1. (10 pts) Consider the following property, AllButFirstReplace with \( \Sigma = \{a, b, c\} \).

\[
\text{AllButFirstReplace}(L) = \{w \mid w \in L, \text{ and } w \text{ does not contain an } a\} \cup \{wax \mid wav \in L, \text{ and } w \text{ does not contain an } a \text{ and } x \text{ is } v \text{ with every occurrence of } a \text{ in } v \text{ replaced by } b\}
\]

The property AllButFirstReplace applied to a language \( L \) accepts strings from \( L \) with all but the first occurrence of \( a \) replaced with \( b \). For example, if the string \( babaab \) is in \( L \), then \( babbbb \) is in AllButFirstReplace(\( L \)). If \( aabcab \in L \), then \( abbcbb \in \text{AllButFirstReplace}(L) \). If \( bcb \in L \), then \( bcb \in \text{AllButFirstReplace}(L) \).

**Prove** that the regular languages are closed under the AllButFirstReplace(\( L \)) property. (Show all steps!)

**Proof:**

[Idea: start with a DFA for any language and show how to convert it into an NFA that accepts AllButFirstReplace(\( L \))]  

\( L \) is regular, so there exists a DFA for \( L \) called \( M \).

\( M \) is defined as \((Q, \Sigma, \delta, q_0, F)\).

\( M \) has one start state and some number of final states and can be seen as a black box as:

(not everything shown).

\[ M: \]
Construct an NFA $M'$ that recognizes $\text{AllButFirstReplace}(L)$.

Idea: Make two copies of $M$. For each arc $a$ in the first copy of $M$ going from state $p$ to $q$, replace it with an arc going from state $p$ in the first copy to state $q$ in the second copy. Make all arcs on $a$ in the second copy now an arc on $a \ b$.

Here is $M$ again, drawn with some possible arcs.

\[ M: \]

Here is the new NFA $M'$ which is 2 copies of $M$ with the changes described above.
Note that in M', you start in the first copy and process the string as before. If there is no a, you end up in a final state. If there is an a, when you come to the first a, you move over into the 2nd copy. Here all a's have been changed to b's. So you finish up with b's for what was a's before.

So the string abcabbca was in M, and the string abcbbcbcb is in the NFA M'.
Formally (and you must give this also).

We’ll call the second copy of M everything with ” on them. so the second copy of M is $M''$ is defined as $(Q'', \Sigma, \delta'', q_0'', F'')$.

Then define the new NFA $M'$ for AllButFirstReplace($L$) as $(Q', \Sigma, \delta', q_0', F')$.

where

\[ Q' = Q \cup Q'' \] (the union of the two states from the two copies of M)

\[ q_0' = q_0 \] is the start state in the first copy of M

\[ F' = F \cup F'' \] is the set of final states in both copies of M

\[ \delta' = \delta - \{(p, a, q) \mid p, q, \in Q\} \cup \delta'' - \{(p'', a, q'') \mid p'', q'\in Q''\} \cup \{(p'', b, q'') \mid p'', q'' \in Q'' \text{ and } (p, a, q) \in \delta\} \cup \{(p, a, q'') \mid p \in Q, q'' \in Q'' \text{ and } (p, a, q) \in \delta\} \]

The new $\delta'$ is the $\delta$ from the first copy minus all the $a$ arcs, plus all the arcs in the second copy minus all the $a$ arcs, plus $b$ arcs added to the second copy if there was an $a$ arc before, plus $a$ arcs added between the two copies from $p$ to $q''$ if there was an $a$ in the original $\delta$ from $p$ to $q$ (note that $q''$ is the copy of the state $q$ from the first copy).