CS46 lab 9

This homework is due at 11:59pm on Sunday, March 27. This is a 6-point homework.

For this homework, you will work with a partner. It's ok to discuss approaches at a high level with other students, but your detailed discussions should be only with your partner. The only exception to this rule is work you've done with another student *while in lab*. In this case, note who you've worked with and what parts were solved during lab. Your partnership's write-up and code 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/asking Lila), then you **must** cite these in your post-homework survey. Please refer to the course webpage or directly ask any questions you have about this policy.

The main **learning goal** of this homework is to work with and think about Turing machines and decidability. You should feel free as always to cite and use techniques and theorems from class or the textbook.

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

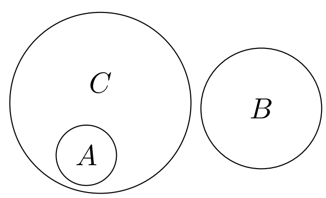

Figure 1: In this diagram, $A \cap B = \emptyset$, $A \subseteq C$, and $C \cap B = \emptyset$.

   (a) Give two example languages which are decidably separable. (And explain why they are.)
   (b) Give two example disjoint languages which are *not* decidably separable. (And explain.)
   (c) If A and B are disjoint languages which are both co-Turing-recognizable, show that A is decidably separable from B.
   (d) **(extra challenge)** Given a Turing-recognizable language A, define:

$$\text{machine}(A) = \{\langle M \rangle \mid L(M) = A\}$$

   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.