ accepts the right quotient $L/L'$ of $L$ by $L'$.

Note that this definition is not effective. A suitable set $F'$ exists, but it may not be possible to actually compute it. The difference between parts (d) and (e) is that for part (d) one can show how to construct $F'$.

(g) Let $M'$ be the finite automaton $(K \cup \{ s_0 \}, \Sigma, \Delta, s_0, \{ s \})$, where $s_0$ is a new state not contained in $K$ and

$$ \Delta = \{ (q, \sigma, p) : \delta(p, \sigma) = q \} \cup \{ (s_0, e, q) : q \in F \}. $$

Informally, we obtain $M'$ from $M$ by reversing all transitions and switching initial and final states. (We need to introduce a new initial state for $M'$ as $M$ may have more than one final state.) The automaton $M'$ accepts the set of all reverse strings of $L$.

Exercise 2.4.5

(b) We use the Pumping Theorem to prove that the language $L = \{ ww : w \in \{ a, b \}^* \}$ is not regular.
The proof is by contradiction. Suppose \( L \) is regular. Then by the Pumping Theorem there exists an integer \( n \geq 1 \) such that every string \( ww \in L \), with \( |ww| \geq n \), can be written as \( ww = xyz \), where \( |xy| \leq n \), \( y \neq e \), and \( xy^iz \in L \), for each \( i \geq 0 \).

Take the string \( w = a^n b \). Since \( |ww| \geq n \), the string \( ww \) can be written as \( xyz \), for suitable strings \( x \), \( y \), and \( z \) as described above. Since \( |xy| \leq n \), the string \( xy \) consists of \( a \)'s only. Therefore \( xy^yz \) is a string of the form \( a^{k+i} b a^k b \), where \( m > 0 \). This string is clearly not of the form \( uu \), for any string \( u \), which contradicts that it is an element of \( L \). We conclude that \( L \) is not a regular language.

Exercise 2.4.7

Let \( M = (K, \Sigma, \delta, s, F) \) be a deterministic finite automaton and let \( n \) be the number of states of \( M \).

First observe that if a string \( w \) is accepted by \( M \) and \( n \leq |w| \), then, by the same arguments as in the proof of Theorem 2.4.1, \( w \) can be written as \( xyz \), for suitable strings \( x \), \( y \), and \( z \) as described above. Consequently, the automaton \( M \) accepts \( xy^iz \), for each \( i \geq 0 \).

An immediate consequence of this observation is that if \( M \) accepts any string of length greater than or equal to \( |K| \), then it accepts infinitely many strings.

On the other hand, if \( M \) accepts infinitely many strings, then it must accept some strings of length greater than or equal to \( |K| \). Let \( w \) be any shortest such string. By the above observation \( w \) can be written as \( xyz \), for suitable strings \( x \), \( y \), and \( z \), such that \( 1 \leq |y| \leq |K| \) and \( M \) accepts \( xz \). Since the string \( xz \) is shorter than \( w \) we may infer that \( |xz| < |K| \) (for otherwise \( w = xyz \) could not be a shortest string of length greater than or equal to \( |K| \) that is accepted by \( M \)) and hence \( |K| \leq |w| < 2|K| \).

Exercise 2.4.8

(a) A subset of a regular language need not be regular. For example, the set \( \{a, b\}^* \) is regular, but its subset \( \{a^ib^i : i \geq 0\} \) is nonregular.

(c) If \( L \) is regular, so is its complement \( \Sigma^*-L \). Thus, the concatenation of the two sets, \( L(\Sigma^*-L) = \{xy : x \in L \text{ and } y \notin L\} \), is regular as well.
(d) The Pumping Theorem can be used to show that the set \( \{w : w = w^R\} \) is not regular.

(e) If \( L \) is regular, so is the set \( L^R \); see Problem 2.3.6 (g). Since \( w^R \in L \) if and only if \( w \in L^R \), we may infer that the set

\[
L \cap L^R = \{w : w \in L \text{ and } w \in L^R\} = \{w : w \in L \text{ and } w^R \in L\}
\]

is regular.

(g) Note that \( \Sigma^* = \{y : y \in \Sigma^*\} \) is a subset of \( \{xyx^R : x, y \in \Sigma^*\} \) (as the former set is obtained from the latter by taking \( x \) to be the empty string. This of course implies that the two sets are equal (and regular).