Intro to Undecidability

**Question 1.** Is $A_{TM} = \{\langle M, w \rangle : M \text{ is a Turing Machine and } M \text{ accepts } w \}$ a Turing-recognizable language?

**Question 2.** How many Turing Machines exist? I don’t mean how many were built in the physical world, but rather how many possible Turing Machines are there?

**Question 3.** How large is the set of all infinite binary sequences?

**Question 4.** How large is the set of all languages? For context of what I am asking, recall that the set of strings accepted by a given automaton (including Turing Machines) is a *language*.

**Question 5.** Is $A_{TM} = \{\langle M, w \rangle : M \text{ is a Turing Machine and } M \text{ accepts } w \}$ a Turing-decidable language?
We say a language is **co-Turing-recognizable** if it is the complement of a Turing-recognizable language.

**Question 6.** What can we say if a language is both Turing-recognizable and also co-Turing-recognizable? Prove that this is the case.

**Question 7.** We saw that $A_{TM} = \{\langle M, w \rangle : M \text{ is a Turing Machine and } M \text{ accepts } w \}$ a Turing-recognizable language. What about $\overline{A_{TM}}$?

**The Halting Problem**

**Question 8.** The *Halting Problem* is as follows. The language $HALT_{TM} = \{\langle M, w \rangle : M \text{ is a Turing Machine and } M \text{ halts on input } w \}$. Show that $HALT_{TM}$ is undecidable. Why is this easier than the undecidable proofs we did last time?
Question 9. Show that $E_{TM} = \{ \langle M \rangle : M \text{ is a Turing Machine and } L(M) = \emptyset \}$ is undecidable.
Suppose FSOC that $E_{TM}$ is decidable and I want to decide $A_{TM}$ given input $\langle M, w \rangle$.

Question 10. Show that $EQ_{TM} = \{ \langle M_1, M_2 \rangle : M_1 \text{ and } M_2 \text{ are Turing Machines and } L(M_1) = L(M_2) \}$ is undecidable.

*Hint*: is there any language one of those machines could recognize where the equality of it and a mystery machine would be useful information for you?

**General Strategy** to prove $X$ is undecidable

- Suppose I had a TM that decides $X$
- Pick an undecidable problem $Y$
- Write a TM to decide $Y$
  - Must be a valid TM **EXCEPT**
    it assumes existence of TM to decide $X$
- But that would decide $Y$
- By contradiction, no TM for $X$