The Theory of Computation

Bernard M.E. Moret

Addison-Wesley-Longman, 1998

Solution Manual
Chapter 3, Problem 3.11

The challenge comes from the need to keep the machine deterministic. Otherwise, it would be extremely simple: just add a new start state and a ε transition from it to the original start state. However, it is not much harder to do it deterministically. Let the original DFA be given by the quintuple $(\Sigma, Q, q_0, F, \delta)$. We define a new DFA as follows. The new set of states is $Q' = Q \cup \{q'\}$, with $q' \notin Q$; the new start state is q'; the set of accepting states is unchanged if we had $q_0 \notin F$ and is $F \cup \{q'\}$ otherwise; and the new transition function δ' includes all of the transitions of δ, plus the transitions $\delta'(q', a) = \delta(q_0, a)$ for each $a \in \Sigma$.

Obviously, the start state, once left, will never be re-entered, since no transition in the entire machine leads to it. Equally obviously, our new machine is deterministic whenever the original one is. To verify that our new machine accepts exactly the same language as the original, we observe first that it accepts ε exactly when the original does (because of our definition of the set of accepting states), then that, after reading one character, it finds itself in the "old machine"—that is, in a state of Q, from which point its behavior is identical to that of the old machine. Moreover, that state reached after one transition is the same as the old machine would have reached, because of our definition of $\delta'(q', a)$.