HW6

Due: October 30, 2025

Reminder: Submit your assignment on Gradescope by the due date. Submissions must be typeset. Each page should include work for only one problem (i.e., make a new page/new pages for each problem). See the course syllabus for the late policy.

While collaboration is encouraged, please remember not to take away notes from any labs or collaboration sessions. Your work should be your own. Use of other third-party resources is strictly forbidden.

Please monitor Ed discussion, as we will post clarifications of questions there.

Problem 1

If A $\leq_m$ B and B is a regular language, does that imply that A is a regular language? Prove why or why not?

Solution: If A $\leq_M$ B, and B is a regular language, it doesn't imply that A is a is a regular language.

We will provide a counterexample. Let $A = \{0^n 1^n\}$ and let $B = \{1\}$. We have that A is mapping reducible to B if and only if there exists a computable function $f : \Sigma^* \to \Sigma^*$ such that for every $x \in \Sigma^*$, we have $x \in A$ if and only if $f(x) \in B$. Let

$$f(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{otherwise.} \end{cases}$$

Note that A is a context-free language (we have seen this in class). We also know that context-free languages are decidable. Thus, there exists a decider M that decides A, and the function f can simply use this decider to output 1 if M accepts the input string and 0 otherwise. Thus, f is computable so $A \leq_m B$. But, B is clearly regular since it is finite, and A is a CFL, so we have disproven the statement.

Problem 2

Consider the following languages. For each language, determine whether you can use Rice's Theorem to prove it is undecidable. If so, use Rice's

Theorem to prove it is undecidable. If not, explain why you cannot use Rice's Theorem, and prove it is undecidable without using Rice's Theorem.

a. $L_{inf} = \{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is not finite}\}$

b. $L_{101} = \{\langle M \rangle \mid M \text{ is a TM and } 101 \in L(M)\}$

c. $L_{seq} = \{\langle M_1 \rangle, \langle M_2 \rangle \mid M_1, M_2 \text{ are TMs and } L(M_1) \subseteq L(M_2)\}$

Solution:

a. In this case, you can use Rice's Theorem to prove that L_{inf} is undecidable.

   Rice's theorem states that if a language L consisting of descriptions of Turing Machines satisfies the following three conditions, it is undecidable:

   (a) $\langle M \rangle \in L$ and $L(M) = L(N) \Rightarrow \langle N \rangle \in L$.
   (b) $\exists \langle M \rangle \in L$.
   (c) $\exists \langle M \rangle$ not in L.

   L_{inf} satisfies (1): Suppose $\langle M \rangle \in L$ and $L(M) = L(N)$. Since $L(M)$ consists of the TM encodings that recognize infinite languages, we also have that $L(N)$ consists of TM encodings that recognize infinite languages, which implies that $\langle N \rangle \in L$.

   L_{inf} satisfies (2): Let M be the TM that accepts immediately on all inputs. Then, $L(M)$ is not finite, and $\langle M \rangle$ is in L_{inf}.

   L_{inf} satisfies (3): Let M be the TM that rejects immediately on all inputs. Then, $\mid L(M) \mid = 0$, and $\langle M \rangle$ is not in L_{inf}.

   Therefore, by Rice's Theorem, L_{inf} must be undecidable.

b. In this case, you can use Rice's Theorem to prove that L_{101} is undecidable.

   Rice's theorem states that if a language L consisting of descriptions of Turing Machines satisfies the following three conditions, it is undecidable:

   (a) $\langle M \rangle \in L$ and $L(M) = L(N) \Rightarrow \langle N \rangle \in L$.
   (b) $\exists \langle M \rangle \in L$.
   (c) $\exists \langle M \rangle$ not in L.

L_{101} satisfies (1): Suppose $\langle M \rangle \in L$ and $L(M) = L(N)$. Since $L(M)$ contains 101, it must be that $L(N)$ contains 101.

L_{101} satisfies (2): Let M be the TM that accepts immediately on all inputs. Then $101 \in L(M)$ and $\langle M \rangle$ is in L_{101}.

L_{101} satisfies (3): Let M be the TM that rejects immediately on all inputs. Then $101 \notin L(M)$ and $\langle M \rangle$ is not in L_{101}.

Therefore, by Rice's Theorem, L_{101} must be undecidable.

c. We cannot use Rice's theorem to conclude that L_{SEQ} is undecidable because we are examining a language that is composed of tuples of TMs. The property is not solely about the language of a single TM; it is dependent on the language of another TM and their relation with one another.

