You have 90 minutes to complete this exam. You may state without proof any fact taught in class or assigned as homework.

1. Give a simple verbal description of the language recognized by the following NFA:

```
0
```

2. Draw NFAs for the following languages, taking full advantage of nondeterminism:

(a) binary strings in which every pair of 1s are separated by at least one 0;
(b) binary strings that have odd length or do not contain a 0.
3 Prove that the following languages are regular:

(2 pts) a. binary strings whose length is a prime number less than 100;
(2 pts) b. binary strings with precisely one occurrence of 01 as a substring;
(2 pts) c. binary strings of odd length in which the first and last symbols are not the same.
Let $L$ be a regular language over the binary alphabet. Let $L'$ be the set of strings obtained by taking a nonempty string in $L$ and flipping its first bit. Prove that $L'$ is regular.

Your friend draws a DFA for you and colors a subset of its states purple. Let $L$ be the set of strings $v$ such that the DFA never reaches a purple state while processing $v$. (For example, coloring the start state purple forces $L = \emptyset$.) Prove that $L$ is regular.
For a string $v$, a supersequence of $v$ is either $v$ itself or any string obtained by inserting one or more characters into $v$. For example, transit, site, snippet are supersequences of sit, but it is not. Prove that for any regular language $L$, the set of all supersequences of strings in $L$ is a regular language.

Fix a regular language $L$. Consider the language $L'$ of strings of the form $w_1w_2\ldots w_{2018}$, where $w_1, w_2, \ldots, w_{2018}$ are odd-length strings in $L$. Prove that $L'$ is regular.
SOLUTIONS
You have 90 minutes to complete this exam. You may state without proof any fact taught in class or assigned as homework.

(3 pts) 1 Give a simple verbal description of the language recognized by the following NFA:

![NFA Diagram]

**Solution:** binary strings without three consecutive 1s.

2 Draw NFAs for the following languages, taking full advantage of nondeterminism:

(2 pts) a. binary strings in which every pair of 1s are separated by at least one 0;
(2 pts) b. binary strings that have odd length or do not contain a 0.

**Solution.**

a.

![NFA Diagram for a]

b.

![NFA Diagram for b]
3. Prove that the following languages are regular:

(2 pts) a. binary strings whose length is a prime number less than 100;

(2 pts) b. binary strings with precisely one occurrence of 01 as a substring;

(2 pts) c. binary strings of odd length in which the first and last symbols are not the same.

Solution.

a. This language contains at most \(1 + 2 + 2^2 + 2^3 + \cdots + 2^{99} = 2^{100} - 1\) strings and therefore is finite. As shown in class, every finite language is regular.

b. This language is recognized by the following NFA:

![NFA Diagram](image)

c. This language corresponds to the intersection \(L' \cap L''\), where \(L'\) is the language of binary strings of odd length, and \(L''\) is the language of nonempty binary strings in which the first and last symbols are distinct. Then \(L'\) and \(L''\) are recognized by the following automata, respectively:

![Automata Diagram](image)

Since \(L'\) and \(L''\) are regular, the closure properties imply that the intersection \(L' \cap L''\) is also a regular language.
(3 pts) 4 Let $L$ be a regular language over the binary alphabet. Let $L'$ be the set of strings obtained by taking a nonempty string in $L$ and flipping its first bit. Prove that $L'$ is regular.

Solution. Let $D = (Q, \Sigma, \delta, q_0, F)$ be a DFA for $L$. A tempting, but incorrect, solution is to simply flip the labels of the transitions out of $q_0$. That approach fails because it potentially accepts the empty string and potentially flips bits other than the first. Instead, we will introduce a new start state $q_{\text{new}}$ that only interchanges the roles of 0 and 1 as far as the first symbol is concerned. Formally, $L'$ is recognized by the DFA $(Q \cup \{q_{\text{new}}\}, \Sigma, \Delta, q_{\text{new}}, F)$, where

$$
\Delta(q, \sigma) = \begin{cases} 
\delta(q_0, \neg \sigma) & \text{if } q = q_{\text{new}}, \\
\delta(q, \sigma) & \text{otherwise}.
\end{cases}
$$

(3 pts) 5 Your friend draws a DFA for you and colors a subset of its states purple. Let $L$ be the set of strings $v$ such that the DFA never reaches a purple state while processing $v$. (For example, coloring the start state purple forces $L = \emptyset$.) Prove that $L$ is regular.

Solution. If the start state is purple, then $L$ is empty and hence regular. If the start state is not purple, an NFA for $L$ can be obtained by making every state an accept state and erasing all transitions into purple states. The former modification guarantees the acceptance of every execution that avoids the purple states, whereas the latter modification disallows any execution that reaches a purple state.
For a string $v$, a supersequence of $v$ is either $v$ itself or any string obtained by inserting one or more characters into $v$. For example, transit, site, snippet are supersequences of sit, but it is not. Prove that for any regular language $L$, the set of all supersequences of strings in $L$ is a regular language.

**Solution.** Let $(Q, \Sigma, \delta, q_0, F)$ be a DFA for $L$. To recognize the supersequences of strings in $L$, simply add a self-loop to every state of the DFA on every symbol. The role of the self-loops is to absorb any symbols interjected into a string of $L$. Formally, the new automaton is the NFA $(Q, \Sigma, \delta', q_0, F)$, where $\delta'(q, \sigma) = \delta(q, \sigma) \cup \{q\}$.

Fix a regular language $L$. Consider the language $L'$ of strings of the form $w_1w_2\ldots w_{2018}$, where $w_1, w_2, \ldots, w_{2018}$ are odd-length strings in $L$. Prove that $L'$ is regular.

**Solution.** Let $\Sigma$ be the alphabet in question, and let $L_{\text{ODD}}$ be the language of odd-length strings over $\Sigma$. Then

$$L' = (L \cap L_{\text{ODD}})(L \cap L_{\text{ODD}})\cdots(L \cap L_{\text{ODD}}).$$

In this expression, $L$ is regular by hypothesis, whereas $L_{\text{ODD}}$ is regular because it is recognized by the DFA $(\{0, 1\}, \Sigma, \delta, 0, \{1\})$ with $\delta(q, \sigma) = (q + 1) \mod 2$. Since regular languages are closed under intersection and concatenation, it follows that $L'$ is regular as well.