Exercise 0.1. For any language $L$, let $\text{NOREPEATB}(L)$ be the language of strings in $L$, but with any $b$'s that are immediately preceded by another $b$ removed. So, for example, if $babbaababbb \in L$, then $babaabab \in \text{NOREPEATB}(L)$. Show that if $L$ is regular, then so is $\text{NOREPEATB}(L)$.

Solution:

We want to construct a machine that accepts strings $s$ without consecutive $b$'s such that some number of $b$'s can be added after any $b$ in $s$ to yield a string in $L$. The intuition is to modify a DFA for $L$ so that its states track the previous character read. If the previous character read was an $a$, then it should act much like the DFA for $L$. If the previous character read was a $b$, then the machine should be able to follow any number of $\epsilon$ transitions which would simulate what would happen if the DFA for $L$ read that number of $b$'s instead of just one.

Specifically, if $M = (Q, \Sigma, \delta, q_0, F)$ is the original DFA, let $M' = (Q', \Sigma, \delta', q_0, F')$ be the NFA that will recognize $\text{NOREPEATB}(L)$. We define $Q'$ as follows: for each $q_i \in Q$, let there be a corresponding $q_i', q_i'' \in Q'$, i.e., $Q'$ contains every state in $Q$ plus a corresponding “prime” state. We will use this notation throughout the proof. Define $\delta'$ as follows: suppose $\delta(q_i, a) = q_j$ and $\delta(q_i, b) = q_k$. Then

\[
\begin{align*}
\delta'(q_i, a) &= \{q_j\} \\
\delta'(q_i, b) &= \{q_k\} \\
\delta'(q_i', a) &= \{q_j\} \\
\delta'(q_i', \epsilon) &= \{q_k\} \\
\delta'(q_i', b) &= \emptyset
\end{align*}
\]

To summarize, $M'$ behaves identically to $M$ when it reads $a$: if it is in a prime or a non-prime state, it transitions to the non-prime state that corresponds to the state $M$ would transition to. When $M'$ reads $b$, its behavior depends on the previous input. If it is in a non-prime state, it transitions to the prime state that corresponds to the state $M$ would transition to. If it is in a prime state, it enters into a null state on reading $b$. Finally, for every $b$ transition in $M$ from $q_i$ to $q_j$, there is a corresponding $\epsilon$ transition in $M'$ from $q_i'$ to $q_j'$. Finally, for all $q_i \in F$, we say let the corresponding states $q_i, q_i' \in F'$.
Now, suppose we have a string $w \in L$. If $w$ contains no $b$'s or no repeat $b$'s, then \text{NOREPEATB}(w)$, which we define to be $w$ with all repeat $b$'s removed, has no $b$'s and $M'$ will behave identically to $M$ and hence will accept $\text{NOREPEATB}(l)$. So, suppose $w$ has at least one string of one or more repeated $b$'s. Then $M'$ will read $\text{NOREPEATB}(w)$ until it hits the first $b$, at which time it will enter a prime state. It will then undergo $\epsilon$ transitions for each prime state corresponding to $M$’s original behavior on $w$ on reading the repeated $b$’s. Thus, $M'$’s behavior on $\text{NOREPEATB}(w)$ will mimic that of $M$ on $w$, so $M'$ will accept.

Finally, suppose we have some $w \notin \text{NOREPEATB}(L)$. Then we cannot repeat $b$’s in $w$ to achieve a string in $L$. Thus, there can be no computation path in $M'$ that leads to an accept state on reading $w$, since the only difference between $M$ and $M'$ is that $M'$ allows $\epsilon$ transitions in place of transitions in the case of a repeated $b$ that correspond exactly to $M$’s behavior.

Thus $M'$ accept $\text{NOREPEATB}(L)$ and only this language, so that $\text{NOREPEATB}(L)$ is regular.