CS 381 - A4

Martin Mueller

Due: Friday, March 6^{th}, 2020

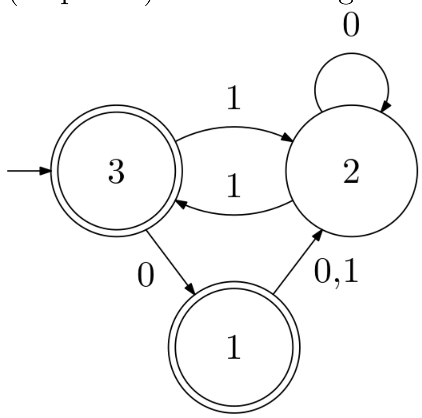
1. (10 points) Prove that the language $\{0^n 1^m \mid n \neq m\}$ is not regular.

Let's call the above language L. For the sake of obtaining a contradiction, assume L is regular. Let p be the pumping length given by the Pumping Lemma. Let $s = 0^p 1^{p+p!}$. Since $|s| \geq p$, the Pumping Lemma says we can write $s = xyz$ where $|xy| \leq p$, $|y| > 0$, and $xy^i z \in L$ for all $i \geq 0$. From this, we know that y must be comprised of entirely zeros because $|xy| \leq p$. We can now rewrite s as $0^{p-q}0^{qi}1^{p+p!}$ for all $i \geq 0$. If we chose to pump up s by setting $i = 1 + p!/q$ where q is $|y|$, then we arrive at

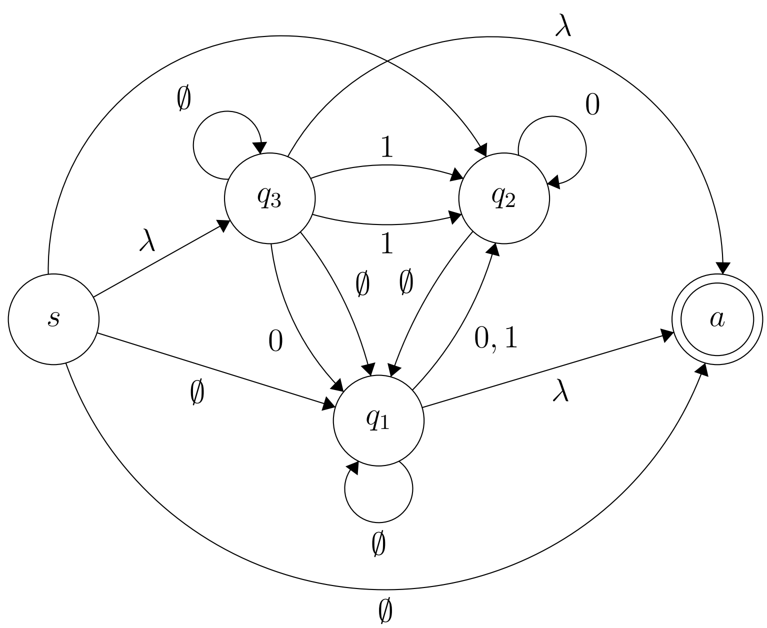
$$\begin{aligned} s &= 0^{p-q}0^{q(1+p!/q)}1^{p+p!} \\ &= 0^{p-q+q+p!q/q}1^{p+p!} \\ &= 0^{p+p!}1^{p+p!} \notin L \end{aligned}$$

As we can see, pumping up the string s yields a string that is no longer in the language. Therefore, L is not a regular language by the pumping lemma because the string $xy^i z$ is not in L for all possible values of i greater than 0.

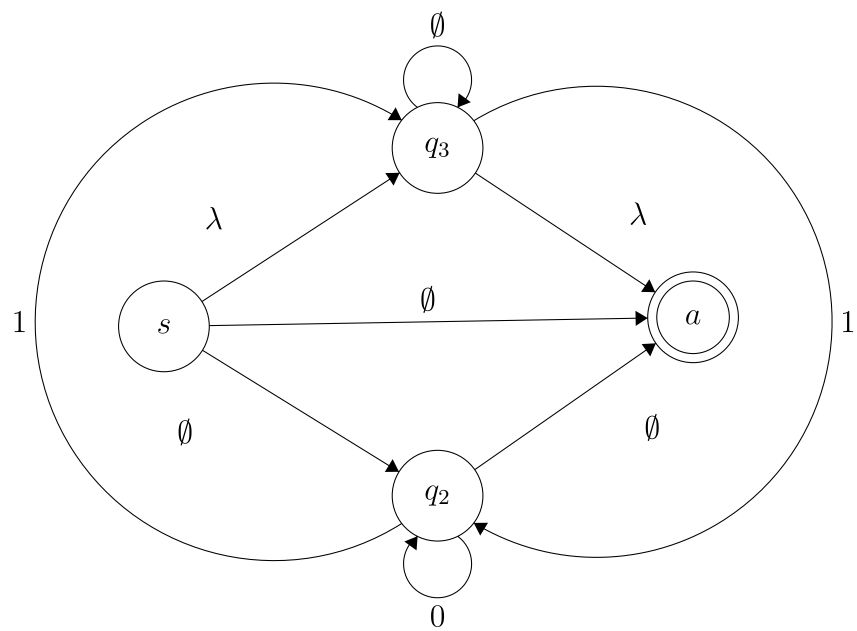
2. (10 points) Convert the given machine into a corresponding regular expression.



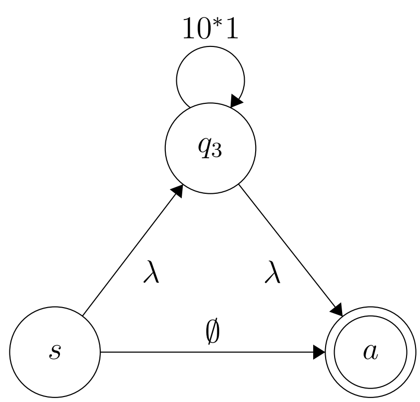
Constructed initial GNFA by adding a new start state with a λ transition to the old start state, adding a new accept state that replaces all other accept states, adding λ transitions from the old accept states to the new accept state, and adding $\emptyset$ transitions where needed.



Ripped state state q_1 and updated all transitions as needed.



Ripped state state q_2 and updated all transitions as needed.



Ripped state state q_3 and updated all transitions as needed.