

Homework 1

COMS W3261, Summer A 2022

This homework is due **Tuesday, 5/31/2022, at 11:59pm EST**. Submit to GradeScope (course code: 2KGDW8).

Grading policy reminder: \LaTeX is preferred, but neatly typed or handwritten solutions are acceptable.¹ I recommend using the .tex file for the homework as a template to write up your answers. 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.

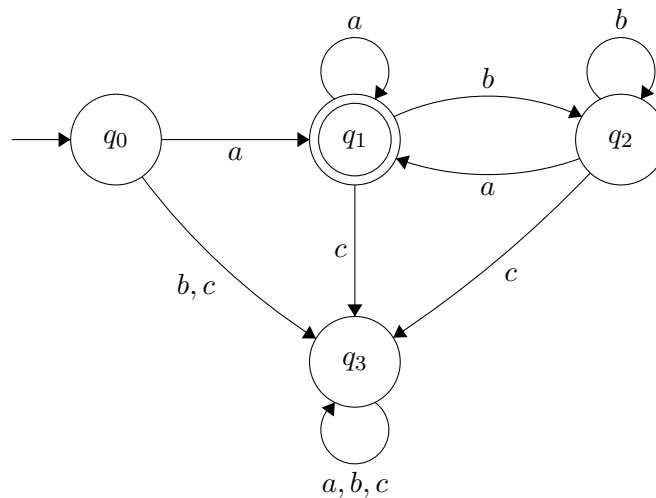
If a question is ambiguous, please state your assumptions. This way, we can give you credit for correct work. (Even better, post on Ed so that we can resolve the ambiguity.)

\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 making 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.

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

1 Problem 1 (6 points)



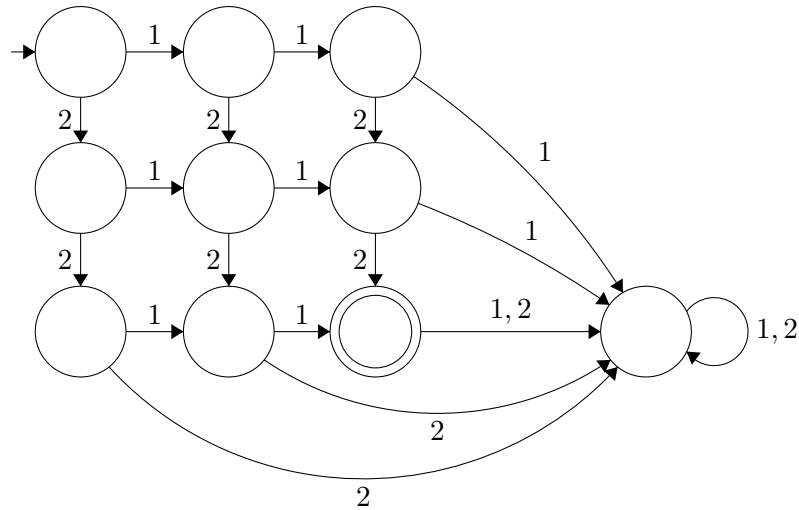
1. (3 points.) The DFA state diagram above is defined on the alphabet $\Sigma = \{a, b, c\}$. Write out its formal definition (as a 5-tuple). When specifying the transition function δ , you can draw a table or simply describe the output of δ on all possible inputs.
2. (3 points.) Describe the language recognized by the DFA in one sentence, 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 linked in the Lecture 1 [Course Skeleton](#).

2 Problem 2 (8 points).

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



- (4 points.) Draw a state diagram for a DFA **with at most 4 states** that recognizes the language over the alphabet $\Sigma = \{1, 2, 3\}$ consisting of all strings of digits that never decrease (that is, we never see a 1 after we've seen a 2 or a 3, or a 2 after we've seen a 3.)

Explain in words why your DFA recognizes the language specified.

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 linked in the Lecture 1 [Course Skeleton](#).

