**Problem 1**

(25 points) Give the state diagram of a finite automaton recognizing the following language. The alphabet is \{0, 1\}.

\[ \{ w | w \text{ has length exactly 3 and its last symbol is different from its first symbol} \} \]

Construction idea: We condition on the first input symbol and go to \(q_0\) or \(q_1\). The second symbol can be arbitrary, so when in \(q_2\) and \(q_3\) we have seen two symbols with the first being 0 and 1 respectively, so we should travel to the accept state \(q_4\) on 1 and 0 respectively (so that the last [3rd] symbol is different from the first). Otherwise we travel to \(q_5\) which is a rejecting self loop on all input (e.g. a 'garbage state' in that once you are on \(q_5\) you can never again get to an accept state). Also, if we are on the accept state but we see another symbol, then we have seen a total of 4 symbols and should reject, by transitioning on all input to state \(q_5\).

**Problem 2**

(25 points) Give a finite automaton (both a state diagram and a formal description) recognizing the following language. The alphabet is \{0, 1\}.

\[ \{ w | w \text{ is not the empty string and every odd position of } w \text{ is a 1} \} \]

Construction idea: We condition on the first input symbol and go to \(q_0\) or \(q_1\). The second symbol can be arbitrary, so when in \(q_2\) and \(q_3\) we have seen two symbols with the first being 0 and 1 respectively, so we should travel to the accept state \(q_4\) on 1 and 0 respectively (so that the last [3rd] symbol is different from the first). Otherwise we travel to \(q_5\) which is a rejecting self loop on all input (e.g. a 'garbage state' in that once you are on \(q_5\) you can never again get to an accept state). Also, if we are on the accept state but we see another symbol, then we have seen a total of 4 symbols and should reject, by transitioning on all input to state \(q_5\).
Formal definition:
\[ M = (Q, \Sigma, \delta, q_s, F) \]
where
\[ Q = \{q_0, q_1, q_2, q_3\} \]
\[ \Sigma = \{0, 1\} \]
\[ q_s = q_0 \]
\[ F = \{q_1, q_2\} \]

\[ \delta(q, \sigma) = \begin{cases} 
q_3 & \text{if } \sigma = 0 \text{ and } q = q_0 \\
q_1 & \text{if } \sigma = 1 \text{ and } q = q_0 \\
q_2 & \text{if } \sigma = 0 \text{ and } q = q_1 \\
q_2 & \text{if } \sigma = 1 \text{ and } q = q_1 \\
q_3 & \text{if } \sigma = 0 \text{ and } q = q_2 \\
q_1 & \text{if } \sigma = 1 \text{ and } q = q_2 \\
q_3 & \text{if } \sigma = 0 \text{ and } q = q_3 \\
q_3 & \text{if } \sigma = 1 \text{ and } q = q_3 
\end{cases} \]

Problem 3

(25 points) Show that the following language is regular, with the alphabet being \{0, 1\}.
\[ \{w | w \text{ contains an equal number of occurrences of the substrings 01 and 10}\} \]

Hint: First find an equivalent and simpler characterization of the language.

To formally prove the correctness of this construction, we must show that \( L(M) = A \), where \( M \) is the constructed DFA and \( A \) is the language described in the problem.

Proof. (\( L(M) = A \)) We prove this conjecture by showing \( L(M) \subseteq A \) and also \( A \subseteq L(M) \), which implies that \( L(M) = A \).

We first need to argue that the number of ’01’ and ’10’ substrings of a binary string are equal iff the first and last symbol of the string are equal.

Lemma 1. Let \( w \) be an arbitrary length \( n \) binary string. Define mappings from \( \{0, 1\}^* \) to \( \mathbb{N} \): \( n_{10}(w) = (\# \text{ of ’10’ substrings in } w) \) and \( n_{01}(w) = (\# \text{ of ’01’ substrings in } w) \). We conjecture that \( |n_{01}(w) - n_{10}(w)| \leq 1 \forall \ w \in \{0, 1\}^* \), and that \( n_{01}(w) = n_{10}(w) \) if and only if \( w_0 = w_n \) (first and last symbol same).
Proof. Create $w'$ from $w$ by replacing all strings of consecutive 0s in $w$ by a single 0 and all strings of consecutive 1s in $w$ by a single 1. Now consider all overlapping pairs $(w'_0, w'_1), (w'_1, w'_2), \ldots, (w'_{n-2}, w'_{n-1}), (w'_{n-1}, w'_n)$. Each pair is a '01' or a '10' substring since there are no consecutive symbols, and adjacent pairs cannot be the same substring because then the inner symbol would be 0 and 1 at the same time (e.g. $(w_{i-1}, w_i)$ and $(w_i, w_{i+1})$ cannot be '01' and '01' or '10' and '10'). Therefore it must be that $|n_{01}(w') - n_{10}(w')| \leq 1$, and since $w$ is simply $w'$ with added strings of consecutive symbols, which would not affect the number of '01' or '10' substrings, we conclude that $|n_{01}(w) - n_{10}(w)| \leq 1$. Now note that in the special case that $w_0 = w_n$, then $w'_0 = w'_n$, so one of $(w'_0, w'_1)$ and $(w'_{n-1}, w'_n)$ is a '01' substring and one is a '10' substring. Since all adjacent overlapping pairs are different, $n_{10}(w') = n_{01}(w')$ and therefore $n_{10}(w) = n_{01}(w)$. Conversely, if $n_{10}(w) = n_{01}(w)$, then $n_{10}(w') = n_{01}(w')$, which means there must be an even number of overlapping pairs. Since adjacent pairs are different substrings, the first and last pairs must be different, meaning one is '01' and one is '10'. Either way, $w'_0 = w'_n$ and therefore $w_0 = w_n$. \qed

Now we use the lemma to prove the main result.

