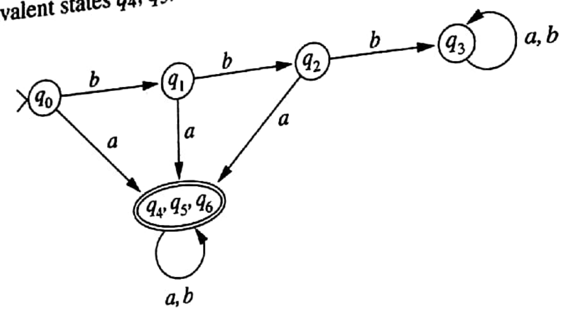
Merging equivalent states q_4, q_5, and q_6 produces

M:



□

The minimization algorithm completes the sequence of algorithms required for the construction of optimal DFAs. Nondeterminism and λ transitions provide tools for designing finite automata to solve complex problems or accept complex languages. Algorithm 6.6.3 can then be used to transform the nondeterministic machine into a DFA. The resulting deterministic machine need not be minimal. Algorithm 6.7.2 completes the process by producing the minimal state DFA.

Using the characterization of languages accepted by finite automata established in Section 7.7, we will prove that the resulting machine M′ is the unique minimal state DFA that accepts L.

Exercises

1. Let M be the deterministic finite automaton

$Q = \{q_0, q_1, q_2\}$

$\Sigma = \{a, b\}$

$F = \{q_2\}$

δ	a	b
q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_2	q_0

   a) Give the state diagram of M.
   b) Trace the computations of M that process the strings
      i) *abaa*
      ii) *bbbabb*
      iii) *bababa*
      iv) *bbbaa*
   c) Which of the strings from part (b) are accepted by M?
   d) Give a regular expression for L(M).

2. Let M be the deterministic finite automaton

$Q = \{q_0, q_1, q_2\}$

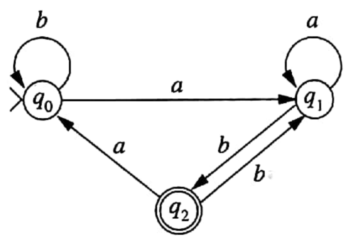
$\Sigma = \{a, b\}$

$F = \{q_0\}$

δ	a	b
q_0	q_1	q_0
q_1	q_1	q_2
q_2	q_1	q_0

   a) Give the state diagram of M.
   b) Trace the computation of M that processes *babaab*.
   c) Give a regular expression for L(M).
   d) Give a regular expression for the language accepted if both q_0 and q_1 are accepting states.

3. Let M be the DFA whose state diagram is given below.



   a) Construct the transition table of M.
   b) Which of the strings *baba*, *baab*, *abab*, *abaaab* are accepted by M?
   c) Give a regular expression for L(M).

4. The recursive step in the definition of the extended transition function (Definition 6.2.4) may be replaced by $\hat{\delta}'(q_i, au) = \hat{\delta}'(\delta(q_i, a), u)$, for all $u \in \Sigma^*$, $a \in \Sigma$, and $q_i \in Q$. Prove that $\hat{\delta} = \hat{\delta}'$.

For Exercises 5 through 15, build a DFA that accepts the described language.

5. The set of strings over $\{a, b, c\}$ in which all the a's precede the b's, which in turn precede the c's. It is possible that there are no a's, b's, or c's.
6. The set of strings over $\{a, b\}$ in which the substring aa occurs at least twice.
7. The set of strings over $\{a, b\}$ that do not begin with the substring aaa.
8. The set of strings over $\{a, b\}$ that do not contain the substring aaa.
9. The set of strings over $\{a, b, c\}$ that begin with a, contain exactly two b's, and end with cc.
10. The set of strings over $\{a, b, c\}$ in which every b is immediately followed by at least one c.
11. The set of strings over $\{a, b\}$ in which the number of a's is divisible by 3.