Homework 1

COMS W3261, Summer A 2023

This homework is due **Monday, 5/29/2023, at 11:59pm EST**. Submit to GradeScope (course code: K3VK75). If you use late days, the absolute latest we can accept a submission is Friday at 11:59 PM EST.

Grading policy reminder: LaTeX is preferred, but neatly typed or handwritten solutions are acceptable.[1] Feel free to use the .tex file for the homework as a template to write up your answers, or use the template posted on the course website. Your TAs may dock points for indecipherable writing.

Proofs should be complete; that is, include enough information that a reader can clearly tell that the argument is rigorous.

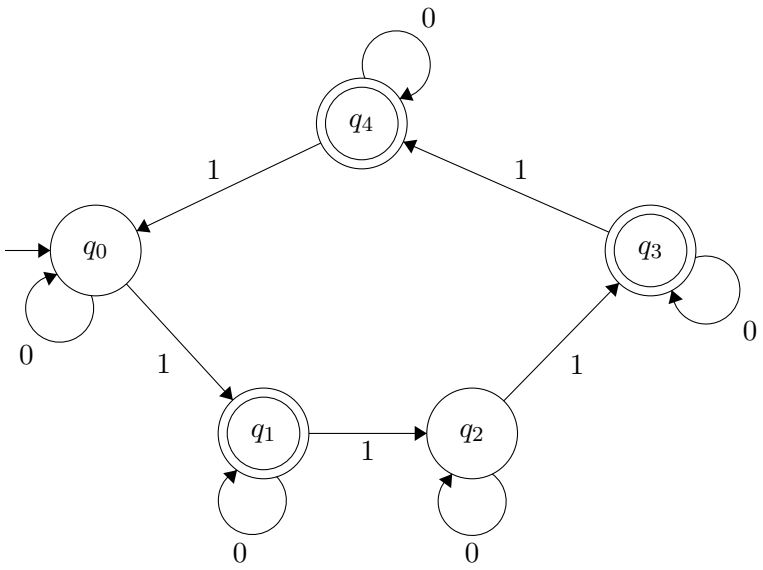
If a question is ambiguous, please state your assumptions. This way, we can give you credit for correct work. (Even better, post on Ed so that we can resolve the ambiguity.)

LaTeX resources.

- Detexify is a nice tool that lets you draw a symbol and returns the LaTeX codes for similar symbols.
- The tool Table Generator makes building tables in LaTeX much easier.
- The tool Finite State Machine Designer may be useful for drawing automata. See also this example (PDF) (.tex) of how to make fancy edges (courtesy of Eumin Hong).
- The website mathcha.io allows you to draw diagrams and convert them to LaTeX code.
- To use the previous drawing tools (and for most drawing in LaTeX), you'll need to use the package Tikz (add the command "\usepackage{tikz}" to the preamble of your .tex file to import the package).
- This tutorial is a helpful guide to positioning figures.

[1] The website Overleaf (essentially Google Docs for LaTeX) may make compiling and organizing your .tex files easier. Here's a quick tutorial.

Problem 1



1. (3 points.) The DFA state diagram above is defined on the alphabet $\Sigma = \{0, 1\}$. Write out its formal definition (as a 5-tuple). When specifying the transition function δ, you can draw a table or simply describe the output of δ for all states on all possible inputs.

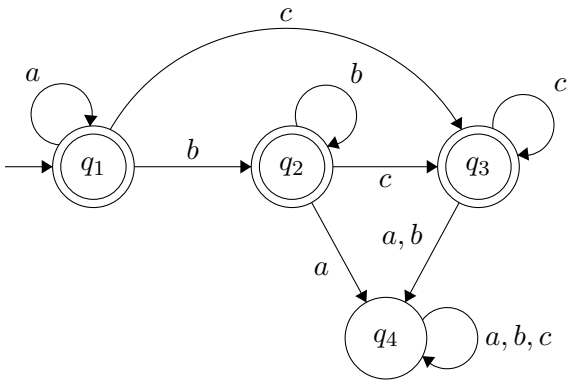
2. (3 points.) Describe the language recognized by the DFA in one or two sentences, then explain why the DFA accepts a string if and only if it matches your description. [Hint: Test some strings and look for a pattern; never forget the empty string ϵ!]

