# HW6CSE105W25: Homework assignment 6

CSE105W25

Due: March 13, 2025 at 5pm, via Gradescope

**In this assignment,**

You will practice analyzing, designing, and working with reductions to compare the difficulty level of computational problems. You will explore various ways to encode machines as strings so that computational problems can be recognized.

**Resources**: To review the topics for this assignment, see the class material from Weeks 8 and 9. We will post frequently asked questions and our answers to them in a pinned Piazza post.

**Reading and extra practice problems**: Sipser Sections 4.2, 5.3, 5.1. Chapter 4 exercises 4.9, 4.12. Chapter 5 exercises 5.4, 5.5, 5.6, 5.7. Chapter 5 problems 5.22, 5.23, 5.24, 5.28

**For all HW assignments:** Weekly homework may be done individually or in groups of up to 3 students. You may switch HW partners for different HW assignments. Please ensure your name(s) and PID(s) are clearly visible on the first page of your homework 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(s) to the Gradescope submission by selecting their name(s) in the "Add Group Members" dialog box. You will need to re-add your group member(s) every time you resubmit a new version of your assignment. Each homework question will be graded either for correctness (including clear and precise explanations and justifications of all answers) or fair effort completeness. On the "graded for correctness" questions, you may only collaborate with CSE 105 students in your group; if your group has questions about a problem, you may ask in drop-in help hours or post a private post (visible only to the Instructors) on Piazza. On the "graded for completeness" questions, you may collaborate with all other CSE 105 students this quarter, and you may make public posts about these questions on Piazza.

All submitted homework for this class must be typed. You can use a word processing editor if you like (Microsoft Word, Open Office, Notepad, Vim, Google Docs, etc.) but you might find it useful to take this opportunity to learn LaTeX. LaTeX is a markup language used widely in computer science and mathematics. The homework assignments are typed using LaTeX and you can use the source files as templates for typesetting your solutions. To generate state diagrams of

machines, you can (1) use the LaTex tikzpicture environment (see templates in the class notes), or (2) use the software tools Flap.js or JFLAP described in the class syllabus (and include a screenshot in your PDF), or (3) you can carefully and clearly hand-draw the diagram and take a picture and include it in your PDF. 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.

**Integrity reminders**

- Problems should be solved together, not divided up between the partners. The homework is designed to give you practice with the main concepts and techniques of the course, while getting to know and learn from your classmates.
- On the "graded for correctness" questions, you may only collaborate with CSE 105 students in your group. You may ask questions about the homework in office hours (of the instructor, TAs, and/or tutors) and on Piazza (as private notes viewable only to the Instructors). You *cannot* use any online resources about the course content other than the class material from this quarter – this is primarily to ensure that we all use consistent notation and definitions (aligned with the textbook) and also to protect the learning experience you will have when the 'aha' moments of solving the problem authentically happen.
- Do not share written solutions or partial solutions for homework with other students in the class who are not in your group. Doing so would dilute their learning experience and detract from their success in the class.

You will submit this assignment via Gradescope (https://www.gradescope.com) in the assignment called "hw6CSE105W25".

**Assigned questions**

1. **What's wrong with these reductions? (if anything)** (16 points): Suppose your friends are practicing coming up with mapping reductions $A \leq_m B$ and their witnessing functions $f : \Sigma^* \to \Sigma^*$. For each of the following attempts, determine if it has error(s) or is correct. Do so by labelling each attempt with all and only the labels below that apply, and justifying this labelling.

- *Error Type 1:* The given function can't witness the claimed mapping reduction because there exists an $x \in A$ such that $f(x) \notin B$.
- *Error Type 2:* The given function can't witness the claimed mapping reduction because there exists an $x \notin A$ such that $f(x) \in B$.
- *Error Type 3:* The given function can't witness the claimed mapping reduction because the specified function is not computable.
- *Correct:* The claimed mapping reduction is true and is witnessed by the given function.

