1. Give a DFA $D$ with alphabet $\{a, b\}$ and no more than 4 states for the language $L = \{w \in \{a, b\}^* \mid w \text{ ends in } aba\}$. Prove that $L(D) = L$ by providing the pathsets of the states of $D$.

2. The transition diagram of NFA $N$ is given by

   ![Transition Diagram of NFA N](image)

   (a) Using the NFA-to-DFA algorithm transform $N$ into a DFA $D$ that accepts $L(N)$. Provide both the transition table and the transition diagram of $D$. [hint: the resulting DFA has 3 states]

   (b) Using the NFA-to-RE algorithm based on GNFA construct starting with NFA $N$ a regular expression for $L(N)$. Provide all intermediate GNFA and the constructed regular expression.

3. Given are an alphabet $\Sigma$, and a regular language $L$ over $\Sigma$. Function $\text{dbl} : \Sigma^* \to \Sigma^*$ is defined recursively by

   $\text{dbl}(\varepsilon) = \varepsilon$

   $\text{dbl}(au) = aa \text{ dbl}(u)$ for all $a \in \Sigma, u \in \Sigma^*$

   (i.e. $\text{dbl}(c_1c_2\ldots c_k) = c_1c_1c_2c_2\ldots c_kc_k$ for all $k \geq 0$ and $c_1, c_2, \ldots, c_k \in \Sigma$). Language $L'$ is defined by

   $L' = \{\text{dbl}(w) \mid w \in L\}$.

   Let $D = (Q, \Sigma, \rightarrow, q_0, F)$ be a DFA with $L(D) = L$. Show how to construct using $D$ an NFA $N$ such that $L(N) = L'$. Give some informal arguments why the constructed NFA $N$ accepts language $L'$ (no formal proof required).
4. Given is language \( L = \{a^{2n}b^n \mid n \geq 0\} \).

(a) Construct a push-down automaton with no more than 5 states accepting \( L \) and give the invariant table for it.

(b) Give a context-free grammar \( G \) for \( L \) with only 1 variable and prove \( \mathcal{L}(G) \subseteq L \).

5. Function \( f : \{a,b\}^* \rightarrow \{A\}^* \) is defined by

\[
\begin{align*}
  f(\varepsilon) &= \varepsilon \\
  f(aw) &= AAf(w) \quad \text{for all } w \in \{a,b\}^* \\
  f(bw) &= f(w) \quad \text{for all } w \in \{a,b\}^*
\end{align*}
\]

(string \( f(w) \) is obtained from \( w \) by deleting all \( b \)'s, and changing every \( a \) into two \( A \)'s). Give a classical Turing machine with at most 5 states that computes \( f \) and show the computation of \( f(abab) \) by this Turing machine.

6. Given are labeled transition systems \( S \) and \( T \):

\[
\begin{array}{c}
S : \\
\begin{array}{c}
p_0 \quad \tau \quad a \\
p_1 \quad a \\
p_2 \\
p_3 \quad b \\
\end{array} \\
T : \\
\begin{array}{c}
q_0 \quad \tau \quad a \\
q_1 \\
q_2 \\
q_3 \quad a \quad \tau \\
\end{array}
\end{array}
\]

Compute a coloring scheme for the combination of these LTSs (your computation should be presented in the form of one table for the combined colorings of \( S \) and \( T \)) and indicate whether and why \( S \) and \( T \) are branching bisimilar or not.