CSE 105 - Theory of Computation (Winter 2026)
Homework 6: Computational Problems, Recognizability, Decidability
Due Wednesday, February 25, 11:59pm

Instructions: There are two parts to this homework:

- Concept check (Question 1): Every student must complete this individually on Gradescope.
- Written Homework (Question 2-4): 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: Chapter 4 exercises 4.1, 4.3, 4.4., 4.5. Chapter 4 Problems 4.29, 4.30.

Problems:

1. **Concept check**

   Complete the assignment "Homework 6 - Concept Check" on Gradescope.

2. **Explicit encodings**

   In a computational problem, the elements of the language are encodings of machines. For example, consider the language

   $$E_{\text{DFA}} := \{\langle M\rangle \mid M \text{ is a DFA, and } L(M) = \varnothing\}$$

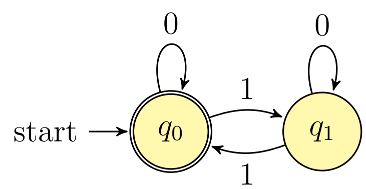
   where each string $\langle M\rangle$ in the language encodes a DFA $M = (Q, \Sigma, \delta, q_0, F)$. Usually, we purposefully drop the details about how this encoding is done because they can distract from the central computational properties of the language. In fact, any encoding can be used so long as there exists a decider for syntactic questions about the DFAs being encoded. In this question, we will build some specific explicit examples of encodings of DFAs to get more comfortable with these ideas.

(a) *Encoding with delimiters:* Perhaps the most straightforward way to create an encoding is to have it mirror the structure of the tuple $(Q, \Sigma, \delta, q_0, F)$ for the DFA. Your task: describe an encoding that maps each DFA M to a distinct string $\langle M \rangle$ that uniquely identifies M. That is, if you "decode" the encoding, you get the exact same machine back.

Hints, tips, notes of caution:

- You may use special characters like # and \$ as delimiters in your encoding to separate the various components.
- Your encoding alphabet must be finite

(b) Use your encoding from part (a) to produce the string encoding the DFA below:



(c) Show that it is possible to have the same kind of delimited encoding without using special delimiter characters. In particular, prove that for every DFA M, we can assume that $\langle M \rangle \subseteq \{0, 1\}^*$.

Challenge; not graded:
For the delimited encoding schemes above, there are strings over the encoding alphabet (Σ) that nevertheless do not correspond to a valid DFA. Prove/disprove: There exists an encoding scheme for which this is not true; that is,

$$\{\langle M \rangle \mid M \textit{ is a DFA}\} = \Sigma^*.$$

3. **Closure**

Recall the function

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

(a) Prove that, given any nondeterministic Turing machine over Γ, N_L, there is a nondeterministic Turing machine over Γ that recognizes

$$\textsc{Substring}(\ L(N_L)\ )$$

In other words, you will prove that the class of Turing-recognizable languages over Γ is closed under the SUBSTRING operation. A complete answer will include both a precise construction of the machine and a (brief) justification of why this machine works as required.

(b) Give a different proof that the class of Turing-recognizable languages over Γ is closed under the SUBSTRING operation, this time using only deterministic Turing machines. A complete answer will include both a precise construction of the machine and a (brief) justification of why this machine works as required.

4. **Computational problems**

For each of the following statements, determine if it is true or false. Clearly label your choice by starting your solution with **True** or **False** and then provide a brief (3-4 sentences or so) justification for your answer.

(a) For each regular language K, the language

$$\{\langle M \rangle \mid M \text{ is a DFA and } L(M) = K\}$$

is decidable.

(b) For each regular language L, the language

$$\{\langle M_1, M_2 \rangle \mid M_1, M_2 \text{ are both DFA and } L(M_1) \subseteq L \text{ and } L(M_2) \subseteq \overline{L}\}$$

is decidable.

(c) Let $\text{Model} \in \{\text{DFA}, \text{NFA}, \text{REX}, \text{CFG}, \text{PDA}\}$. If EQ_{Model} is decidable, then E_{Model} is decidable.

Challenge; not graded: Let $\text{Model} \in \{\text{DFA}, \text{NFA}, \text{REX}, \text{CFG}, \text{PDA}\}$. *If* A_{Model} *is decidable, then* EQ_{Model} *is decidable.*