1. Recall that we defined (in class and in a homework problem) the languages

\[ X_{TM} = \{ \langle M \rangle : M \text{ does not accept } \langle M \rangle \} \]

\[ NE_{TM} = \{ \langle M \rangle : L(M) \text{ is not empty} \} \]

We proved that \( X_{TM} \) is not decidable, and not even recognizable, and that \( NE_{TM} \) is recognizable.

(a) Prove that \( NE_{TM} \) is undecidable, by showing that \( X_{TM} \leq_T NE_{TM} \). Specifically, let \( S \) be a decider for \( NE_{TM} \), show a decider \( T \) for \( X_{TM} \).

(b) Why can’t the same proof be used to prove that \( NE_{TM} \) is not recognizable? Specifically, if \( S \) is a recognizer for \( NE_{TM} \), why wouldn’t \( T \) (that you built in part (a)) be a recognizer for \( X_{TM} \)?

2. Show that no Java program could be written which, given as input an arbitrary Java program, can determine whether or not there is any input on which the given program prints out “AN ERROR HAS OCCURED”. (You may assume Java programs are equivalent to Turing Machines).

3. Define the language

\[ SMALL = \{ \langle M \rangle : L(M) = \{0000, 0101, 1110\} \} \]

Prove that SMALL is undecidable by showing \( A_{TM} \leq_T SMALL \).

4. (Extra Credit:) For a class of languages \( C \), we define \( C_{TM} = \{ \langle M \rangle : L(M) \in C \} \).

(a) Let \( C \) be a class of languages such that \( \emptyset \in C \), and such that there exists a TM-recognizable language \( L_N \notin C \).
Prove that \( C_{TM} \) is undecidable, by showing that \( A_{TM} \leq_T C_{TM} \).
Note: in class (and Theorem 5.3 in text) we proved this for the special case of \( C \) being the class of regular languages.

(b) Let \( C \) be a class of languages, and \( L_N, L_Y \) be two TM-recognizable languages, such that \( L_Y \in C \), and \( L_N \notin C \). Prove that \( C_{TM} \) is undecidable, by showing that \( A_{TM} \leq_T C_{TM} \).
Note: Obviously, the previous part (4a) is a special case of this one. Another special case is the previous problem (3).