1. **Regular or not?** Prove or disprove that each of the languages below is regular (or not). Let $\Sigma^+ \text{ denote the set of all nonempty strings over alphabet } \Sigma; \text{ in other words, } \Sigma^+ = \Sigma \cdot \Sigma^*$. Denote $n(w)$ the integer corresponding to the binary string $w$.

(a) $\{ 3x=y : x, y \in \{0, 1\}^+, n(y) = 3n(x) \}$

(b) $\{ \frac{3x}{y} : \frac{x}{y} \in \{0, 0, 1, 1\}^* \}, n(y) = 3n(x) \}$

(c) $\{ wxw^R : w, x \in \Sigma^+ \}$

(d) $\{ ww^R x : w, x \in \Sigma^+ \}$

[Hint: To prove that a language $L$ is regular, construct an NFA that recognizes $L$; to disprove that $L$ is regular, construct a fooling set for $L$ and argue that the construction is correct.]

2. **Telling DFAs apart.**

Let $M_1$ and $M_2$ be two DFAs, each with exactly $n$ states. Assume that the languages associated with the two machines are different (that is, $L(M_1) \neq L(M_2)$), there is always some string in the symmetric difference of the two languages.

Prove that there is a string $w$ of length polynomial in $n$ in the symmetric difference of $L(M_1)$ and $L(M_2)$. What is the best upper bound you can get on the length of $w$?

3. **Telling strings apart.**

Let $w_1$ and $w_2$ be two strings over binary alphabet $\Sigma = \{0, 1\}$, each of length exactly $n$. Assume that the two strings are different, there is always some DFA that accepts exactly one of the two strings.

Prove that there is a DFA $M$ of size $o(n)$ such that exactly one of $w_1$ and $w_2$ is in $L(M)$. What is the best upper bound you can get on the size of $M$?