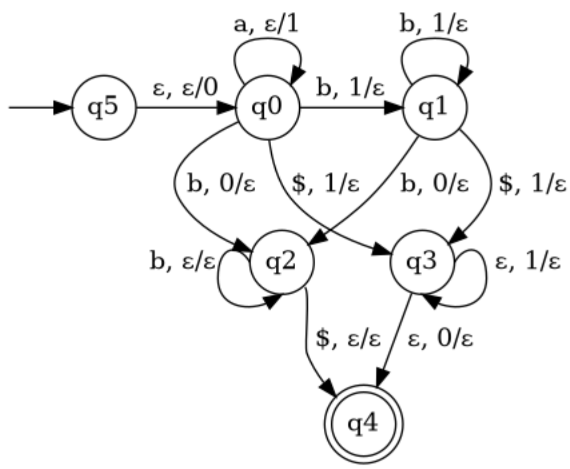
Show that the language $\{ a^n b^m \mid n \neq m \}$ is deterministic context-free.

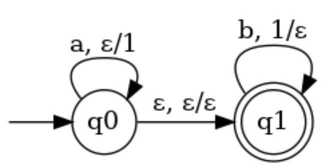
Answer:

Definition 4.5 says that L is deterministic context-free if there is a deterministic pushdown automaton accepting $L\$$. The following is such an automaton.

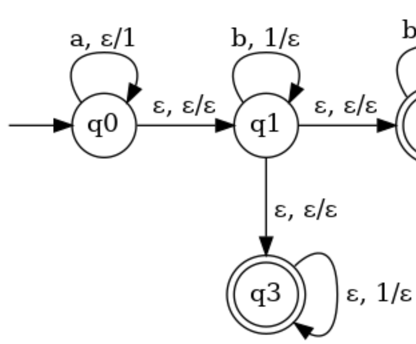


Discussion: A way to start is with a pushdown automaton that accepts something similar to L, then modify it to accept $L\$$ and finally make it deterministic.

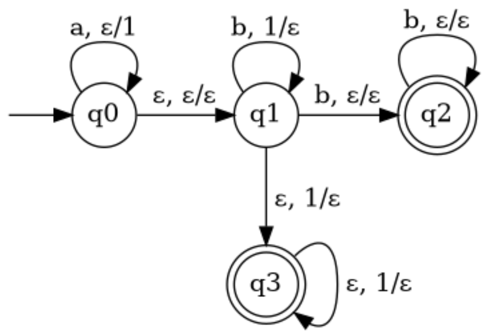
We've seen a pushdown automaton for $\{ a^n b^m \mid n = m \}$, so let's start with that.



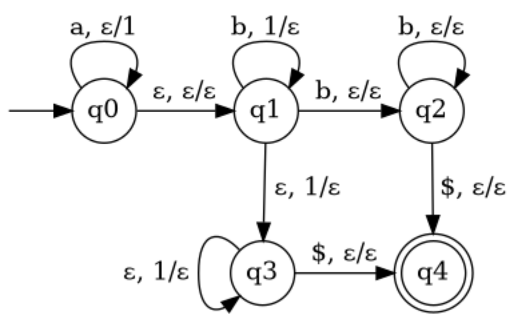
Now modify it for $n \neq m$. There are two possibilities — if there are more as than bs, the machine will end up in state q_1 with an empty string but a non-empty stack, and if there are more bs than as, the machine will end up in state q_1 with an empty stack but with at least one b left in the string.



This accepts $n \neq m$, but there's a problem — it also still accepts $n = m$. (Why?) The solution is to consume something additional (from the string or from the stack) in order to move to the final states. That ensures that there is an additional b beyond what matches all the as, or that there is an additional a beyond what matches all the bs.



This accepts L. Now modify it to accept $L\$$ — just consume the final \$ once a final state for L has been reached.



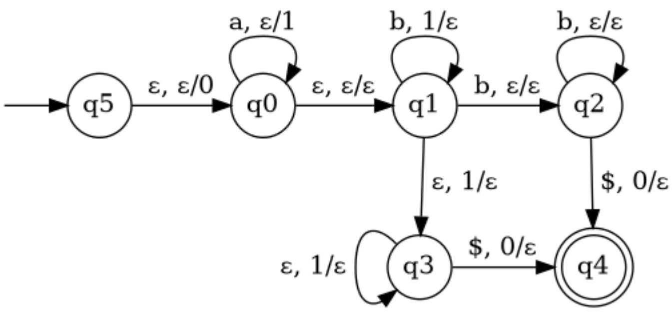
This accepts $L\$$, but it is non-deterministic because of the transitions leaving q_0 and q_1. All three of these transitions are meant to apply only in specific circumstances — after all of the as have been read (for leaving q_0) and after an equal number of as and bs have been read (for leaving q_1). The problem is transitions that allow nothing to be consumed; the solution is to require something distinct to be read from the string and/or the stack.

Let's start with q_0. The ϵ-transition is meant to apply when all of the as have been consumed. There are three possibilities for the string: one or more as are followed by one or more bs, one or more as are followed by the end of the string (no bs), and zero as are followed by one or more bs. (Zero as followed by zero bs isn't in the language because it has an equal number of as and bs.)

