Name: YOUR NAME HERE

CS46 lab 4

This lab assignment is due at 11:59PM on Tuesday, 17 February. Write your solution using LaTeX. Submit this assignment using **github**. There are total of **16 points** for this lab.

This homework has 2 parts:

- part 1 is individual
- part 2 may be completed with a partner

For part 1 your work should be your own. For part 2, your discussions should be just with your partner. Your partnership's write-up is your own: do not share it, and do not read other teams' write-ups. If you use any out-of-class references (anything except class notes, the textbook, or asking the instructor), then you **must** cite these in your post-lab survey. Please refer to the course webpage or ask me any questions you have about this policy.

The main **learning goal** of this labs is to work with NFAs, regular expressions. As always, we shall continue to monotonically improve our proof-writing, clarity, and organization skills.

1 Automata Tutor problems

These problems should be completed[1] on Automata Tutor. You are allowed three attempts at each problem. I recommend that you *first* try to solve the problems on paper, *then* use the site to debug your solutions.

1. Let $\Sigma = \{a, b\}$, let $L_1 = \{w \mid \text{the length of } w \text{ is even}\}$, and let $L_2 = \{w \mid w \text{ begins and ends with } a\}$.
   (a) Construct an NFA for the language $L = L_1 \cup L_2$.
   (b) Construct an NFA for $L_1 \circ L_2$.
2. Write a regular expression for the following language over the alphabet $\Sigma = \{a, b\}$:

   $$\{w \mid w \text{ contains exactly two } a\text{s or at least two } b\text{s}\}$$

   Ponder the fact that union is pretty easy with NFAs and with regular expressions.
3. Let $\Sigma = \{0, 1\}$ and $L = \{w \mid w \text{contains the substring } 0ab0 \text{ or } 1ab1 \text{ where } a, b \in \Sigma\}$.
   (a) Construct a regular expression for L.
   (b) Construct a DFA that recognizes L.
   (It is *strongly* recommended that you plan on paper first!)
   (c) Construct an NFA that recognizes L.
   Your NFA should have substantially fewer states than your DFA. (Phew!)

[1] If you want to use late days on this assignment, you will need to submit solutions to these problems via github. The automatatutor site has only one deadline.

4. Let $L = \{w \mid$ if w contains an a, then it contains at least three as in a row$\}$ over $\Sigma = \{a, b\}$. So for example, L contains $abaaa$, $bbaabaaabaa$, ε, and bb. L does *not* contain baa, $abaaba$, or $bbba$.

   (a) Construct a DFA that recognizes L.

   (b) Construct an NFA that recognizes L.

   (c) Construct a regular expression that recognizes L.

5. Let $\Sigma = \{a, b\}$ and let $L = \{w \mid w$ contains an even number of as and an odd number of bs$\}$.

   (a) Construct an NFA recognizing L^*.
   (Hint: You can use the construction of Theorem 1.49 to get a correct answer, but this NFA will be capable of being simplified. Try describing L^* in English first.)

   (b) Think for a little while about a regular expression for L (not L^*). Ponder the fact that intersection was much easier in automata than in regular expressions.

   You might try to approach this problem by starting with a DFA and converting it to a regular expression, or by coming up with a regular expression based on your intuition about what sort of strings are in this language.

2 Written problems

These problems should be typeset in LaTeXand submitted using **github**.

1. **Understanding regular expressions.**

   For each of the following regular expressions over $\Sigma = \{a, b\}$, explain in English what language they describe. (Show your thought process.)

   (a) $b^*a(b^*a^*)^*$

   (b) $(a^*\emptyset b \cup ab \cup b^*\emptyset^*a)(b \cup \emptyset)$

   (c) $\epsilon \cup a(ba)^* \cup b(ab)^*$

2. **Regular expression identities.**

   Let R and S be regular expressions. Prove or disprove the following "identities". To prove the identity, you must argue that a string in the language defined on the left-hand side is in the language defined on the right-hand side, and vice versa. To disprove the identity, you must give a small counterexample string with real examples of regular expressions for R and S.

   (a) $(R^*)^* = R^*$

   (b) $(R \cup S)^* = R^* \cup S^*$

   (c) $(R^*S^*)^* = (R \cup S)^*$