- $(L(M) \subseteq A)$ Suppose $w \in L(M)$ where $w$ is a binary string. Then $w$ is accepted by machine $M$, which means $\exists$ a sequence of states $r = r_0, \ldots, r_n$ in $Q(M)$ (set of states of machine $M$) such that $r_0 = q_0, \delta(r_i, w_{i+1}) = r_{i+1}$, for $i = 0, \ldots, n - 1$, and $r_n \in F$. Suppose WLOG $w_0 = 1$, then $r_1 = q_1$. Also, $r_2, \ldots, r_n \in \{q_1, q_2\}$ because those are the only states reachable from $q_1$. But $r_n \in F$, so $r_n \in F \cap \{q_1, q_2\} = \{q_1\}$ so $r_n = q_1$. This means $w_n = '1'$, since $q_1$ is only reachable by transitions on '1'. By the lemma, since $w_0 = w_n = 1$ we know that $n_{01}(w) = n_{10}(w)$ so $w \in A$. A similar argument can be made for when $w_0 = 0$, where you can argue that $w_n = 0$ and then use the lemma.

- $(A \subseteq L(M))$ Suppose $w \in A$. Assume WLOG $w_0 = 1$. Create a sequence of states $\{r_i\}_{i=1}^n$ where $r_i = q_1$ if $w_i = 1$ and $r_i = q_2$ if $w_i = 0$. We now take $r_0 = q_0$ and see that $r_0$ is the start state of $M$, $\delta(r_i, w_{i+1}) = r_{i+1}$ $\forall i = 1, \ldots, n - 1$ which was forced on the created sequence. Finally since $w \in A$, which implies $w_0 = w_n$ by the lemma, we have that $w_n = 1$ and therefore $r_n = q_1 \in F$. This means that $M$ accepts $w$ so $w \in L(M)$.

\qed

Problem 4

(25 points) For any string $w = w_1w_2 \ldots w_n$, the reverse of $w$, written as $w^R$, is the string $w$ in reverse order, $w_n \ldots w_2w_1$. For any language $A$, let $A^R = \{w^R \mid w \in A\}$. Show that if $A$ is regular, then so is $A^R$.

Assume that $A$ is regular. Then there exists an NFA $M = (Q, \Sigma, \delta, q_s, F)$ such that $L(M) = A$. Produce a new NFA $M'$ using the following process:

1. Let $M' = M = (Q', \Sigma, \delta', q'_s, F')$
2. Set $Q' = Q \cup \{q_{new}\}$
3. Set $F' = \{q_s\}$
4. Set $q'_s = q_{new}$
5. Remove each transition rule $\delta'(q, \sigma) = p$ and replace it with $\delta'(p, \sigma) = q$ for $\sigma \in \Sigma \cup \epsilon$ (e.g. reverse all the arrows).
6. $\forall q \in F$ set $\delta'(q_{new}, \epsilon) = q$

We now argue that $L(M') = A^R$. 
Proof. We use the strategy of proving both $L'(M') \subset A^R$ and $A^R \subset L'(M')$

- $(L'(M') \subset A^R)$ Assume $w \in L'(M')$ then $\exists$ a sequence of states $r = q'_s, r_0, \ldots, r_n$ in $Q'$ such that the first state is $q'_s$, $\delta'(r_i, w_{i+1}) = r_{i+1}$, for $i = 0, \ldots, n-1$ (also $\delta'(q'_s, \epsilon) = r_0$), and $r_n \in F'$. But $F' = \{q_s\}$ so $r_n = q_s$, and $\delta'(r_i, w_{i+1}) = r_{i+1}$ implies $\delta(r_{i+1}, w_{i+1}) = r_i$ because $\delta'$ was constructed by reversing transitions of $\delta$. Finally $q'_s$ has an epsilon transition to each $q \in F$, so $r_0 \in F$. Therefore by reversing the sequence $r$ (without the leading $q'_s$) we have a computation for $w^R$ on machine $M$, so $M$ accepts $w^R$, so $w^R \in A$, so $w^R = w \in A^R$

- $(A^R \subset L'(M'))$ Assume $w \in A^R$. Then $w^R \in A$ which means $w^R \in L(M)$, so there $\exists$ a sequence of states $r = r_0, \ldots, r_n$ in $Q$ such that $r_0 = q_s$, $\delta(r_i, w_{n-i+1}) = r_{i+1}$, for $i = 0, \ldots, n-1$, and $r_n \in F$. Following the construction above we can find a new sequence $r'_0, \ldots, r'_n$ where $r'_0 \in F, r'_n \in F'$, and $\delta'(r'_i, w_{i+1}) = r'_{i+1}$. By taking $r' = r^R$ e.g. the reversal of $r$. Finally we add $q'_s$ to the beginning of this sequence and note that $\delta'(q'_s, \epsilon) = r'_0$ since $r'_0 \in F$. Now the full sequence $q'_s, r'_0, \ldots r'_n$ is a computation for the string $w$ on machine $M'$, so $w \in L(M')$. 

$\square$