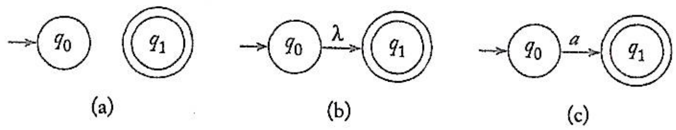

FIGURE 3.1 (a) nfa accepts $\varnothing$. (b) nfa accepts $\{\lambda\}$. (c) nfa accepts $\{a\}$.

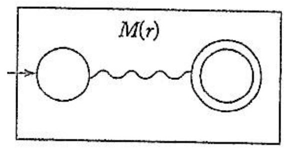

FIGURE 3.2 Schematic representation of an nfa accepting $L(r)$.

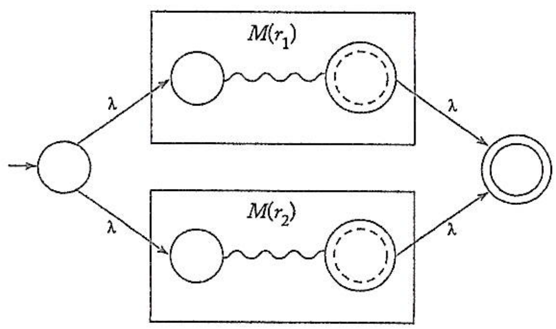

FIGURE 3.3 Automaton for $L(r_1 + r_2)$.

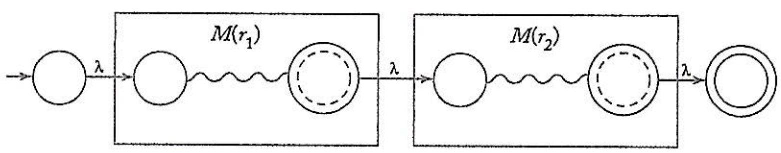

FIGURE 3.4 Automaton for $L(r_1 r_2)$.

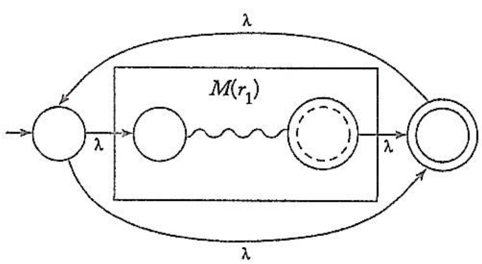

FIGURE 3.5 Automaton for $L(r_1^*)$.

EXAMPLE 3.7

Find an nfa that accepts $L(r)$, where

$$r = (a + bb)^* (ba^* + \lambda).$$

Automata for $(a + bb)$ and $(ba^* + \lambda)$, constructed directly from first principles, are given in Figure 3.6. Putting these together using the construction in Theorem 3.1, we get the solution in Figure 3.7.

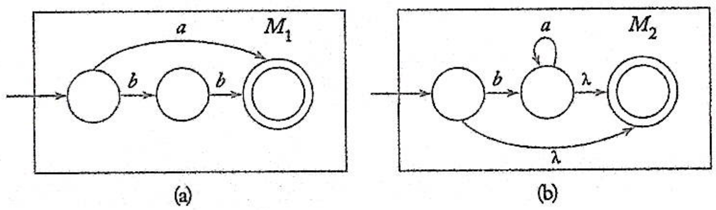

FIGURE 3.6 (a) M_1 accepts $L(a + bb)$. (b) M_2 accepts $L(ba^* + \lambda)$.

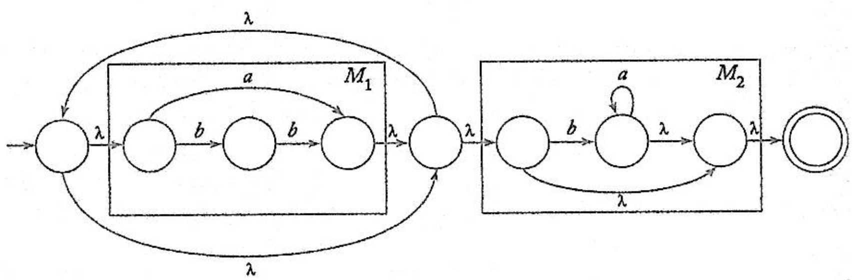

FIGURE 3.7 Automaton accepts $L((a + bb)^*\ (ba^* + \lambda))$.

EXAMPLE 3.9

The GTG in Figure 3.9(a) is not complete. Figure 3.9(b) shows how it is completed.

