Homework 10 Solutions

1. If $A \leq_{\mathrm{m}} B$ and B is a regular language, does that imply that A is a regular language?

   Answer: No. For example, define the languages $A = \{ \mathtt{0}^n \mathtt{1}^n \mid n \geq 0 \}$ and $B = \{\mathtt{1}\}$, both over the alphabet $\Sigma = \{\mathtt{0}, \mathtt{1}\}$. Define the function $f : \Sigma^* \to \Sigma^*$ as

   $$f(w) = \begin{cases} \mathtt{1} & \text{if } w \in A, \\ \mathtt{0} & \text{if } w \notin A. \end{cases}$$

   Observe that A is a context-free language (slide 2-5), so it is also Turing-decidable (Theorem 4.9). Thus, f is a computable function. Also, $w \in A$ if and only if $f(w) = \mathtt{1}$, which is true if and only if $f(w) \in B$. Hence, $A \leq_{\mathrm{m}} B$. Language A is nonregular (slide 1-105), but B is regular since it is finite (slide 1-95).

2. Show that A_{TM} is not mapping reducible to E_{TM}. In other words, show that no computable function reduces A_{TM} to E_{TM}. (Hint: Use a proof by contradiction, and facts you already know about A_{TM} and E_{TM}.)

   Answer: Suppose for a contradiction that $A_{\mathrm{TM}} \leq_{\mathrm{m}} E_{\mathrm{TM}}$ via reduction f. This means that $w \in A_{\mathrm{TM}}$ if and only if $f(w) \in E_{\mathrm{TM}}$, which is equivalent to saying $w \notin A_{\mathrm{TM}}$ if and only if $f(w) \notin E_{\mathrm{TM}}$. Therefore, using the same reduction function f, we have that $\overline{A_{\mathrm{TM}}} \leq_{\mathrm{m}} \overline{E_{\mathrm{TM}}}$. However, $\overline{E_{\mathrm{TM}}}$ is Turing-recognizable (HW 8, problem 4) and $\overline{A_{\mathrm{TM}}}$ is not Turing-recognizable (Corollary 4.23), contradicting Theorem 5.22.

3. Consider the language

   $$A\varepsilon_{\mathrm{TM}} = \{ \langle M \rangle \mid M \text{ is a TM that accepts } \varepsilon \}.$$

   Show that $A\varepsilon_{\mathrm{TM}}$ is undecidable.

   Answer: We will show that A_{TM} reduces to $A\varepsilon_{\mathrm{TM}}$. Suppose for contradiction that $A\varepsilon_{\mathrm{TM}}$ is decidable, and let R be a TM that decides $A\varepsilon_{\mathrm{TM}}$. We construct another TM S with input $\langle M, w \rangle$ that does the following. It first uses M and w to construct a new TM M_2, which takes input x. If $x \neq \varepsilon$, then M_2 accepts; otherwise, M_2 runs M on input w and M_2 accepts if M accepts w. Note that M_2 recognizes the language $\Sigma^* - \{\varepsilon\}$ if M does not accept w; otherwise, M_2 recognizes the language Σ^*. In other words, M_2 accepts ε if and only if M accepts w. So our TM S decides A_{TM}, which is a contradiction since we know A_{TM} is undecidable.

Here are the details of our TM S:

S = "On input $\langle M, w\rangle$, where M is a TM and w is a string:

0. Check if $\langle M, w\rangle$ is a valid encoding of a TM M and string w. If not, *reject.*

1. Construct the following TM M_2 from M and w:

M_2 = "On input x:

1. If $x \neq \varepsilon$, *accept.*

2. If $x = \varepsilon$, then run M on input w and *accept* if M accepts w."

2. Run R on input $\langle M_2\rangle$.

3. If R accepts, *accept*; if R rejects, *reject.*"

4. A *useless state* in a Turing machine is one that is never entered on any input string. Consider the problem of determining whether a state in a Turing machine is useless. Formulate this problem as a language and show it is undecidable.

Answer: We define the decision problem with the language

$$\mathit{USELESS}_{\text{TM}} = \{\langle M, q\rangle \mid q \text{ is a useless state in TM } M\}.$$

We show that $\mathit{USELESS}_{\text{TM}}$ is undecidable by reducing E_{TM} to $\mathit{USELESS}_{\text{TM}}$, where $E_{\text{TM}} = \{\langle M\rangle \mid M \text{ is a TM and } L(M) = \emptyset\}$. We know E_{TM} is undecidable by Theorem 5.2.

Suppose that $\mathit{USELESS}_{\text{TM}}$ is decidable and that TM R decides it. Note that for any Turing machine M with accept state q_{accept}, q_{accept} is useless if and only if $L(M) = \emptyset$. Thus, because TM R solves $\mathit{USELESS}_{\text{TM}}$, we can use R to check if q_{accept} is a useless state to decide E_{TM}. Specifically, below is a TM S that decides E_{TM} by using the decider R for $\mathit{USELESS}_{\text{TM}}$ as a subroutine:

S = "On input $\langle M\rangle$, where M is a TM:

1. Run TM R on input $\langle M, q_{\text{accept}}\rangle$, where q_{accept} is the accept state of M.

2. If R accepts, *accept.* If R rejects, *reject.*"

However, because we known E_{TM} is undecidable, there cannot exist a TM that decides $\mathit{USELESS}_{\text{TM}}$.