CSE 105
Group Homework 1
Due: Saturday April 14, 2018 at 11:00PM (on Gradescope)

INSTRUCTIONS
One member of the group should upload your group submission to Gradescope. During the submission process, they will be prompted to add the names of the rest of the group members. All group members' names and PIDs should be on **each** page of the submission.

Your homework must be typed. We recommend that you submit early drafts to Gradescope so that in case of any technical difficulties, at least some of your work is present. You may update your submission as many times as you'd like up to the deadline.

Your assignments in this class will be evaluated not only on the correctness of your answers, but on your ability to present your ideas clearly and logically. You should always explain how you arrived at your conclusions, using mathematically sound reasoning. Whether you use formal proof techniques or write a more informal argument for why something is true, your answers should always be well-supported. Your goal should be to convince the reader that your results and methods are sound.

READING Sipser Section 1.3, 1.1

KEY CONCEPTS Strings, languages, union, concatenation, Kleene star, regular expressions, language described by a regular expression, deterministic finite automata (DFAs), computations, language recognized by a DFA.

1. (**10 points**) Regular expressions are built up using three operations: union ($\cup$), concatenation ($\circ$ or placing regular expressions next to one another), and Kleene star (*). In this question, you'll explore whether some of these are superfluous. Recall that regular expressions R_1 and R_2 are called **equivalent** if the languages they describe are equal sets, that is if $L(R_1) = L(R_2)$.

For this question, the alphabet is $\{a, b\}$.

a. Can the regular expression $(a \cup b)^*$ be rewritten as an equivalent regular expression that does not include any union operations? If so, give that equivalent regular expression; if not, briefly justify why not.

b. Can the regular expression $(a \cup b)$ be rewritten as an equivalent regular expression that does not include any union operations? If so, give that equivalent regular expression; if not, briefly justify why not.

c. Can the regular expression $(a \cup b)^*$ be rewritten as an equivalent regular expression that does not include any Kleene star operations? If so, give that equivalent regular expression; if not, briefly justify why not.

d. Can the regular expression (ε^*) be rewritten as an equivalent regular expression that does not include any Kleene star operations? If so, give that equivalent regular expression; if not, briefly justify why not.

Extension (not for credit - do not hand in): Make a conjecture for a characterization of those languages that can be described by regular expressions that do not include any Kleene star operations. Similarly, make a conjecture for a characterization of those languages that can be described by regular expressions that do not include any union operations. Can you prove your conjectures?

2. (**10 points**) Consider the alphabet $\Sigma = \{0, 1\}$. We'll use the same sets from this week's individual homework, generalized to arbitrary positive integers $k \geq 2$.

$L_k = \{w \in \{0,1\}^* \mid \text{the length of } w \text{ is a multiple of } k\}$
$N_k = \{w \in \{1\}^* \mid \text{the length of } w \text{ is a multiple of } k\}$
$B_k = \{w \in \{0,1\}^* \mid \text{interpreting } w \text{ as a binary number (potentially with leading 0s), } w \text{ is a multiple of } k\}$

a. Design a DFA over Σ recognizing L_5. Draw the state diagram of your DFA in JFLAP, export the image as a png or jpg file, and include it as part of your submission. You do not need to justify your construction for credit, **but** if you describe how your state diagram works by briefly describing the role of each state and the transitions between them we may be able to award partial credit if your answer is incorrect.

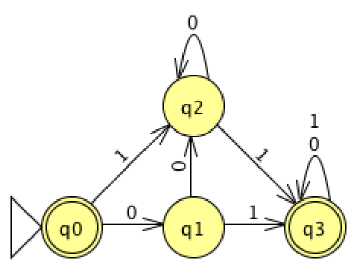
b. Design a DFA over Σ recognizing N_5. Draw the state diagram of your DFA in JFLAP, export the image as a png or jpg file, and include it as part of your submission. You do not need to justify your construction for credit, **but** if you describe how your state diagram works by briefly describing the role of each state and the transitions between them we may be able to award partial credit if your answer is incorrect.

c. Design a DFA over Σ recognizing B_5. Draw the state diagram of your DFA in JFLAP, export the image as a png or jpg file, and include it as part of your submission. You do not need to justify your construction for credit, **but** if you describe how your state diagram works by briefly describing the role of each state and the transitions between them we may be able to award partial credit if your answer is incorrect.

d. Generalize your work from part a. to define a DFA over Σ that recognizes L_k for arbitrary $k \geq 2$. Since k is not fixed, you **will not** be able to draw the state diagram. Instead, define this DFA formally by specifying each of the parameters $(Q, \Sigma, \delta, q_0, F)$.

Note: part c. of this problem has shown up in technical interviews for internships and full-time positions, using the terminology "Suppose you are working with an incoming bitstream (sequence of bits) that might be truncated (stopped) at any time. Find an efficient algorithm to determine if the number represented by this truncated sequence of bits is an integer multiple of 5*." It turns out that the DFA representation gives a very efficient algorithm!*

3. (**10 points**) Consider the DFA, M, whose state diagram is given by:



a. **True** or **False** Any string which has a nonempty prefix accepted by this DFA is also accepted by the DFA. Briefly justify your answer.

b. **True** or **False** There is a positive integer n such that all strings of length n are in $L(M)$. *Recall: "positive" means strictly positive,* $n > 0$. Briefly justify your answer.

c. Give a regular expression R so that $L(R)$ is an infinite subset of the language of this DFA. If $L(R) \subsetneq L(M)$, give an example string that is accepted by M but which is not in $L(R)$. If $L(R) = L(M)$, briefly justify the set equality. *Note: there are many correct answers for this question.*