CPSC 351 — Tutorial Exercise #8
Additional Practice Problems

About These Problems

These problems will not be discussed during the tutorial, and solutions for these problems will not be made available. They can be used as "practice" problems that can help you practice skills considered in the lecture presentations for Lecture #8 and Lecture #9, or in Tutorial Exercise #8.

Practice Problems

1. Let $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$ and let $L \subseteq \Sigma^\star$. Show that each of the following languages is ***oracle-reducible*** to L.

   (a) The language $L_{\text{end}} \subseteq \Sigma^\star$ that includes all strings in $\Sigma^\star$ the *end* with a string in L — that is, the language

   $$L_{\text{end}} = \{\mu \cdot \nu \mid \mu \in \Sigma^\star \text{ and } \nu \in L\}.$$

   (b) The language $L_{\text{xorA}} \subseteq \Sigma^\star$ consisting of all strings $\omega \in \Sigma^\star$ which satisfy ***exactly one*** of the following conditions:

   - ω begins with "a".
   - $\omega \in L$.

   Thus, if $\mu \in \Sigma^\star$ then the string $\mathtt{a} \cdot \mu$ is in L_{xorA} if and only if $\mathtt{a} \cdot \mu \notin L$. On the other hand, the string $\mathtt{b} \cdot \mu$ is in L_{xorA} if and only if $\mathtt{b} \cdot \mu \in L$.

2. Let $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, and let $L_1, L_2 \subseteq \Sigma^\star$ such that

   $$L_2 = \{\omega \cdot \mathtt{a} \mid \omega \in L_1\}.$$

   Suppose that L_1 is undecidable. Give a ***many-one reduction*** to prove that L_2 is also undecidable.

In the preparatory material for Lecture #9, a language "All_{TM}" was proved to be undecidable. This was the language of Yes-instances of the following decision problem:

The "Accepting All" Problem
Instance: A Turing machine $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$.
Question: Does M accept every input string? That is, is $L(M) = \Sigma^\star$?

So one could also say that the "Accepting All" Problem was proved to be undecidable.

Consider the following the following decision problem:

The "Same Language" Problem
Instance: A pair of of Turing machines, M_1 and M_2, with the same input alphabet.
Question: Does M_1 and M_2 have the same language? This is, is $L(M_1) = L(M_2)$?

3. To begin, describe a high-level mapping that can (probably) be used to show that if the "Accepting All" problem is undecidable then the "Same Language" problem is also undecidable too.

In order to continue, it is necessary to consider encodings of instances of the "Same Language" problem. Let $\Sigma_{\text{2TM}} = \Sigma_{\text{TM}} \cup \{\#\}$.

- A *pair* of Turing machines M_1 and M_2 can be encoded as a string $\alpha\#\beta \in \Sigma_{\text{2TM}}^\star$ where $\alpha \in \text{TM} \subseteq \Sigma_{\text{TM}}^\star$ is the encoding for M_1 and $\beta \in \text{TM} \subseteq \Sigma_{\text{TM}}^\star$ is the encoding for M_2.
- Let $\text{Pair}_{\text{TM}} \subseteq \Sigma_{\text{2TM}}^\star$ be the language of encodings of pairs of Turing machines

  $$M_1 = (Q_1, \Sigma, \Gamma_1, \delta_1, q_{0,1}, q_{A,1}, q_{R,1})$$

  and

  $$M_2 = (Q_2, \Sigma, \Gamma_2, \delta_2, q_{0,2}, q_{A,2}, q_{R,2})$$

  with the same input alphabet $\boldsymbol{\Sigma}$**.**
- Now let

  $$\text{E}_{\text{TM}} \subseteq \text{Pair}_{\text{TM}} \subseteq \Sigma_{\text{2TM}}^\star$$

  be the language including encodings of pairs of Turing machines M_1 and M_2, with the same input alphabet Σ, such that $L(M_1) = L(M_2)$.

4. Sketch a proof that the language Pair_{TM} is decidable.
5. Prove that $\text{All}_{\text{TM}} \preceq_{\text{M}} \text{E}_{\text{TM}}$ — so that the language E_{TM} is undecidable.