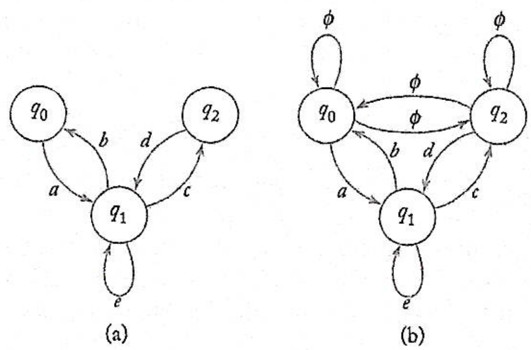

FIGURE 3.9

Suppose now that we have the simple two-state complete GTG shown in Figure 3.10. By mentally tracing through this GTG you can convince yourself that the regular expression

$$r = r_1^* r_2 (r_4 + r_3 r_1^* r_2)^* \tag{3.1}$$

covers all possible paths and so is the correct regular expression associated with the graph.

When a GTG has more than two states, we can find an equivalent graph by removing one state at a time. We will illustrate this with an example before going to the general method.

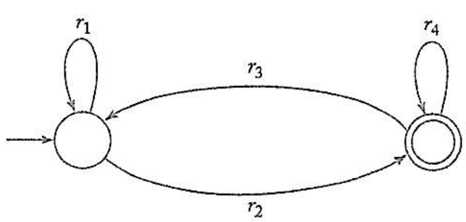

FIGURE 3.10

EXAMPLE 3.10

Consider the complete GTG in Figure 3.11. To remove q_2, we first introduce some new edges. We

create an edge from q_1 to q_1 and label it $e + af^*b$,
create an edge from q_1 to q_3 and label it $h + af^*c$,
create an edge from q_3 to q_1 and label it $i + df^*b$,
create an edge from q_3 to q_3 and label it $g + df^*c$.

When this is done, we remove q_2 and all associated edges. This gives the GTG in Figure 3.12. You can explore the equivalence of the two GTGs by seeing how regular expressions such as af^*c and e^*ab are generated.

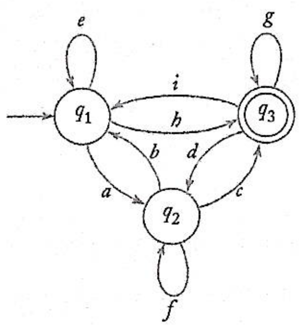

FIGURE 3.11

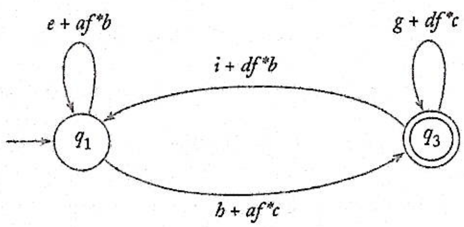

FIGURE 3.12

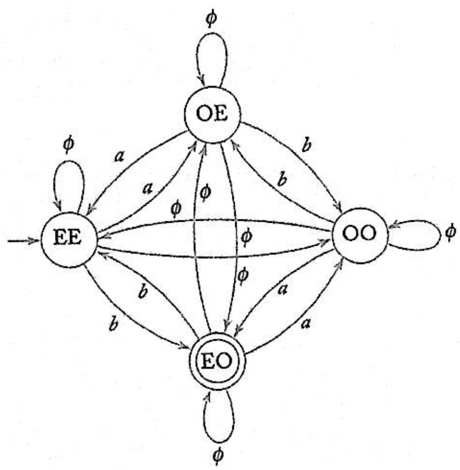

FIGURE 3.13

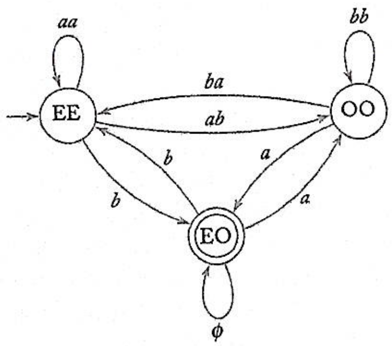

FIGURE 3.14

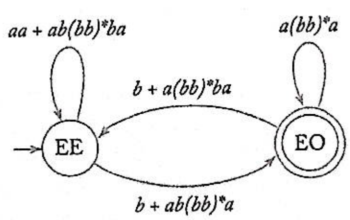

FIGURE 3.15

We continue in this manner until we get the GTG in Figure 3.14. Next, the state OO is removed, which gives Figure 3.15. Finally, we get the correct regular expression from Equation (3.2).