All solutions should be neatly written or type set. All major steps in proofs must be justified. Please start each problem solution on a new page and put your name on every page.

1. (10 points) This problem is designed to strengthen your ability to prove facts by induction. The reversal $w^R$ of a string $w$ can be defined recursively in the following way.

\[
\varepsilon^R = \varepsilon \\
(ax)^R = ax^R
\]

where $a \in \Sigma$.

Prove the following: For all strings $x$ and $y$ over $\Sigma$, $(xy)^R = y^Rx^R$. For this your proof should be by induction on the length of $y$. You may use recursive definition of reversal and any basic facts about concatenations such as associativity and the identity properties of $\varepsilon$. That is: $x(yz) = (xy)z$ and $\varepsilon x = x \varepsilon = x$ for all strings $x, y, z$.

2. (10 points) Design deterministic finite automata using a state transition diagram for each of the following languages.

(a) $\{x \in \{0, 1\}^* : 101 \text{ is a substring of } x\}$.
(b) $\{x \in \{0, 1\}^* : 111 \text{ is not a substring of } x\}$.
(c) $\{x \in \{0, 1\}^* : x \text{ contains exactly 5 0's}\}$.
(d) $\{x \in \{0, 1\}^* : x \text{ has an odd number of 0's or an even number of 1's}\}$.

3. (10 points) Consider the languages

\[
L_k = \{x \in \{0, 1\}^* : x \text{ contains exactly } k \text{ 0's}\}
\]

for $k \geq 0$. 


(a) Formally define a deterministic finite automaton $M_k$ with exactly $k + 2$ states that accepts $L_k$.

(b) Prove by contradiction that every deterministic finite automaton that accepts $L_k$ has at least $k + 2$ states. The ideas from problem 1 of assignment 1 are useful.