THEOREM 4.10.1 *The languages $L(G)$ and $L(G) \cup \{\epsilon\}$ generated by regular grammars G and recognized by finite-state machines are the same.*

Proof Given a regular grammar G, we construct a corresponding NFSM M that accepts exactly the strings generated by G. Similarly, given a DFSM M we construct a regular grammar G that generates the strings recognized by M.

From a regular grammar $G = (\mathcal{N}, \mathcal{T}, \mathcal{R}, \text{S})$ with rules $\mathcal{R}$ of the form $\text{A} \rightarrow a$ and $\text{A} \rightarrow b\text{C}$ we create a grammar G' generating the same language by replacing a rule $\text{A} \rightarrow a$ with rules $\text{A} \rightarrow a\text{B}$ and $\text{B} \rightarrow \epsilon$ where B is a new non-terminal unique to $\text{A} \rightarrow a$. Thus, every derivation $\text{S} \overset{*}{\Rightarrow}_G \boldsymbol{w}$, $\boldsymbol{w} \in \mathcal{T}^*$, now corresponds to a derivation $\text{S} \overset{*}{\Rightarrow}_{G'} \boldsymbol{w}\text{B}$ where $\text{B} \rightarrow \epsilon$. Hence, the strings generated by G and G' are the same.

Now construct an NFSM $M_{G'}$ whose states correspond to the non-terminals of this new regular grammar and whose input alphabet is its set of terminals. Let the start state of $M_{G'}$ be labeled S. Let there be a transition from state A to state B on input a if there is a rule $\text{A} \rightarrow a\text{B}$ in G'. Let a state B be a final state if there is a rule of the form $\text{B} \rightarrow \epsilon$ in G'. Clearly, every derivation of a string $\boldsymbol{w}$ in $L(G')$ corresponds to a path in M that begins in the start state and ends on a final state. Hence, $\boldsymbol{w}$ is accepted by $M_{G'}$. On the other hand, if a string $\boldsymbol{w}$ is accepted by $M_{G'}$, given the one-to-one correspondence between edges and rules, there is a derivation of $\boldsymbol{w}$ from S in G'. Thus, the strings generated by G and the strings accepted by $M_{G'}$ are the same.

Now assume we are given a DFSM M that accepts a language L_M. Create a grammar G_M whose non-terminals are the states of M and whose start symbol is the start state of M. G_M has a rule of the form $q_1 \rightarrow aq_2$ if M makes a transition from state q_1 to q_2 on input a. If state q is a final state of M, add the rule $q \rightarrow \epsilon$. If a string is accepted by M, that is, it causes M to move to a final state, then G_M generates the same string. Since G_M generates only strings of this kind, the language accepted by M is is $L(G_M)$. Now convert G_M to a regular grammar $\widetilde{G}_M$ by replacing each pair of rules $q_1 \rightarrow aq_2$, $q_2 \rightarrow \epsilon$ by the pair $q_1 \rightarrow aq_2$, $q_1 \rightarrow a$, deleting all rules $q \rightarrow \epsilon$ corresponding to unreachable final states q, and deleting the rule $\text{S} \rightarrow \epsilon$ if $\epsilon \in L_M$. Then, $L_M - \{\epsilon\} = L(G_M) - \{\epsilon\} = L(\widetilde{G}_M)$. ■

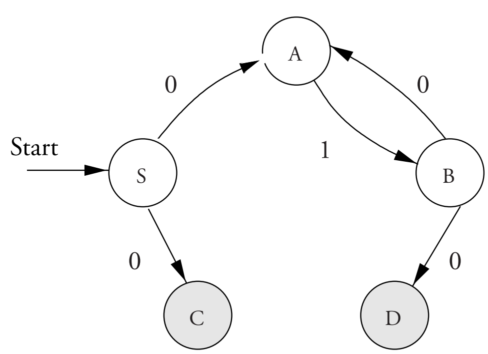

Figure 4.27 A nondeterministic FSM that accepts a language generated by a regular language in which all rules are of the form $\text{A} \rightarrow b\text{C}$ or $\text{A} \rightarrow \epsilon$. A state is associated with each non-terminal, the start symbol S is associated with the start state, and final states are associated with non-terminals A such that $\text{A} \rightarrow \epsilon$. This particular NFSM accepts the language $L(G_4)$ of Example 4.9.4.