Exercise 10.1
An \( \omega \)-automaton has acceptance on transitions if the acceptance condition specifies which transitions must appear infinitely often in a run. All classes of \( \omega \)-automata (Büchi, co-Büchi, etc.) can be defined with acceptance on transitions rather than states.

Give minimal deterministic automata for the language of words over \( \{a, b\} \) containing infinitely many \( a \) and infinitely many \( b \) of the following kinds: (a) Büchi, (b) generalized Büchi, (c) Büchi with acceptance on transitions, and (d) generalized Büchi with acceptance on transitions.

Exercise 10.2
The limit of a language \( L \subseteq \Sigma^\ast \) is the \( \omega \)-language \( \lim(L) \) defined as: \( w \in \lim(L) \) iff infinitely many prefixes of \( w \) are words of \( L \), e.g. the limit of \( (ab)^\ast \) is \( (ab)^\omega \).

(a) Determine the limit of the following regular languages over \( \{a, b\} \):
   (i) \( (a + b)^\ast a^\ast \),
   (ii) \( (bab)^\ast b \).
   (iii) \( \{w : \text{At least one } a \text{ appears in } w\} \)
   (iv) \( \{w : \text{Number of appearances of } a \text{ in } w \text{ is odd and at least 3}\} \)

(b) Prove the following: An \( \omega \)-language is recognizable by a deterministic Büchi automaton iff it is the limit of a regular language.

(c) Exhibit a non-regular language whose limit is \( \omega \)-regular.

(d) Exhibit a non-regular language whose limit is not \( \omega \)-regular.

Exercise 10.3
Let \( L_1 = (ab)^\omega \) and let \( L_2 \) be the language of all words over \( \{a, b\} \) containing infinitely many \( a \) and infinitely many \( b \).

(a) Exhibit three different DBAs with three states recognizing \( L_1 \).

(b) Exhibit six different DBAs with three states recognizing \( L_2 \).

(c) Show that no DBA with at most two states recognizes \( L_1 \) or \( L_2 \).

Exercise 10.4
Show that for every NCA there is an equivalent NBA.
Solution 10.1
Automata (a), (b), (c) and (d) are respectively as follows, where colored patterns indicate the sets of accepting states or transitions:

Solution 10.2
(a) (i) \([a,b]_\omega\).
   (ii) The empty \(\omega\)-language.
   (iii) The set of \(\omega\)-words containing infinitely many \(a\).
   (iv) The set of \(\omega\)-words containing infinitely many \(a\), plus the set of \(\omega\)-words such that the number of \(a\)'s appearing in them is finite, odd and bigger than 3.

(b) Let \(B\) be a DFA recognizing \(L'\). Consider \(B\) as a DBA, and let \(L\) be the \(\omega\)-language recognized by \(B\). We show that \(L = \lim(L')\). If \(w \in \lim(L')\), then \(B\) (as a DFA) accepts infinitely many prefixes of \(w\). Since \(B\) is deterministic, the runs of \(B\) on these prefixes are prefixes of the unique infinite run of \(B\) (as a DBA) on \(w\). So the infinite run visits accepting states infinitely often, and so \(w \in L\). If \(w \in L\), then the unique run of \(B\) on \(w\) (as a DBA) visits accepting states infinitely often, and so infinitely many prefixes of \(w\) are accepted by \(B\) (as a DFA). Thus, \(w \in \lim(L')\).

If \(L\) is the limit of a regular language \(L'\), then by the above argument, it is clear that \(L\) is an \(\omega\)-language recognizable by a DBA.

Suppose \(L\) is an \(\omega\)-language recognizable by a DBA (say \(B\)). Consider \(B\) as a DFA and let \(L'\) be the language recognized by it. By the above argument, it is clear that \(L = \lim(L')\) and so \(L\) is the limit of a regular language.

(c) Let \(L = \{a^n b^n : n \geq 0\}\), which is not a regular language. Then \(\lim(L) = \emptyset\), which is \(\omega\)-regular.

(d) Let \(L = \{a^n b^n c^m : n, m \geq 0\}\). We have \(\lim(L) = \{a^n b^n c^\omega : n \geq 0\}\). Suppose this language is \(\omega\)-regular and hence recognized by a Büchi automaton \(B\). By the pigeonhole principle, there are distinct \(n_1, n_2 \in \mathbb{N}\) and accepting runs \(\rho_1, \rho_2\) of \(B\) on \(a^{n_1} b^{n_1} c^\omega\) and \(a^{n_2} b^{n_2} c^\omega\) such that the state reached in \(\rho_1\) after reading \(a^{n_1}\) and the state eached in \(\rho_2\) after reading \(a^{n_2}\) coincide. This means that \(B\) accepts \(a^{n_1} b^{n_2} c^\omega\), which contradicts the assumption that \(B\) recognizes \(L\).

Solution 10.3
(a) We obtain three DBAs for \(L_1\) from the one below by making either \(q_0\), \(q_1\) or both accepting:
(b) Here are two different DBAs for \( L_2 \). We obtain two further DBAs from each of these automata by making either \( q_1 \) or \( q_2 \) the initial state.