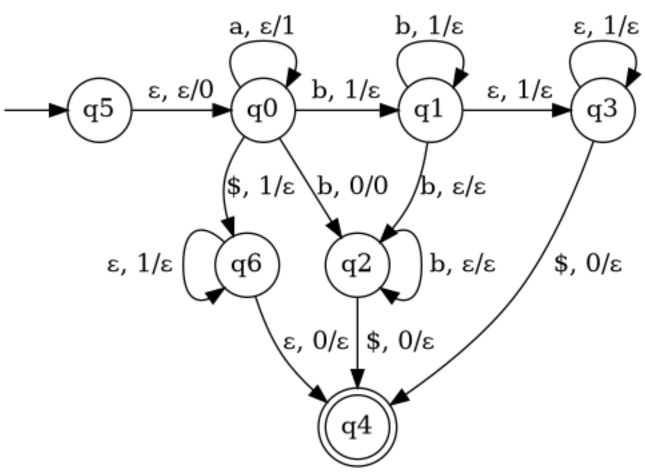
- In the first case (as followed by bs), the next thing after the as is to consume a b and match it with one of the as counted on the stack. Add a transition $q_0 \xrightarrow{b,1/\epsilon}$.

- In the second case (as with no bs), the next thing after the as is the string-ending \$, so consume that — and match with a 1 on the stack to prevent accepting just \$. Add a transition $q_0 \xrightarrow{\$,1/\epsilon}$.
- In the third case (bs with no as), the next thing is also to consume a b, but there's nothing on the stack to match it with — and adding $q_0 \xrightarrow{b,\epsilon/\epsilon}$ means we haven't fixed the non-deterministic problem.

Avoiding non-determinism here requires being able to detect an empty stack, which can be fixed with a trick similar to why \$ was added to mark the end of the string — push something (say 0) onto the stack right off the bat so the bottom of the stack can be recognized. Add a new start state to do this, then also be sure to pop the 0 at the end — say when the \$ is consumed.



Now it is possible to replace the ϵ-transition leaving q_0:

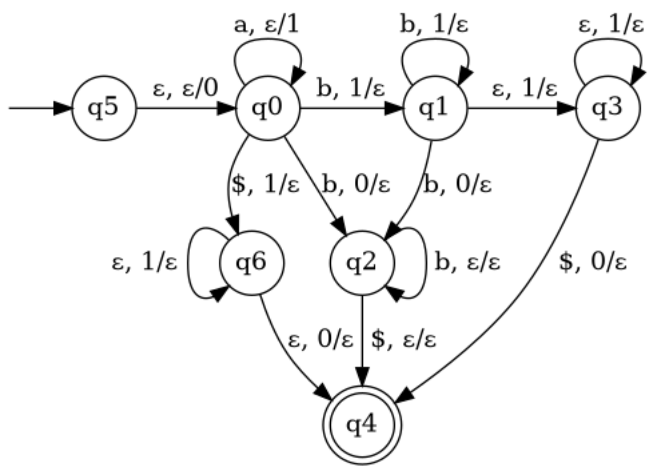


To keep track of what is going on here —

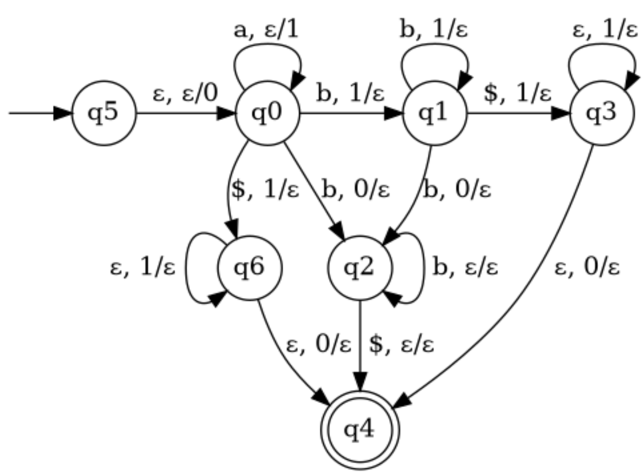
- q_0 counts as, pushing a 1 onto the stack for each a
- q_1 matches bs to as, popping a 1 for each b
- q_2 consumes extra bs after all of the as have been matched

- q_3 pops extra 1s after all of the bs have been used
- q_6 pops extra 1s in the case of no bs; it is distinct from q_3 because the $ has already been consumed

q_1 is now the only state remaining with non-deterministic transitions. Consider first the two b transitions — the self-loop matches bs with as and the transition to q_2 is intended to be used when there are no more as to be matched with but there are still bs remaining. No more as means the stack is "empty" — there's only the bottom-marker 0. So the transition to q_2 should pop the 0 from the stack — and then 0 won't be popped later with the $, and it should be popped on the transition from q_0.



Now consider the two pop-1 transitions leaving q_1 — the self-loop matches bs with as and the transition to q_3 is intended to be used when all of the bs have been matched but there are still as remaining. No more bs means the next thing in the string is $, so consume that in the transition q_3 and not in the transition $q_3 \longrightarrow q_4$.



This is now a deterministic pushdown automaton accepting $L\$$, so L is thus deterministic context-free. We can stop here since we just need *a* deterministic pushdown automaton accepting $L\$$, but it is worth noting that q_3 and q_6 are now redundant and q_6 can be eliminated.