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

Solution Manual
Chapter 3, Additional Problem 3.2

Basically, we want to run the two machines in parallel on a variety of input strings, checking whether or not the two accept the same strings. The key question is how long the strings need to be before we are convinced that the two machines accept the same language. Since running the two machines together can be simulated by running a "product" machine, the states of which correspond to ordered pairs of states from the two machines, we need only ask how many states the product machine has. Given a DFA of m states and one of n states, the product machine has mn states, so that the longest string that we need to run through it need not exceed $mn - 1$ symbols (after that, we must have a loop). Now each different string may cause the machine to follow a different path among its states; we need to explore as many paths as we can in order to reach every paired state that can be reached, so we must be prepared to test every string of length up to $mn - 1$, or order $|\Sigma|^{mn}$ strings.

An important problem that would arise in practice in such a test is whether, after running a string through a DFA, there is any way to force the DFA back to its start state. In our answer above, we have assumed such to be the case—automagically (we have some sort of reset button). In practice while testing a sequential machine (such as a CPU), no such assumption can be made (even if there is a reset line, it too must be tested). What one then needs is an input sequence that will take the machine (if it functions properly) from any state back to the start state in some guaranteed number of moves. Such sequences are known as homing or synchronizing sequences, but not every DFA has one.