Introduction to Theoretical Computer Science, Fall 2024
Assignment 2 (Due September 23 Monday 4:00 pm)

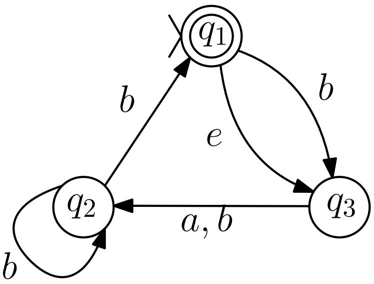
Only the problems in part I will be graded.

1 Part I

Q1. Let M be an arbitrary NFA. Let M' be the NFA obtained from M by exchanging the role of final and non-final states. Is it always true that $L(M) \cap L(M') = \emptyset$? Give a proof or a counter-example.

Q2. Let $L = \{w \in \{a, b, c\}^* : |w| \geq 1$ and the last symbol of w has appeared at least twice in $w\}$. Construct a NFA to accept L. Your NFA should have no more than 5 states.

Q3. Convert the following NFA to an equivalent DFA. Give only the portion of the DFA that is reachable from the initial state.



2 Part II

Q4. Design a NFA to accept the following languages. Your NFA should have at most 4 states.

$$\{w \in \{0, 1\}^* : w \text{ has a pair of 1's that are separated by odd number of symbols}\}$$

Q5. Let A and B be two regular languages over some alphabet Σ.

(a) Let $M = (K, \Sigma, \delta, s, F)$ be a DFA that accepts A. Use M to construct a FA that accepts the following language C. (Hint: To determine whether a given string w is in C, basically you should test whether it is possible for M to go from the initial state to some final state after taking $|w|$ steps. How to do this? You may utilize NFA's ability to make right guess.)

$$C = \{w \in \Sigma^* : |w| = |u| \text{ for some } u \in A\}$$

(b) Use the conclusion of (a) to show that the following language is regular. (Hint: you may use the closure properties of regular languages)

$$D = \{w \in B : |w| = |u| \text{ for some } u \in A\}$$