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 NFA $(\Sigma, \delta, 0, \Sigma)$, where

$$\Sigma = \{0, 1, 2, \ldots, 9\},$$

$$\delta(q, \sigma) = \begin{cases} 
\emptyset & \text{if } \sigma = \varepsilon \text{ or } \sigma = q, \\
\{\sigma\} & \text{otherwise}.
\end{cases}$$

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

   (a) binary strings in which there is a pair of 1s separated by an odd number of 0s;
   (b) binary strings that have length at most 2, or have a 0 in the third position from the end.
3 Prove that the following languages are regular:

(2 pts) **a.** binary strings that contain as a substring 1111 or 0110 but not both;

(2 pts) **b.** binary strings in which the substring 01 occurs an even number of times;

(2 pts) **c.** strings of the form $a_1 b_1 a_2 b_2 \ldots a_k b_k$ for some $k$ and some $a_1, b_1, a_2, b_2, \ldots, a_k, b_k \in \{0, 1\}$, where $a_1 a_2 \ldots a_k$ strictly precedes $b_1 b_2 \ldots b_k$ in lexicographic order.
Let $L$ be a given regular language. Let $L'$ be the set of strings $w$ such that $w^R w \in L$. Prove that $L'$ is regular.

An edit operation on a string is the insertion of a new symbol or the deletion of an existing symbol. For a language $L$, define $\text{close}(L)$ to be the set of strings $w$ such that $w$ can be obtained from some string in $L$ using at most 2017 edit operations. Prove that $\text{close}(L)$ is regular whenever $L$ is regular.
For a language $L$, let $L^\dagger$ denote the set of strings $w$ such that every prefix of $w$ is in $L$. Prove that $L^\dagger$ is regular whenever $L$ is regular.

Describe an algorithm that takes as input a DFA with the binary alphabet and determines whether the DFA accepts every string with more than 2017 ones. Your algorithm must run in finite time.
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 NFA \((\Sigma, \Sigma, \delta, 0, \Sigma)\), where 
\[
\Sigma = \{0, 1, 2, \ldots, 9\},
\]
\[
\delta(q, \sigma) = \begin{cases} 
\emptyset & \text{if } \sigma = \varepsilon \text{ or } \sigma = q, \\
\{\sigma\} & \text{otherwise.}
\end{cases}
\]

Solution: the empty string, or any positive integer in which no pair of consecutive digits are the same.

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

(a) binary strings in which there is a pair of 1s separated by an odd number of 0s;

(b) binary strings that have length at most 2, or have a 0 in the third position from the end.

Solution.

a.

b.
Prove that the following languages are regular:

(a) binary strings that contain as a substring 1111 or 0110 but not both;
(b) binary strings in which the substring 01 occurs an even number of times;
(c) strings of the form $a_1b_1a_2b_2\ldots a_kb_k$ for some $k$ and some $a_1, b_1, a_2, b_2, \ldots, a_k, b_k \in \{0, 1\}$, where $a_1a_2\ldots a_k$ strictly precedes $b_1b_2\ldots b_k$ in lexicographic order.

\section*{Solution.}

(a) For a string $w = w_1w_2\ldots w_n$, let $L_w$ denote the language of strings that contain $w$ as a substring. This language is regular because it is recognized by the following NFA:

In particular, $L_{1111}$ and $L_{0110}$ are regular. Since regular languages are closed under union, intersection, and set difference, the language $(L_{1111} \cup L_{0110}) \setminus (L_{1111} \cap L_{0110})$ in the problem statement is regular as well.

(b) This language is recognized by the following DFA:

(c) This language is recognized by the following NFA:
Let $L$ be a given regular language. Let $L'$ be the set of strings $w$ such that $w^Rw \in L$. Prove that $L'$ is regular.

**Solution.** Let $D = (Q, \Sigma, \delta, q_0, F)$ be a DFA for $L$. We can check if $w \in L'$ using the following nondeterministic procedure. Pick a state $q \in Q$ nondeterministically, and start with two fingers at that state (that’s our guess at the DFA’s end state on reading $w^R$). Then read $w$ symbol by symbol, simulating the operation of the DFA at each step by moving the first finger backward and the second finger forward from their current locations. When you have processed the last symbol of $w$, accept if and only if the first finger is at the start state and the second finger at an accept state. This solution can be formalized as the NFA $((Q \times Q) \cup \{q_{\text{start}}\}, \Sigma, \Delta, q_{\text{start}}, \{q_0\} \times F)$, where

\[
\begin{align*}
\Delta(q_{\text{start}}, \epsilon) &= \{(q, q) : q \in Q\}, \\
\Delta(q_{\text{start}}, \sigma) &= \emptyset \text{ for all } \sigma \in \Sigma, \\
\Delta((q, q'), \epsilon) &= \emptyset \text{ for all } q, q' \in Q, \\
\Delta((q, q'), \sigma) &= \{(q'', q''') : \delta(q'', \sigma) = q \text{ and } \delta(q', \sigma) = q'''} \text{ for all } q, q' \in Q, \sigma \in \Sigma.
\end{align*}
\]

An *edit operation* on a string is the insertion of a new symbol or the deletion of an existing symbol. For a language $L$, define close($L$) to be the set of strings $w$ such that $w$ can be obtained from some string in $L$ using at most 2017 edit operations. Prove that close($L$) is regular whenever $L$ is regular.

**Solution.** Observe that

\[
\text{close}(L) = \bigcup_{i+j \leq 2017} \text{skip}(...) \text{skip} \text{skip} \text{add}(...) \text{add} \text{add}(L),
\]

where the union is over all nonnegative integers $i$ and $j$ that sum to at most 2017. As proved in class, regular languages are closed under the add and skip operations. Therefore, each of the languages in the finite union above is regular. Since regular languages are closed under finite unions, close($L$) is regular.
For a language $L$, let $L^\downarrow$ denote the set of strings $w$ such that every prefix of $w$ is in $L$. Prove that $L^\downarrow$ is regular whenever $L$ is regular.

**Solution.** To obtain an NFA for $L^\downarrow$, start with a DFA for $L$ and simply remove the outgoing transitions from every reject state. That way, when the new automaton encounters a reject state (indicating a rejected prefix) as it works through an input string, the automaton gets stuck and cannot reach an accept state from then on.

Describe an algorithm that takes as input a DFA with the binary alphabet and determines whether the DFA accepts every string with more than 2017 ones. Your algorithm must run in finite time.

**Solution.** Let $D_1$ be the input DFA. The language of strings with more than 2017 ones is recognized by the DFA $D_2 = (\{0, 1, 2, \ldots, 2018\}, \{0, 1\}, \delta, 0, \{2018\})$, where the transition function is $\delta(i, \sigma) = \min\{i + \sigma, 2018\}$. To rephrase the problem, we are to check whether $D_1$ rejects some string accepted by $D_2$. For this, swap the accept and rejects states of $D_1$ to obtain a DFA $\overline{D_1}$ for the complementary language. Then, use the Cartesian product construction from class to obtain a DFA $D$ that recognizes the intersection of the languages of $\overline{D_1}$ and $D_2$. Finally, check to see if $D$ has a path from the start state to an accept state. This last step can be done either by trying out all candidate paths in a brute force manner, or by using an efficient graph algorithm that you have encountered in CS 180, such as depth- or breadth-first search.