Assignment 4
CSCI-661 Foundations of Computer Science Theory
due Thursday, February 29, 2024

1. **(5 points)** Let $R = (00^* \cup 1)^*\ 00$. Use the construction from the proof of Lemma 1.55 (given any regular expression, we can construct an NFA that recognizes the described language) to construct an NFA N such that $L(N) = L(R)$. Apply the construction literally (do not optimize the resulting NFA—keep all those ϵ arrows in the NFA). You should only draw the final NFA.

2. **(6 points)** Follow the construction from the proof of Lemma 1.60 (given any DFA, we can determine a regular expression that describes the language of the DFA) to generate a regular expression for the DFA $M = (\{q_0, q_1\}, \{a, b\}, \delta, q_0, \{q_0\})$ where $\delta(q_i, a) = q_0$ and $\delta(q_i, b) = q_1$, for $i \in \{0, 1\}$.

   (a) Draw the initial corresponding GNFA.

   (b) Draw a corresponding GNFA after removing one state.

   (c) Give the final regular expression.

   As in Sipser's Example 1.66, you do not have to draw GNFA arrows labeled $\emptyset$, even though they are present. You may also simplify the regular expressions on the transitions.

3. **(4 points)** Give regular expressions for the following languages:

   (a) The language of all strings over $\{a, b\}$ except the empty string.

   (b) The language of all strings over $\{a, b\}$ that contain both bab and bba as substrings.

   (c) Language L_3 from question 7 of the previous homework. That is,
   $L_k = \{w \in \{a, b\}^* \mid w \text{ contains a substring having 3 more } b\text{'s than } a\text{'s}\}$.

   (d) The language of all strings over $\{a, b\}$ that have a b in every odd position (first symbol is considered position 1; empty string should be accepted) or start with a single a.

4. **(4 points)** Let $L = \{w \in \{a, b\}^* \mid w \text{ contains at least 2 more } a\text{'s than } b\text{'s}\}$. Use Myhill-Nerode to prove that L is not regular.

5. **(10 points)** Let $L = aa^*bb^* \cup b\Sigma^*$.

   (a) Draw a minimal DFA M for L. (Note: if your DFA is not minimal, it will become clear below when you try to show - and can't - that all the strings in your index set X are pairwise distinguishable.)

   (b) How many states does a minimal DFA for L have?

(c) What is the value of the index of L?

(d) How many equivalence classes does $\equiv_L$ have?

(e) Give a largest set X of strings that is pairwise distinguishable by L. (Note: You can read these off from your minimal DFA. For every state q in M, put one string x with $\delta(q_0, x) = q$ in X.)

(f) Show that X is pairwise distinguishable by L.

(g) Describe the equivalence classes of $\equiv_L$. Give a simple regular expression for each equivalence class. (Note: You can find the equivalence classes in your minimal DFA. For every state q in M, $\{x \mid \delta(q_0, x) = q\}$ is an equivalence class.)

6. **(7 points)** Given the following DFA $M = (\{A, B, C, D, E, F\}, \{0, 1\}, \delta, A, \{A, E\})$, where transition function δ is defined as follows:

δ	0	1
A	B	D
B	C	B
C	C	E
D	C	D
E	D	F
F	C	F

(a) Draw the state diagram for the given DFA.

(b) Using the algorithmic DFA minimization technique (not Myhill-Nerode), determine distinguishable and indistinguishable states. Show your table that indicates distinguishability (or lack thereof).

(c) Draw the state diagram for the minimal DFA for this language. Label your states so that it is clear which states were combined from the original machine.