The Theory of Computation
Bernard M.E. Moret
Addison-Wesley-Longman, 1998

Short Solution
Chapter 3, Problem 3.13

This solution is written in a somewhat informal style suited for communication among practitioners; a detailed solution would require an algorithmic description of the construction (particularly of each path from the LHS to the accepting state) and a formal proof that the construction does what it claims to do.

The simplest way to proceed is to use nondeterminism twice, first to get an equivalent machine with a single accepting state (a trivial exercise with nondeterminism—for instance, we can use ε transitions from the existing accepting states to a new state and make that new state the single accepting state) and secondly to set up a third equivalent machine that has a planar transition diagram. We do this second part by observing that, except for the number of times that a path can go through a loop, the number of distinct paths from any state of the NFA to its single accepting state is finite. We use this observation to create a new NFA that has all of the states of the old one and many more. Think of arraying all of the old states except the accepting state in a column on the left and the accepting state alone on the right; from each old state, we have a bundle of paths to the accepting state; ε transitions are used to choose a path in the bundle and additional states are used as necessary to implement each path—but all of these paths converge in a fan from the old states to the accepting state and thus do not cross. (Most of the states along these paths have only a few transitions, most often just one, defined and implicitly reject any other input, thereby drastically lowering the degree at each node.) From the accepting state, the old machine had some transitions back to other states; in our new machines, these transitions also exist and lead "around the back" and in the order in which old states were arrayed into the left column, so that they do not cross each other or any of the paths leading from the old states to the accepting state. The result is a planar, single-accepting state NFA that accepts exactly the same language as our original machine.