Problem 1: In class, I made the claim that not every non-regular language can be shown to be non-regular using the pumping lemma. Consider the language \( L = \{a^i b^j c^k : i, j, k \geq 0 \text{ and if } i = 1 \text{ then } j = k \} \).

(a) Show that the pumping lemma cannot be used to prove that \( L \) is not regular.

(b) Read about the Myhill-Nerode theorem, which is exercise 1.52 of your book. Note that the solution to the exercise appears at the end of the chapter. Use the Myhill-Nerode theorem to prove that \( L \) is not regular.

Problem 2: Prove that the languages recognized by NFAs are closed under complement.

Problem 3: The Myhill-Nerode theorem implies that for any regular language \( L \), any DFA recognizing \( L \) has to have at least \( i \) states, where \( i \) is the index of \( L \). In fact, there is a DFA of size \( i \) that accepts \( L \). This is a minimal DFA recognizing \( L \).

(a) Propose an algorithm for DFA minimization.

(b) Prove that your algorithm is correct.

(c) Use your algorithm to minimize the following DFA.

![DFA Diagram](image)

Problem 4: In class we studied finite-state automata that operate on strings of finite length. How about finite-state automata that operate on strings of infinite length? Define an *Infinite Input Finite Automaton* (IIFA) to be a tuple \((Q, \Sigma, T, Q_0, F)\) where:

- \( Q \) is a finite set of states.
- \( \Sigma \) is the alphabet.
- \( \delta : \Sigma \times Q \to \mathcal{P}(Q) \) is the transition function.
• \( Q_0 \subseteq Q \) is a set of initial states.
• \( F \subseteq Q \) is a set of accepting states.

Given an infinite string \( s = s_0s_1 \ldots \) over \( \Sigma \), a run \( r \) of IIFA \( A \) on \( s \) is an infinite sequence of states \( r = r_0, r_1, \ldots \) where \( r_0 \in Q_0 \) and \( r_{i+1} \in \delta(r_i, s_i) \) for all \( i \geq 0 \). There is no final state, so we need a different notion of acceptance than we had with NFAs. Let \( \text{lim}(r) = \{ q : q = r_i \text{ for infinitely many } i \text{'s} \} \). That is, \( \text{lim}(r) \) is the set of states that appear infinitely often in run \( r \). Run \( r \) is accepting if \( \text{lim}(r) \cap F \neq \emptyset \), i.e., some accepting state is visited infinitely often. Automaton \( A \) accepts string \( s \) if there is an accepting run \( r \) of \( A \) on \( s \). The language of \( A \), denoted \( L(A) \), is the set of infinite strings accepted by \( A \).

(a) Show that if \( A \) and \( B \) are IIFAs, then there is an IIFA \( C \) such that \( L(C) = L(A) \cup L(B) \).
(b) Show that if \( A \) and \( B \) are IIFAs, then there is an IIFA \( C \) such that \( L(C) = L(A) \cap L(B) \).
(c) Are nondeterministic IIFAs more expressive than deterministic IIFAs? Provide a proof.