Rationale: The goal of this question is to make sure you're comfortable reading state diagrams, evaluating DFAs on strings, and reasoning about what they do.

References: Sipser 1.1 pp. 34-37; Lightning review of DFAs from the Resource Archive section of the course webpage.

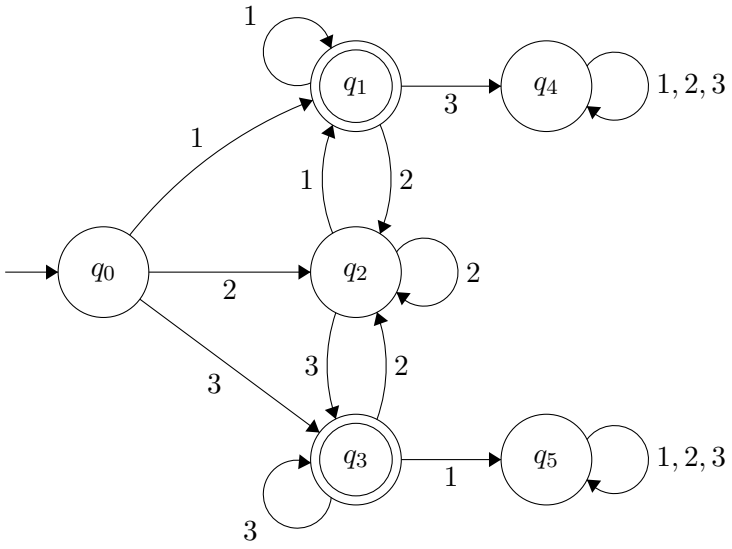
Problem 2.

1. (4 points.) Consider the DFA drawn below. What is the language recognized by this DFA (over what alphabet)? Describe your reasoning.



2. (4 points.) The DFA drawn below recognizes all strings over the alphabet $\{1, 2, 3\}$ that end in a 1 or a 3 and don't contain 1's and 3's adjacent to each other (that is, the substrings 13 and 31 are forbidden). Draw a DFA state diagram that recognizes the same language (that is, accepts exactly the same set of strings over $\{1, 2, 3\}$) but which has **at most four states** in total.

   Explain in words why your DFA recognizes the same language as the pictured DFA.



Rationale: More practice reading state diagrams and evaluating DFAs on strings; also, practice designing DFAs to perform a task and simplifying them.

References: Sipser 1.1 pp. 34-37 (DFAs); Sipser 1.1 pp. 41-43 (designing DFAs); Lightning reviews of DFAs and designing DFAs from the Resource Archive section of the course webpage.

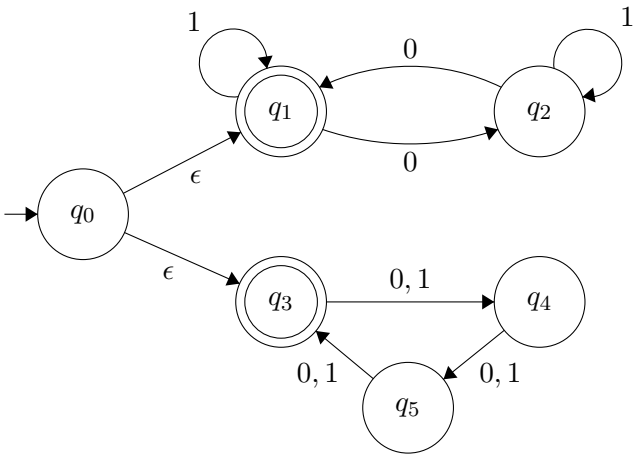
Problem 3

1. (4 points). In binary, the natural numbers are $0, 1, 10, 11, 100, 101, ...$, etc. Draw a state diagram for an NFA on the alphabet $\Sigma = \{0, 1\}$ that recognizes the language of binary natural numbers divisible by 4. (For instance, your machine should accept 100, but reject 11 and 101. Reject strings with extra leading zeroes, like 0100.)