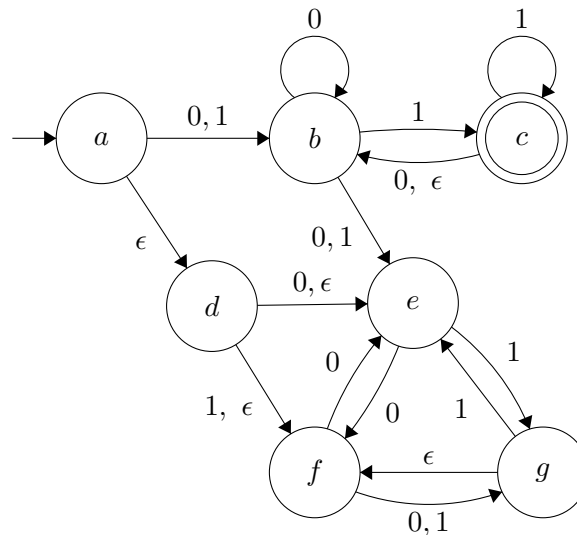
3 Problem 3 (8 points)

- (3 points). Draw a state diagram for an NFA on the alphabet $\Sigma = \{0\}$ that recognizes the language

$$L = \{w \mid |w| \text{ is divisible by 3 or by 5 (or by both 3 and 5)}. \}$$

(Recall that $|w|$ denotes the length of the string w .) Explain in words why your NFA recognizes the language specified.

- (5 points). Draw the state diagram of a DFA (with alphabet $\Sigma = \{0,1\}$) **with at most 3 states** that recognizes the same language as the NFA whose state diagram is pictured below. Explain in words why your DFA captures the same language as the original NFA. [Hint: A DFA with three states can only do so much. So, if this question is possible, the NFA below must be simpler than it seems. Try to work out what it does (testing strings is never a bad option.)]



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

References: Sipser 1.2 pp. 47-53 (NFAs), also see the Lightning Review video on NFAs linked in the [Course Skeleton](#).

4 Problem 4 (8 points)

1. (5 points.) Given languages A and B , define the *XNOR* operation \odot as follows:

$$A \odot B := \{x \mid x \in (A \cap B), \text{ or } x \in (\overline{A} \cap \overline{B})\}.$$

Prove that the class of regular languages is closed under \odot by showing that, given two arbitrary DFAs D_1 and D_2 , we can build a new DFA D_3 that recognizes the language $L(D_1) \odot L(D_2)$. [Hint: recall our proof that the regular languages are closed under the union operation, available on p.45 of Sipser.]

2. (3 points.) Prove that the class of regular languages is closed under \odot in a different way: by reasoning from the fact that the regular languages are closed under union, intersection, Kleene star, and complement (you may assume these facts without proof.)

Rationale: The goal of this question is to get comfortable building DFAs programmatically (combining two generic DFAs using their formal definitions), and also thinking about the logic of closure (if X and Y are regular and the regular languages are closed under \clubsuit , then $X \clubsuit Y$ is regular).

References: Sipser pp.45-46 (proof that the regular languages are closed under the union operation), [Wiki: Closure](#).

5 Problem 5 (9 points)

1. (9 points). Consider the language L on the alphabet $\{0, 1, 2\}$ consisting of all strings that meet both of the following conditions:
 - Every 0 is immediately followed by a 1, every 1 is immediately followed by a 2, and every 2 is immediately followed by an 0.
 - The string starts and ends with the same digit.

For example, the strings '1', '0120120', and '1201' are in the language, while '111', '210', and '012012' are not.

- (a) (6 points). Show that the following languages are regular:
 - i. L_{inc} , the language of strings over $\{0, 1, 2\}$ such that every 0 is immediately followed by a 1, every 1 is immediately followed by a 2, and every 2 is immediately followed by an 0.
 - ii. L_0 , L_1 , and L_2 , the languages of strings over $\{0, 1, 2\}$ that start and end in 0, 1, and 2 respectively.
- (b) (3 points). Use closure properties of the regular languages (closure under union, intersection, concatenation, etc), along with your results from the previous part to conclude that L is regular.

Rationale: The point of this question is to practice combining several types of regularity proofs including using DFAs, NFAs, and closure properties.

References: Sipser 1.2 pp. 47-53 (NFAs), the Lightning Review video on NFAs linked in the [Course Skeleton](#), [Wiki: Closure](#).

6 Problem 6 (1 point)

1. What was the muddiest (most confusing) point of class during week 1, in your opinion?
2. What was your favorite part of class during week 1?
3. (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.