CSE 105 - Theory of Computation (Winter 2026)
Homework 1: Describing languages and finite automata
Due Monday, January 12, 11:59pm

Instructions: There are two parts to this homework:

- Concept check (Question 1): Every student must complete this individually on Gradescope.
- Written Homework (Questions 2 to 5): You may work individually or in a team of up to 3 people. Please ensure your name(s) and PID(s) are clearly visible on the first page of your submission, and then upload the PDF to Gradescope. If working in a group, submit only one submission per group: one partner uploads the submission through their Gradescope account and then adds the other group member to the Gradescope submission by selecting their name in the "Add Group Members" dialog box. You will need to re-add your group member every time you resubmit a new version of your assignment.

It is highly recommended (though not required) that you type your answers. It is your responsibility to make any handwriting clear and legible for grading. A LaTeX template for the homework is provided on Canvas. For ease of grading, please start each new problem on a separate page.

We will only be grading some of the problems below for correctness. However, because all of the concepts are important, we will not reveal which problems are being graded for correctness until after the assignment has been submitted. The remaining problems will be graded for completeness (i.e., does it look like there was a good-faith effort to solve the problem?).

Reading and extra practice problems: Sipser Section 0, 1.1. Chapter 1 exercises 1.1, 1.2, 1.3.

Problems:

1. **Concept check**

   Complete the assignment "Homework 1 - Concept Check" on Gradescope. It consists of a few multiple choice questions that test some of the basic concepts we've learned in the class. A detailed explanation will appear if you've gotten the answer correct. The score you get on this part of the assignment *will* contribute to your homework grade, but you have unlimited attempts to get the right answer.

2. **Functions over sets of strings**

   Let Γ be an arbitrary alphabet, and let K be any language over Γ. Define

   $$\textsc{Substring}(K) := \{w \in \Gamma^* \mid \text{there exist } a, b \in \Gamma^* \text{ such that } awb \in K\}.$$

   That is, $\textsc{Substring}$ is a function that takes a language K as input and produces a new language as output. For the rest of this problem, assume $\Gamma = \{0, 1, 2\}$.

(a) Consider $w = 0112$ (which is a string in Γ^*). List every element of the set $\textsc{Substring}(\{w\})$. In other words, fill in the blank

$$\textsc{Substring}(\{w\}) = \{\underline{\hspace{6cm}}\}$$

Briefly justify your answer by referring back to the relevant definitions.

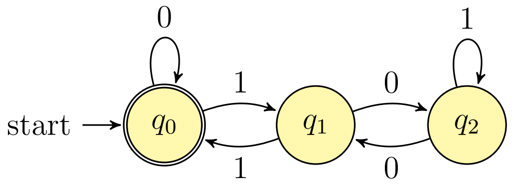
Not graded, but good to think about: Why do we need the curly braces—"{" and "}"—around w for the input to $\textsc{Substring}$*?*

(b) Specify an example language A over Γ such that $A \neq \Gamma^*$ and yet $\textsc{Substring}(A) = \Gamma^*$, or explain why there is no such example. A complete solution will include either (1) a precise and clear description of your example language A and a precise and clear description of the result of computing $\textsc{Substring}(A)$ using relevant definitions to justify this description and to justify the set equality with Γ^*, or (2) a sufficiently general and correct argument why there is no such example, referring back to the relevant definitions.

(c) Specify an example language A over Γ such that $\textsc{Substring}(A) \subset A$, or explain why there is no such example. Note, we are using the "$\subset$" symbol to mean *proper* subset. That is, there should be a string in A which is not in $\textsc{Substring}(A)$. Once again, a complete solution will include either (1) a precise and clear description of your example language A and a precise and clear description of the result of computing $\textsc{Substring}(A)$ using relevant definitions to justify this description and to justify the set equality with Γ^*, or (2) a sufficiently general and correct argument why there is no such example, referring back to the relevant definitions.

3. **The right transition function can make or break a DFA**

Consider the finite automaton $(Q, \Sigma, \delta, q_0, F)$ depicted below



where $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{0, 1\}$, and $F = \{q_0\}$.

(a) Find and fix the mistake in the following symbolic description of the transition function $\delta\colon Q \times \Sigma \to Q$: for each $j \in \{0, 1\}$

$$\delta(q_0, j) = q_j \qquad \delta(q_1, j) = q_{1-j} \qquad \delta(q_2, j) = q_{1+j}$$

(b) Keeping the same set of states Q, alphabet Σ, starting state q_0, and set of accepting states $\{q_0\}$, change the transition function δ so that the resulting finite automaton never accepts a string ending in a 1.

(Challenge question, not graded) There is a beautiful plain English description of the language recognized by the finite automaton with the state diagram depicted at the start of Question 3. What is it?

4. **Drawing DFA**

   For each of the languages below, draw the DFA that recognizes the language. You may assume that each language is over the alphabet $\Sigma = \{0, 1\}$.

   (a) $\{w \in \Sigma^* \mid w$ does not have even length and w does not contain the string $111\}$

   (b) $\{w \in \Sigma^* \mid$ if the number of 1's in w is odd, then the number of 0's in w is also odd$\}$

5. **Being precise with terminology**

   For each of the following statements, determine if it is true, false, or if the question doesn't even make sense (because the statement isn't well formed or doesn't use terms in ways consistent with definitions from class).

   (a) The empty string is in every language.

   (b) Σ^* is a language.

   (c) Every language is a finite automaton.

   (d) Alphabets are infinite.

   (e) There is a finite number $k \in \mathcal{N}$ such that every DFA has fewer than k states.