Complete the proof of Theorem 3.3 by showing how to modify a machine that accepts $L(r)$ into a machine that accepts $L(r^*)$.

Answer: Let M be the machine that accepts $L(r)$. To accept $L(r^*)$, M' should have a new start state q_0' and an ϵ-transition from q_0' to M's start state q_0. In addition, add ϵ-transitions from each of M's final states back to q_0. Finally, designate q_0' a final state. (M's final states should remain final states too.)

Discussion: Observe that $L(r^*) = L(\epsilon|r|rr|rrr|\ldots)$.

Let M be an NFA accepting $L(r)$ and let M_* be an NFA accepting $L(r^*)$.

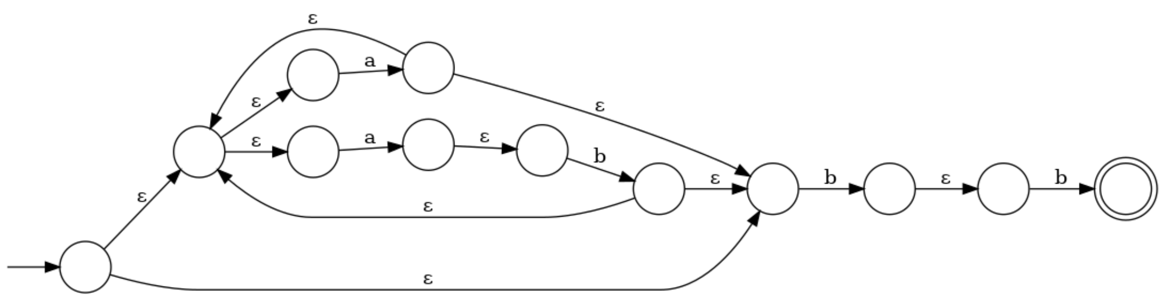
For a fixed number of copies of r, such as $L(rrr)$, we can use the construction for concatenation: connect that many copies of M in sequence, with the final state(s) of each copy connected to the start start of the next with ϵ-transitions, the start state of the whole machine being the start state of the first copy, and the final state(s) for the whole machine being the final state(s) of the last copy.

Using this idea, we can construct a machine M' accepting $L(r|rr|rrr|\ldots)$ with just a single copy of M, connecting the final state(s) of M to its start state with an ϵ-transition and leaving the start state and final state(s) as they are.

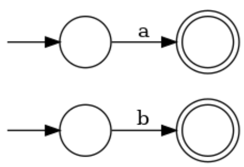
The only thing missing from $L(r^*)$ is the empty string. An NFA M_ϵ accepting $\{\epsilon\}$ consists of a single state which is both the start state and a final state. We then use the construction for | to combine M_ϵ and M': create a new start state which is connected to the start states of M_ϵ and M' with ϵ-transitions. The NFA described in the answer above is a slightly simplified version of this — since nothing connects back to this new start state, it is only reachable when ϵ has been consumed, so making it final and getting rid of M_ϵ doesn't break anything.

Using the construction described in Theorem 3.3, build an NFA that accepts $L((ab|a)^*(bb))$.

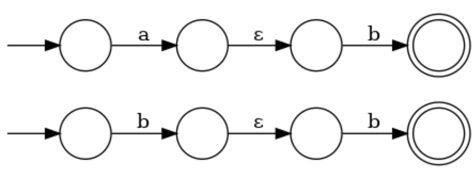
Answer:



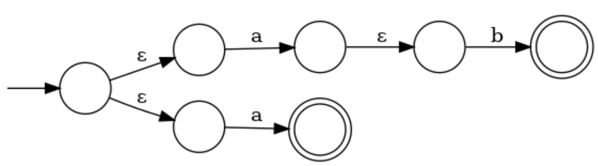
Discussion: Break $(ab|a)^*(bb)$ down into its smallest pieces (individual symbols), build the NFAs for those elements, then combine those NFAs using the construction appropriate to each operator. So, to start, NFAs accepting $L(a)$ and $L(b)$:



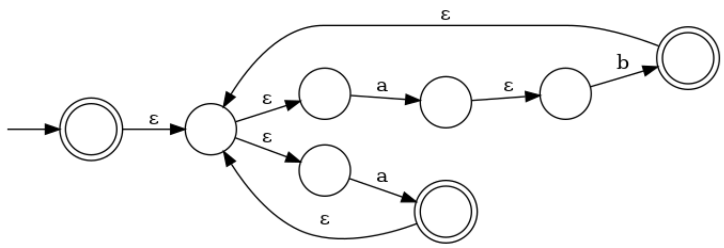
Next, NFAs for $L(ab)$ and $L(bb)$, using the construction for concatenation:



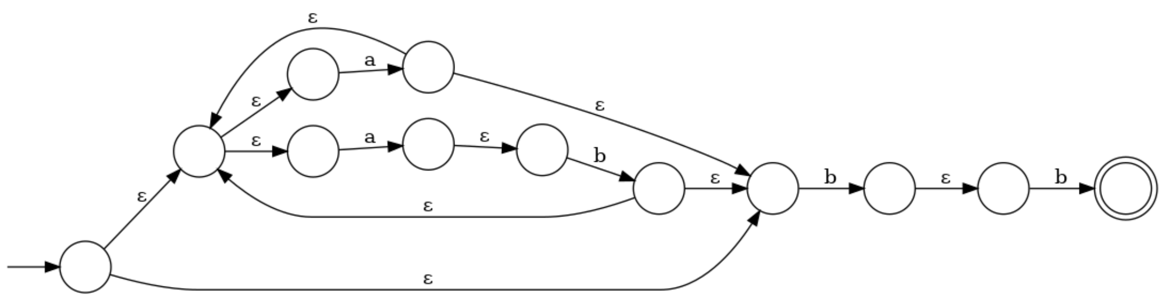
Now, for $L(ab|a)$, using the construction for $|$:



Now, for $L((ab|a)^*)$, using the construction for $*$:



And finally, the NFA for $L((ab|a)^*(bb))$, using the construction for concatenation:



This is certainly not the simplest possible NFA for $L((ab|a)^*(bb))$, but the task here was to build *an* NFA and to utilize the construction from Theorem 3.3.

Show that for any DFA or NFA, there is an NFA with exactly one final state that accepts the same language.

Answer: Let M be the original DFA or NFA. Construct M' with the same states and transitions as M with the following modifications:

- Add a new final state q_f and ϵ-transitions from M's final states to q_f.
- The only final state in M' is q_f. (M's final states are not final in M'.)

Any string that reached a final state in M will be able to reach q_f along the ϵ-transition so M' accepts all of those strings, and q_f is only reachable from one of M's final states so no other strings will be accepted by M'.