Clearly present your answer by providing a brief (3-4 sentences or so) justification for whether **each** of these labels applies to each example.

(a) (*Graded for completeness*) [1] $A_{\text{TM}} \leq_m HALT_{\text{TM}}$ and

$$f(x) = \begin{cases} \langle \text{start} \to (q_{\text{acc}}), \varepsilon \rangle & \text{if } x = \langle M, w \rangle \text{ for a Turing machine } M \text{ and string } w \\ & \quad \text{and } w \in L(M) \\ \langle \text{start} \xrightarrow{0,1,\sqcup \to R} (q_0) \quad (q_{acc}) \rangle & \text{otherwise} \end{cases}$$

(b) (*Graded for completeness*) $A_{\text{TM}} \leq_m EQ_{\text{TM}}$ with

$$f(x) = \begin{cases} \langle \text{start} \to (q_{\text{acc}}), M_w \rangle & \text{if } x = \langle M, w \rangle \text{ for a Turing machine } M \text{ and string } w \\ \langle \text{start} \to (q_{\text{acc}}), \ \text{start} \to (q_{\text{rej}}) \quad (q_{\text{acc}}) \rangle & \text{otherwise.} \end{cases}$$

Where for each Turing machine $M$, we define

$M_w =$ "On input $y$
1. Simulate $M$ on $w$.
2. If it accepts, accept.
3. If it rejects, reject."

(c) (*Graded for correctness*) [2] $HALT_{\text{TM}} \leq_m EQ_{\text{TM}}$ with

$$f(x) = \begin{cases} \langle \text{start} \to (q_{\text{acc}}), M_w \rangle & \text{if } x = \langle M, w \rangle \text{ for a Turing machine } M \text{ and string } w \\ \langle \text{start} \to (q_{\text{acc}}), \ \text{start} \to (q_{\text{rej}}) \quad (q_{\text{acc}}) \rangle & \text{otherwise.} \end{cases}$$

[1] This means you will get full credit so long as your submission demonstrates honest effort to answer the question. You will not be penalized for incorrect answers. To demonstrate your honest effort in answering the question, we expect you to include your attempt to answer *each* part of the question. If you get stuck with your attempt, you can still demonstrate your effort by explaining where you got stuck and what you did to try to get unstuck.

[2] This means your solution will be evaluated not only on the correctness of your answers, but on your ability to present your ideas clearly and logically. You should 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.

Where for each Turing machine $M$, we define

$M_w =$ "On input $y$
1. If $y$ is not the empty string, accept.
2. Else, simulate $M$ on $w$.
3. If it accepts, accept.
4. If it rejects, reject."

(d) (*Graded for correctness*) $\{ww \mid w \in \{0,1\}^*\} \leq \Sigma^*$ and $f(x) = 11$ for each $x \in \{0,1\}^*$.

(e) (*Graded for correctness*) $\Sigma^* \leq_m \{ww \mid w \in \{0,1\}^*\}$ and $f(x) = 11$ for each $x \in \{0,1\}^*$.

2. **Using mapping reductions** (14 points): Consider the following computational problems we've discussed

$$A_{TM} = \{\langle M, w\rangle \mid M \text{ is a Turing machine, } w \text{ is a string and } M \text{ accepts } w\}$$
$$HALT_{TM} = \{\langle M, w\rangle \mid M \text{ is a Turing machine, } w \text{ is a string and } M \text{ halts on } w\}$$
$$E_{TM} = \{\langle M\rangle \mid M \text{ is a Turing machine and } L(M) = \emptyset\}$$
$$EQ_{TM} = \{\langle M_1, M_2\rangle \mid M_1, M_2 \text{ are both Turing machines and } L(M_1) = L(M_2)\}$$

and the new computational problem

$IncludesEmptyString_{TM} = \{\langle M\rangle \mid M$ is a Turing machine and
$M$ accepts the empty string (and maybe other strings too)$\}$

(a) (*Graded for correctness*) Give an example of a string that is an element of $IncludesEmptyString_{TM}$ and a string that is not an element of $IncludesEmptyString_{TM}$ and briefly justify your choices.

