Assignment 11: (Un)decidability
Due: 1:15pm, Tuesday, November 23

1. Read Chapter 4 (pages 165–182) of Sipser. Come up with (at least) two questions about the reading where you're not sure of the answer. These may relate to points where the book is confusing, or simply to some related question or conjecture that occurs to you while doing the reading.

2. What is the smallest alphabet $\Sigma$ such that all Turing machines can be encoded as members of $\Sigma^*$? Justify your answer.

3. Consider the language of Turing machine descriptions

   $$L = \{ \langle M \rangle \mid M \text{ accepts at least 999 strings} \}.$$

   Is $L$ decidable, recognizable, or neither? Prove your answer.

4. Consider the language of Turing machine descriptions

   $$L = \{ \langle M, w \rangle \mid M \text{ accepts } w \text{ using at most 999 tape cells} \}.$$

   Show that $L$ is decidable.
5. For each of the following languages \( L \), state whether \( L \) is decidable, recognizable, or neither. Carefully justify each answer.

(a) \( \{ a \} \)
(b) \( \{ \langle M \rangle \mid L(M) = \{ a \} \} \)
(c) \( \{ \langle M \rangle \mid M \text{ is the only TM accepting } L(M) \} \)
(d) \( \{ \langle M \rangle \mid M \text{ does not accept any string ending in } 0 \} \)
(e) \( \{ \langle M \rangle \mid L(M) \text{ is regular} \} \)
(f) \( \{ \langle M \rangle \mid M \text{ halts on } \epsilon \text{ in no more than } 1000 \text{ steps} \} \)

6. Goldbach's Conjecture states that every even integer greater than 2 can be expressed as the sum of two primes. It has not yet been proved, but (according to Wikipedia) has been verified for all even numbers up to \( 1.609 \times 10^{18} \). Explain how, given a (sadly, mythical) Halt checker, we could determine the truth of Goldbach's Conjecture once and for all.

7. **The Elusiveness of Undecidability.** It is widely known that Halting (membership in \( H = \{ \langle M, w \rangle \mid M \text{ halts on input } w \} \)) is not decidable. But:

(a) Consider the following machine, which we will call \( TM_1 \). It first confirms that the tape contains a number expressed in binary (and if not, rejects). It then enters the following loop:

- All digits are zero, it halts and accepts (ending the loop).
- Otherwise, it decrements the binary number by 1 (by completely overwriting it.)

Does the machine \( TM_1 \) halt on the input \( 10000000000000 \)?

(b) Consider a similar machine, which we will call \( TM_2 \). It is exactly the same as \( TM_1 \), except that each time around the loop it *increments* the binary number by 1. Does the machine \( TM_2 \) halt on the input \( 10000000000000 \)?

(c) Why don’t your answers to Parts (a) and (b) contradict the undecidability of Halting?

(d) \( TM_1 \) and \( TM_2 \) are not very interesting. You are probably wondering, since Halting is undecidable, where the hard Turing Machines are.

Suppose I were to design a very complicated \( TM_3 \), and prove mathematically that there is no method to determine whether \( TM_3 \) terminates on the input \( 10000000000000 \) (i.e., that you will never know). Show that this supposition leads to a contradiction.