Assignment 3
CSCI-661 Foundations of CS Theory
due Monday, February 19, 2024

1. **(4 points)** Draw the state diagram of an NFA accepting the language of all strings over $\{a, b\}$ that end with the substring *bbab* or that start with an even number of a's. Use non-determinism to make your state diagram as simple as possible.

2. **(4 points)** Use the closure properties of regular languages to prove that $\{a^i b^k \mid i \neq k\}$ is not regular. You can use the fact that $\{a^i b^i \mid i \geq 0\}$ is not regular. Do not use any other nonregular languages in your proof.

3. **(4 points)** Draw the state diagram of an NFA accepting the language of all strings over $\{a, b\}$ that end with the substring *abba* or that start with an odd number of b's. Use non-determinism to make your state diagram as simple as possible.

4. **(6 points)** Let $N = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_1\})$ be an NFA with δ given by the following table:

	0	1	ϵ
q_0	$\{q_2\}$	$\emptyset$	$\{q_1\}$
q_1	$\{q_0\}$	$\emptyset$	$\emptyset$
q_2	$\{q_1\}$	$\{q_1, q_2\}$	$\emptyset$

   (a) Draw the state diagram of N.

   (b) Use the subset construction to construct an equivalent DFA M. Draw the state diagram of M. You must label the states. You do not have to draw unreachable states of M. Do not otherwise simplify M.

5. **(4 points)** Suppose L is a language over Σ. Define a new language $chop(L)$ that is formed by removing the first symbol from every string in L. That is,

   $chop(L) = \{w \in \Sigma^* \mid vw \in L \text{ and } |v| = 1\}$.

   You will show that if L is a regular language, then $chop(L)$ is also a regular language. Specifically, let DFA $M = (Q, \{a, b\}, \delta, q_0, F)$ be given. Define the formal components of an NFA $N = (Q', \{a, b\}, \delta', q_0', F')$ such that $L(N) = chop(L(M))$.

6. **(4 points)** For $N = (Q, \Sigma, \delta, q_0, F)$ an NFA, define NFA $\widehat{N}$ as: $\widehat{N} = (Q, \Sigma, \delta, q_0, Q-F)$. Recall that with a DFA, we can recognize the complement of the language of the DFA by toggling the accepting and rejecting states. This question considers whether we achieve the same result if we toggle the accepting and rejecting states of an NFA.

   Consider the following statements.

   (a) For every NFA, N, $L(\widehat{N}) \subseteq \overline{L(N)}$.

   This statement is TRUE / FALSE (circle your answer).

   If you answered "TRUE," informally and briefly explain your answer. If you answered "FALSE," give a simple counterexample (draw the state diagrams of N and $\widehat{N}$ and say what $L(N)$ and $L(\widehat{N})$ are).

   (b) For every NFA, N, $L(\widehat{N}) \supseteq \overline{L(N)}$.

   This statement is TRUE / FALSE (circle your answer).

   If you answered "TRUE," informally and briefly explain your answer. If you answered "FALSE," give a simple counterexample (draw the state diagrams of N and $\widehat{N}$ and say what $L(N)$ and $L(\widehat{N})$ are).

7. **(7 points)** Let $L_k = \{w \in \{a, b\}^* \mid w \text{ contains a substring having } k \text{ more } b\text{'s than } a\text{'s}\}$.

   (a) Although at first glance this may seem to be a non-regular language, it is in fact a regular language. Give the state diagram of an NFA that recognizes L_3 using the following design:

   - Loop continuously in the start state for any symbol read.
   - Any time a b symbol is read, also spawn a new branch that tests whether a substring beginning with this symbol reaches 3 more b's than a's at any point.
   - Any such spawned branch that hasn't yet reached this accepting state can be allowed to die out if it would otherwise return to an equal number of a's and b's (there's no need to retain the extra branch when there are equal a's and b's or an excess of a's; a new branch will take over from the start state if another b is read).

   (b) Give a 5-tuple specifying NFA N_k such that $L(N_k) = L_k$. (Assume $k \geq 3$).