(b) (*Graded for completeness*) Prove that $IncludesEmptyString_{TM}$ is not decidable by showing that $A_{TM} \leq_m IncludesEmptyString_{TM}$.

(c) (*Graded for correctness*) Give a different proof that $IncludesEmptyString_{TM}$ is not decidable by showing that $HALT_{TM} \leq_m IncludesEmptyString_{TM}$.

(d) (*Graded for completeness*) Is $IncludesEmptyString_{TM}$ recognizable? Justify your answer.

3. **Using mapping reductions** (14 points): Consider the following computational problems we've discussed

$$A_{TM} = \{\langle M, w\rangle \mid M \text{ is a Turing machine, } w \text{ is a string and } M \text{ accepts } w\}$$
$$HALT_{TM} = \{\langle M, w\rangle \mid M \text{ is a Turing machine, } w \text{ is a string and } M \text{ halts on } w\}$$
$$E_{TM} = \{\langle M\rangle \mid M \text{ is a Turing machine and } L(M) = \emptyset\}$$
$$EQ_{TM} = \{\langle M_1, M_2\rangle \mid M_1, M_2 \text{ are both Turing machines and } L(M_1) = L(M_2)\}$$

and the new computational problem

$NotIncludesEmptyString_{TM} = \{\langle M\rangle \mid M$ is a Turing machine and $M$ does not accept the empty string$\}$

(a) (*Graded for correctness*) Prove that $NotIncludesEmptyString_{TM}$ is not the complement of $E_{TM}$ and is also not the complement of $IncludesEmptyString_{TM}$.

(b) (*Graded for completeness*) Prove that $NotIncludesEmptyString_{TM}$ is not decidable by showing that $\overline{HALT_{TM}} \leq_m NotIncludesEmptyString_{TM}$.

(c) (*Graded for correctness*) Give a different proof that $NotIncludesEmptyString_{TM}$ is not decidable by showing that $\overline{A_{TM}} \leq_m NotIncludesEmptyString_{TM}$.

(d) (*Graded for completeness*) Is $NotIncludesEmptyString_{TM}$ recognizable? Justify your answer.

4. **Examples of languages** (6 points):

For each part of the question, use precise mathematical notation or English to define your examples and then briefly justify why they work.

For each language $L$ over an alphabet $\Sigma$, we have the associated sets of strings (also over $\Sigma$)

$$L^* = \{w_1 \cdots w_k \mid k \geq 0 \text{ and each } w_i \in L\}$$

and

$$SUBSTRING(L) = \{w \in \Sigma^* \mid \text{there exist } x, y \in \Sigma^* \text{ such that } xwy \in L\}$$

and

$$EXTEND(L) = \{w \in \Sigma^* \mid w = uv \text{ for some strings } u \in L \text{ and } v \in \Sigma^*\}$$

(a) (*Graded for correctness*) Two undecidable languages $L_1$ and $L_2$ over the same alphabet whose union $L_1 \cup L_2$ is co-recognizable, or write **NONE** if there is no such example (and explain why).

(b) (*Graded for correctness*) An unrecognizable language $L_3$ for which $EXTEND(L_3)$ is regular or write **NONE** if there is no such example (and explain why).

(c) (*Graded for completeness*) A co-recognizable language $L_4$ that is NP-complete, or write **NONE** if there is no such example (and explain why). Recall the definition: A language $L$ over an alphabet $\Sigma$ is called **co-recognizable** if its complement, defined as $\Sigma^* \setminus L = \{x \in \Sigma^* \mid x \notin L\}$, is Turing-recognizable.

*This part of the question uses definitions from Week 10 of the course.*