(c) Assume there is a DBA \( B \) with at most two states recognizing \( L_1 \). Since \( L_1 \) is nonempty, \( B \) has at least one (reachable) accepting state \( q \). Consider the transitions leaving \( q \) labeled by \( a \) and \( b \). If any of them leads to \( q \) again, then \( B \) accepts an \( \omega \)-word of the form \( wa\omega \) or \( wb\omega \) for some finite word \( w \). Since no word of this form belongs to \( L_1 \), we reach a contradiction. Thus, \( B \) must have two states \( q \) and \( q' \), and transitions

\[ t_a = q\xrightarrow{a}q' \text{ and } t_b = q\xrightarrow{b}q'. \]

Consider any accepting run \( \rho \) of \( B \). If the word accepted by the run does not belong to \( L_1 \), we are done. So assume it belongs to \( L_1 \). Since \( \rho \) is accepting, it contains some occurrence of \( t_a \) or \( t_b \). Consider the run \( \rho' \) obtained by exchanging the first occurrence of one of them by the other (that is, if \( t_a \) occurs first, then replace it by \( t_b \), and vice versa). Then \( \rho' \) is an accepting run, and the word it accepts is the result of turning an \( a \) into a \( b \), or vice versa. In both cases, the resulting word does not belong to \( L_1 \); so we each again a contradiction, and we are done.

The proof for \( L_2 \) is similar. \( \square \)

**Solution 10.4**

Let \( A = (Q, \Sigma, \delta, Q_0, F) \) be an NCA. We construct an NBA \( B \) which is equivalent to \( A \). Observe that the co-Büchi accepting condition \( \inf(\rho) \cap F = \emptyset \) is equivalent to \( \inf(\rho) \subseteq Q \setminus F \). This condition holds iff \( \rho \) has an infinite suffix that only visits states of \( Q \setminus F \). We design \( B \) in two stages. In the first one, we take two copies of \( A \), that we call \( A_0 \) and \( A_1 \), and put them side by side; \( A_0 \) is a full copy, containing all states and transitions of \( A \), and \( A_1 \) is a partial copy, containing only the states of \( Q \setminus F \) and the transitions between these states. We write \([q,0]\) to denote the copy a state \( q \in Q \) in \( A_0 \), and \([q,1]\) for the copy of state \( q \in Q \setminus F \) in \( A_1 \). In the second stage, we add some transitions that "jump" from \( A_0 \) to \( A_1 \): for every transition \([q,0] \xrightarrow{a}[q',0]\) of \( A_0 \) such that \( q' \in Q \setminus F \), we add a transition \([q,0] \xrightarrow{a}[q',1]\) that "jumps" to \([q',1]\), the "twin state" of \([q',0]\) in \( A_1 \). Note that \([q,0] \xrightarrow{a}[q',1]\) does not replace \([q,0] \xrightarrow{a}[q',0]\), it is an additional transition. As initial states of \( B \), we choose the copy of \( Q_0 \) in \( A_0 \), i.e., \([q,0] : q \in Q_0\), and as accepting states all the states of \( A_1 \), i.e., \([q,1] : q \in Q \setminus F\).

For example, the NCA below on the left is transformed into the NBA on the right:

It remains to show that \( L_\omega(A) = L_\omega(B) \).

\( \subseteq \) Let \( w \in L_\omega(A) \). There is a run \( \rho \) of \( A \) on word \( w \) such that \( \inf(\rho) \cap F = \emptyset \). It follows that \( \rho = \rho_0 \rho_1 \), where \( \rho_0 \) is a finite prefix of \( \rho \), and \( \rho_1 \) is an infinite suffix that only contains states of \( Q \setminus F \). Let \( \rho' \) be the run of \( B \) on \( w \) that simulates \( \rho_0 \) on \( A_0 \), and then "jumps" to \( A_1 \) and simulates \( \rho_1 \) in \( A_1 \). Notice that \( \rho' \) exists because
\( \rho_1 \) only visits states of \( Q \setminus F \). Since all states of \( A_1 \) are accepting, \( \rho' \) is an accepting run of the NBA \( B \), and so \( w \in L_\omega(B) \).

\[ \geq \] Let \( w \in L_\omega(B) \). There is an accepting run \( \rho \) of \( B \) on word \( w \). Thus, \( \rho \) visits states of \( A_1 \) infinitely often. Since a run of \( B \) that enters \( A_1 \) can never return to \( A_0 \) (there are no “back-jumps” from \( A_1 \) to \( A_0 \)), \( \rho \) has an infinite suffix \( \rho_1 \) that only visits states of \( A_1 \), i.e., states \([q, 1]\) such that \( q \in Q \setminus F \). Let \( \rho' \) be the result of replacing \([q, 0]\) and \([q, 1]\) by \( q \) everywhere in \( \rho \). Clearly, \( \rho' \) is a run of \( A \) on \( w \) that only visits \( F \) finitely often. Thus, \( \rho' \) is an accepting run of \( A \), and \( w \in L_A \).