Exercise 1: Semi-Decidable vs. Recursively Enumerable  \(5\) Points

Very often people in computer science use the terms \textit{semi-decidable} and \textit{recursively enumerable} equivalently. The following exercise shows in which way they actually are equivalent. We first recall the definition of both terms.

A language \(L\) is \textit{semi-decidable} if there is a Turing machine which accepts every \(w \in L\) and does not accept any \(w \notin L\) (this means the TM can either reject \(w \notin L\) or simply not stop for \(w \notin L\)).

A language is \textit{recursively enumerable} if there is a Turing machine which eventually outputs every word \(w \in L\) and never outputs a word \(w \notin L\).

(a) Show that any recursively enumerable language is semi-decidable.

(b) Show that any semi-decidable language is recursively enumerable.

Exercise 2: Halting Problem \(3+2+2+2\) Points

The special halting problem is defined as

\[ H_s = \{ \langle M \rangle \mid \langle M \rangle \text{ encodes a TM and } M \text{ halts on } \langle M \rangle \} \]

(a) Show that \(H_s\) is undecidable.

\textit{Hint: Assume that }M\text{ is a TM which decides }H_s\text{ and then construct a TM which halts iff }M\text{ does reject. Use this construction to find a contradiction.}

(b) Show that the special halting problem is recursively enumerable.

(c) Show that the complement of the special halting problem is not recursively enumerable.

\textit{Hint: What can you say about a language }L\text{ if }L\text{ and its complement are recursively enumerable? (if you make some observation for this, also prove it)}

(d) Let \(L_1\) and \(L_2\) be recursively enumerable languages. Is \(L_1 \setminus L_2\) recursively enumerable as well?

(e) Is \(L = \{ w \in H_s \mid |w| \leq 1742 \}\) decidable? Explain your answer!

Exercise 3: Undecidability \(5\) Points

Fix an enumeration of all Turing machines (that have input alphabet \(\Sigma\)): \(\langle M_1 \rangle, \langle M_2 \rangle, \langle M_3 \rangle, \ldots\).

Fix also an enumeration of all words over \(\Sigma\): \(w_1, w_2, w_3, \ldots\).

Prove that language \(L = \{ w \in \Sigma^* \mid w = w_i, \text{ for some } i, \text{ and } M_i \text{ does not accept } w_1 \}\) is not Turing recognizable.

\textit{Hint: Try to find a contradiction to the existence of a Turing machine that recognizes }L.