Problem Set 3

Problem 1. Using the procedure shown in class, convert the following NFA into a DFA for the same language.

\[
\begin{array}{c}
0, 1 \\
A \rightarrow B \leftarrow 1 \\
A \rightarrow C \leftarrow 2 \\
\end{array}
\]

Problem 2. For any language \( L \) let 
\[ \text{noprefix}(L) = \{ w \in L \mid \text{no proper prefix of } w \text{ is a member of } L \} \]
Prove or disprove: if \( L \) is DFA-acceptable then so is \( \text{noprefix}(A) \).

Problem 3. For \( n \geq 0, \) let \( L_n = \{ 1^i : 0 \leq i < n \} \) (where \( 1^0 = \varepsilon \)). Prove that there is a DFA \( M_n \) having \( n \) final states that accepts \( L_n \). Then prove that \( L_n \) cannot be accepted by any DFA having fewer accept states.

Problem 4. Consider applying the product construction to NFAs \( M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1) \) and \( M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2) \) in order to show that the NFA-acceptable languages are closed under intersection.

Part A. Formally specify the product machine \( M = (Q, \Sigma, \delta, q_0, F) \).

Part B. Does the construction work—that is, is \( L(M) = L(M_1) \cap L(M_2) \)? Informally argue your conclusion.