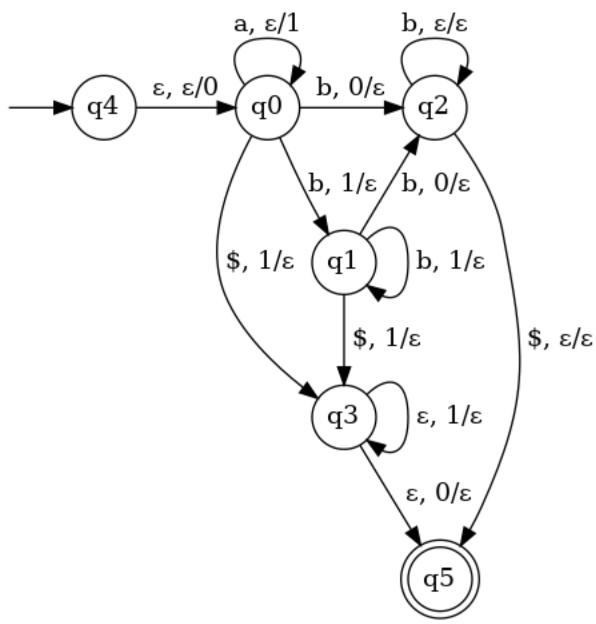
Show that the language $\{ a^n b^m \mid n \neq m \}$ is deterministic context-free.

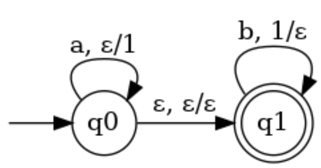
Answer:

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.

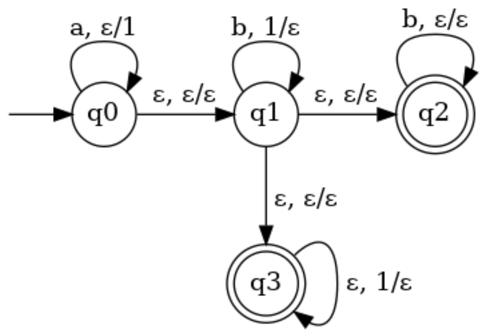


Discussion: A way to start is with a pushdown automaton that accepts something similar to L, then modify it to accept $L\$$ and finally make it deterministic.

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

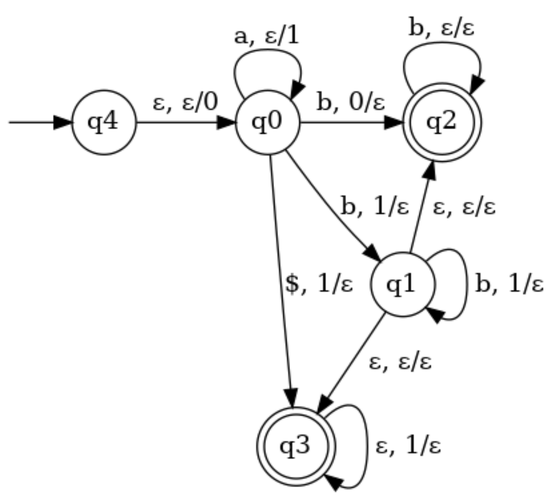


Now modify it for $n \neq m$. There are two possibilities — if there are more as than bs, the machine will end up in state q_1 with an empty string but a non-empty stack, and if there are more bs than as, the machine will end up in state q_1 with an empty stack but with at least one b left in the string.



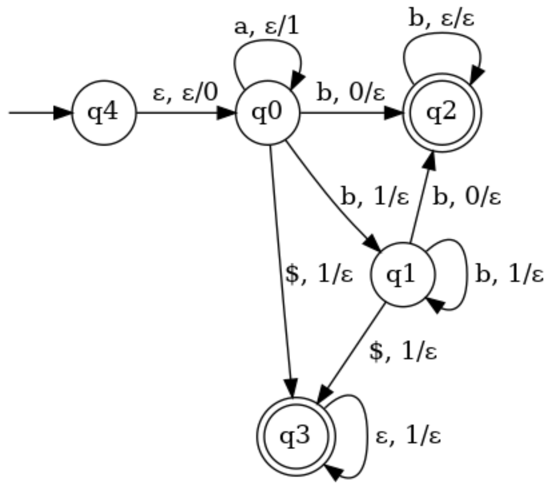
This is non-deterministic because of the ϵ-transitions leaving q_0 and q_1.

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.) The first case is addressed by a transition $\xrightarrow{b,1/\epsilon}$ and the second case is addressed by a transition $\xrightarrow{\$,1/\epsilon}$, but the third case requires a transition $\xrightarrow{b,\epsilon/\epsilon}$ because the stack is empty — but that's still non-deterministic because it could be applied instead of $\xrightarrow{b,1/\epsilon}$. Fixing this needs a trick similar to the role of the $ to mark the end of the string — push something onto the stack right off the bat so the bottom of the stack can be recognized.



Now consider the ϵ-transitions leaving q_1. Start with $q_1 \to q_2$. q_1 handles reading bs while there aren't yet as many bs as as (i.e. the stack isn't empty). The transition to q_2 is intended for when there are more bs than as — the stack is empty so the remaining bs need to be consumed. Change the transition to consume a b (there's at least one or else there would be the same number of as and bs) and pop the bottom-of-stack symbol.

Then consider the ϵ-transition $q_1 \to q_3$. q_3 is used to empty the stack when the end of the string has been reached (more as than bs). Thus this transition applies at the end of the string and should pop a 1 (there is at least one or else the number of as and bs are equal).



This is now deterministic, but it's not quite complete — the end-of-string \$ needs to be consumed and the bottom-of-stack 0 needs to be popped in all cases. A new final state is added to ensure that the \$ is really the last thing read.