Homework 2
CS611: Theory of Computation

Instructions: This homework has three problems that can be solved in a group of at most two, but **individually** working on these is highly recommended to make sure you understand everything. In addition, some problems that will not be graded are given to you for practicing, you don't need to submit solutions for these OPTOINAL and Not Graded problems.

Recommended Reading: Lectures 3, 4.

Problem 1. [Category: Design] Design a DFA for the language $L_{A3} = \{w \in \{a,b\}^* \mid \text{ if } w \text{ starts with an } a$ then it does not end with a $b\}$. **[5 points]**

Problem 2. [Category: Design] Design a DFA for the language $L_{A1} = \{w \in \{a,b\}^* |$ number of a's in w is not divisible by $3\}$. **[5 points]**

Problem 3. [Category: Design] Design a DFA for the language $L_{A4} = \{w \in \{a,b\}^* \mid ba$ appears exactly twice as a substring$\}$. **[5 points]**

Problem 4. [Category: Design+Proof] Let $A_k \subseteq \{a,b\}^*$ be the collection of strings w where there is a position i in w such that the symbol at position i (in w) is a, and the symbol at position $i+k$ is b. For example, consider A_2 (when $k=2$). $baab \in A_2$ because the second position ($i=2$) has an a and the fourth position has a b. On the other hand, $bb \notin A_2$ (because there are no as) and $aba \notin A_2$ (because none of the as are followed by a b 2 positions away).

1. Design a DFA for language A_k, when $k=2$, you just need to draw the transition diagram. **[5 points]**
2. Design an NFA for language A_2 that has at most 4 states. You need not prove that your construction is correct, but the intuition behind your solution should be clear and understandable. **[5 points]**
3. (Optional) Prove that the DFA you give in part 1 is correct , that is the DFA recognize the language A_K when $k=2$. **[5 points]**
4. (Optional) Design a DFA for language A_k using formal definition (by listing states, transitions, etc. and not "drawing the DFA"), the formal definition will depend on the parameter k but should work no matter what k is; see lecture 3 for such an example.
5. (Optional) Prove that any DFA recognizing A_2 has at least 5 states. **[5 points]**

Problem 5. [Category: Proof] In lecture 3, we have give partial proof for the DFA designed for language M_{odd}. Recall that we have four parts together as the whole statement, we have proved Base case for all four parts, and also gave the Induction Hypothesis for all four parts, but for Inductive Step, we only gave complete proof for part (a). In this problem, please give the Inductive Step for part (b). **[10 points]**

Not Graded Problems

Problem 6. [Category: Design+Proof] For a string $w \in \Sigma^*$, let w^R denote the reverse of w, i.e., if $w = w_1w_2\cdots w_n$, where $w_i \in \Sigma$ then $w^R = w_nw_{n-1}\cdots w_1$. For a language L, let $L^R = \{w^R \mid w \in L\}$. Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA.

1. Design a *DFA* M^R that recognizes $\mathbf{L}(M)^R$, i.e., $\mathbf{L}(M^R) = (\mathbf{L}(M))^R$. **[5 points]**
2. Prove that your DFA M^R in the previous part is correct. **[5 points]**