The Theory of Computation

Bernard M.E. Moret

Addison-Wesley-Longman, 1998

Solution Manual
Chapter 3, Problem 3.24

Our proof has two parts; one part is easy, while the converse if a bit harder.

- We prove that, if L is unitary, then, whenever we have $u, uv, w \in L$, then we must also have $wv \in L$. Because L is unitary, it is accepted by a DFA with a single accepting state; let this machine be M and its accepting state be q_F. In accepting uv, machine M must first process u; since u is in the language and since M is deterministic, M ends up in state q_F after processing u. But it also ends up in state q_F after processing uv, so that there is a loop from q_F back to q_F that is followed under input v. Consider then the processing of string wv; since M accepts w, it finds itself in state q_F after processing w and then follows the loop from q_F back to q_F under input v; hence it ends in the one accepting state, so that it accepts wv, so that wv is in L.
- Now we need to prove the converse: namely, that, given some regular language L, if having $u, uv, w \in L$ always implies $wv \in L$, then L must be unitary. We can thus start with some DFA for L and must proceed to show how to build some equivalent DFA that has a single accepting state. Let M be some DFA for L and let us assume, without loss of generality, that every state of M is accessible—that is, that, for each state q of M, there exists some string x_q that causes M to move from its start state to q. (If M had unreachable states, we could simply eliminate them to obtain the required machine; detecting unreachable states is exactly the same problem as graph reachability and is easily solved in linear time through graph traversal.) If M has a single accepting state, we are done. Otherwise we construct a new DFA M' that accepts L and has strictly fewer accepting states than M.

  If M has more than one accepting state and none of them is reachable from any other (including itself), then L is prefix-free and we can replace the accepting states of M by a single accepting state (any transition in M leading to one of its accepting states is replaced by a transition that leads to the new single accepting state in M'), from

which any transition leads to a rejecting trap. We have the desired construction.
If M has accepting states q' and q'' such that q'' is reachable from q', let v be a string that leads M from q' to q'' and let u be a string that leads M from its start state to q'. Note that M must accept, not just u and uv, but also uv^i for any $i \in \mathbb{N}$, because we can set $w = uv$ in the statement of the hypothesis and thus obtain that $wv = uvv$ is also in L—and then repeat with one more suffix of v, etc. Similarly, if w is a string that leads M from its start state to q'', then M must accept wv^i for any $i \in \mathbb{N}$. Moreover, if M accepts some string wt (with $t \neq v$) by reaching q'' and then some third accepting state q''', then it must also accept string ut (because it accepts w, wt, and u). Thus we can eliminate state q'' by producing a new DFA in which each transition into q'' in the original automaton is replaced by a transition into q'. From our arguments above the new machine accepts exactly what the old machine did and has one fewer accepting state. Our inductive step is thus complete.