   Explain in words why your NFA recognizes the language specified.

2. (2 points). Over the binary alphabet $\{0, 1\}$, Let L_1 be the language of all strings whose length is divisible by 3, and L_2 be the language of all strings that have an even number of 0's. The state diagram below depicts an NFA that recognizes $L_1 \cup L_2$.

   Specify the elements of a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ equivalent to the NFA state diagram below. When specifying the transition function, feel free to specify δ only for inputs that transition to a non-empty set of states; for the others, you can assume δ maps to $\emptyset$. No justification required.



3. (4 points). Draw the state diagram of a *DFA* over $\{0, 1\}$, **with at most 6 states**, that also recognizes $L_1 \cup L_2$. Explain in words why your DFA recognizes this language. (We'll discuss a general procedure for converting NFAs to DFAs in Lecture 3, but this question doesn't require it.)

Rationale: The purpose of this question is to get experience in building, testing, and simplifying NFAs, as well as converting NFAs to DFAs.

References: Sipser 1.2 pp. 47-53 (NFAs), also see the Lightning Review video on NFAs in the Resource Archive section of the course webpage.

Problem 4

1. (2 points.) Recall that the (Kleene) star operation is defined as follows:

$$A^* = \{x_1 x_2 \dots x_k \mid x_1, x_2, \dots, x_k \in A, k \geq 0\}.$$

   Let A and B be regular languages. Is it true that $(A \cup B)^*$ and $A^* \cup B^*$ are the same language? If so, provide a proof. If not, provide a counterexample. (Recall the order of operations: $A^* \cup B^* = (A^*) \cup (B^*)$.)

2. (2 points.) Let A be a regular language. Is it true that A^*A^* and $(AA)^*$ are the same language? If so, provide a proof. If not, provide a counterexample.

Some practice reasoning about the regular operations.
References: Sipser p.44-45 (definition of Union, Concatenation, and Star, as well as closure).

Problem 5

1. (6 points.) Given a language A over an alphabet Σ, define $doub(A)$ as follows:

$$doub(A) := \{w_1 w_1 w_2 w_2 \dots w_n w_n \mid w_1 w_2 \dots w_n \in A; w_1, w_2, \dots, w_n \in \Sigma\}.$$

   That is, each string in $doub(A)$ can be obtained by beginning with some string $w_1 w_2 \dots w_n$ in A and replacing each character with two copies of itself.

   Show that the class of regular languages is closed under the operation *doub* by explaining how to modify an arbitrary DFA D_1 so that it recognizes the language $doub(L(D_1))$. For simplicity, you may consider just the case in which the alphabet is $\Sigma = \{0, 1\}$.

2. (2 points.) Given a language A over an alphabet Σ, define $oct(A)$ similarly to $doub(A)$, except that we now replace each character with *eight* copies of itself.

   Show that the class of regular languages is closed under the operation *oct* by reasoning from the fact that the regular languages are closed under other, previously proved, regular expressions. (That is, prove *without* explaining how to modify a DFA.) You may assume that *doub* is regular.

Rationale: The point of this question is to practice proving regularity of operations using various techniques.

References: Sipser pp.45-46 for an example of constructing a DFA that recognizes $A_1 \cup A_2$, given DFAs that recognize A_1 and A_2; for part 2 Sipser p.44-45 (definition of Union, Concatenation, and Star, as well as closure).

Problem 6 (1 point)

1. Do you have any leftover questions or confusions from week 1 that you'd like to see addressed in class?

2. (Optional) Any other thoughts? Thank you!

Rationale: We'll spend more time reviewing the toughest topics from each week, and we'll repeat the activities that are most useful. Also, feedback helps make the course better.