To prove that L_{SEQ} is undecidable, we will do a proof by contradiction. Assume that L_{SEQ} is decidable, then there exists a TM R that decides L_{SEQ}. We construct a TM S that decides $EQ_{TM} = \{\langle M_1 \rangle, \langle M_2 \rangle \mid M_1 \text{ and } M_2 \text{ are TMs and } L(M_1) = L(M_2)\}$ using R as follows:

$S =$ "On input $\langle M_1 \rangle, \langle M_2 \rangle$, where M_1 and M_2 are TMs:

(a) Ensure that M_1 and M_2 are TMs.
(b) Run R on input $\langle M_1 \rangle, \langle M_2 \rangle$. If R accepts, proceed; if R rejects, reject.
(c) Run R on input $\langle M_2 \rangle, \langle M_1 \rangle$. If R accepts, accept; if R rejects, reject."

Since R decides L_{SEQ}, S decides EQ_{TM}. But EQ_{TM} is undecidable as shown in class, so it must be that L_{SEQ} is undecidable, completing the proof.

For the correctness of S, consider $\langle M \rangle, \langle M' \rangle \in EQ_{TM}$, which tells us that M and M' are TMs and $L(M) = L(M')$. Since $L(M) = L(M')$, we have that $L(M) \subseteq L(M')$ and $L(M') \subseteq L(M)$. Therefore, R will accept on inputs $\langle M \rangle, \langle M' \rangle$ and $\langle M' \rangle, \langle M \rangle$, meaning S accepts $\langle M \rangle, \langle M' \rangle$. Now, consider $\langle M \rangle, \langle M' \rangle \notin EQ_{TM}$. Either M is not a TM, M' is not a TM, $L(M) \neq L(M')$, or a combination. If M or M' is not a TM, R will reject immediately. If $L(M) \neq L(M')$, S will reject on either step 2 or step 3. In all cases, S rejects $\langle M \rangle, \langle M' \rangle$.

For the decidability of S, we run R twice, which is decidable from our assumption. Therefore, we have that S is decidable.

M on w.) Then run D on $\langle M', 1\rangle$, and return the result.

Problem 3

a. For any natural number k, consider the language $A_{TM}^{(k)}$:

$$A_{TM}^{(k)} = \{\langle M, w\rangle \mid M \text{ is a TM with at most } k \text{ states and } M \text{ accepts } w\}$$

Prove that there is some natural number k where $A_{TM}^{(j)}$ is undecidable for all $j \geq k$.

b. A *very sparse language* is a language L over an alphabet Σ that contains exactly one string of each length. If L is a very sparse language, it contains the empty string, exactly one string of length one, exactly one string of length two, and so on. Prove that if L is a very sparse language and is Turing-recognizable, then L is decidable.

Solution:

a. To prove this, we can reduce from A_{TM}^C, which we know to be unrecognizable and thus also undecidable. Given an input $\langle M, w\rangle$, a TM that would recognize A_{TM}^C would need to recognize the cases when M rejects w or loops infinitely. Suppose M has n states. Then we already know that a TM for $A_{TM}^{(i)}$ when $i < n$ will not accept no matter what, because the TM has more than i states. If there exists some TM for $A_{TM}^{(i)}$ that rejects $\langle M, w\rangle$ where $i \geq n$, then it must be because M does not accept w. Now suppose for the sake of contradiction that for all natural numbers k, there exists some TM T_j that decides $A_{TM}^{(j)}$ for $j \geq k$.

   Let S be a Turing machine that on input $\langle M, w\rangle$. It simulates M on w, accepts if M accepts, and rejects if M rejects. Let n be the number of states in S. By supposition, we can take some T_j that solves $A_{TM}^{(j)}$ for $j > n$. We can then build a decider for A_{TM}^C as follows:

   On input of $\langle M, w\rangle$, our decider runs T_j on $\langle S, \langle M, w\rangle\rangle$. If T_j rejects, then our decider accepts; otherwise, it rejects.

This decider will accept iff T_j rejects $\langle S, \langle M, w\rangle\rangle$. That is, our decider will accept iff S does not accept $\langle M, w\rangle$; or, finally, our decider will accept iff M does not accept w.

So we have clearly constructed a decider for A_{TM}^C, which we know to be unrecognizable; so our assumption must be false, and there must be some k such that for all $j \geq k$, $A_{TM}^{(j)}$ is undecidable.

b. Let L be a recognizable, very sparse language. Suppose R is a recognizer of L, and we would like to be able to decide L. We can construct such a decider D as follows:

On input x, measure the length n of x. Over a finite alphabet Σ, there are a finite number of strings with length n. So, D runs R on all of these strings simultaneously. Since we know that there is exactly one string in L of each length, eventually one of these strings must be recognized; and because we are running each simultaneously (by doing them one step at a time), even if some loop forever it will not block finding the string in L.

Thus, D's simulations of R will accept exactly once on one string of length n, which must be the corresponding string in L. If the accepted string is x, D accepts because x must be in L. Otherwise D rejects, because the only string in L with the same length as x must not be x. Hence, D is a decider for L.