CS46 Homework 8

This homework is due at 10pm on Sunday, April 12.

For this homework, you will work with a partner or alone. It's ok to discuss approaches at a high level with other students, but most of your discussions should just be with your partner. Your partnership's write-up is your own: do not share it, and do not read other teams' write-ups. If you use any out-of-class references (anything except class notes, the textbook, or asking Lila), then you **must** cite these in your post-homework survey. Please refer to the course webpage or ask me any questions you have about this policy.

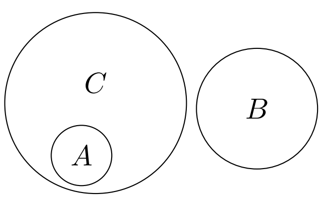
1. **Classifying languages.** Consider the following languages.
   (a) $ALL_{\text{TM}} = \{\langle M \rangle \mid L(M) = \Sigma^*\}$
   (b) $ODD_{\text{TM}} = \{\langle M \rangle \mid L(M) \text{ contains no strings of even length}\}$

   For each, is the language decidable? Turing-recognizable? co-Turing-recognizable?

   Provided an argument for your answers. (Give the deciders/recognizers that you claim exist, and show why they work; if they do not exist, then prove why not.) You may consider the questions in any order, if proving one helps you with another.

2. A **useless state** in a Turing machine is one that is never entered on any input string. Consider the problem of determining whether a Turing machine has any useless states.
   (a) Formulate this problem as a language.
   (b) Show that this language is undecidable.

3. Two disjoint languages A and B are **decidably separable**[1] if there is a decidable language C such that $A \subseteq C$ and $C \cap B = \emptyset$.



   In the above diagram,
   $A \cap B = \emptyset$
   $A \subseteq C$
   $C \cap B = \emptyset$

   (a) Give two example languages which are decidably separable.
   (b) Give two example disjoint languages which are *not* decidably separable.
   (c) If A and B are disjoint languages which are both co-Turing-recognizable, show that A is decidably separable from B.
   (d) **(extra credit)** Given a Turing-recognizable language A, let machine(A) $= \{\langle M \rangle \mid L(M) = A\}$. Show that if A and B are Turing-recognizable languages and $A \subsetneq B$, then machine(A) is not decidably separable from machine(B).

[1]This is an interesting property in the following situation: imagine two undecidable, disjoint languages A and B. If some language C decidable separates them, then C can help give — decidable! — hints for strings of A and B, e.g. every string $w \in C$ is definitely $\notin B$. Since B is undecidable, we might otherwise not have tools to figure out which strings are not in B, so this is helpful even if it doesn't give perfect answers.

4. **Computable functions. (extra credit)** Recall that a function $f : \Sigma^* \to \Sigma^*$ is **computable** if some Turing machine M, on every w, halts with just $f(w)$ on its tape.

   (a) Let $f : \Sigma^* \to \Sigma^*$ be a partial computable function which is one-to-one and onto. Prove that f^{-1} is a total computable function.

   (b) Show that if functions f and g are computable, then their composition $f \circ g$ is computable.