Exercise 1: Decidability? (3 Points)

Let $n$ be a fixed positive integer, $\Sigma = \{0, 1\}$ a fixed alphabet, $M$ a fixed TM and $w \in \Sigma^*$ a fixed word.

$L_{\Sigma,M,n,w} := \begin{cases} \{1^n\}, & M \text{ stops on } w \text{ in at most } n \text{ steps} \\ \{0^n\}, & M \text{ stops on } w \text{ after } > n \text{ steps} \\ \emptyset, & M \text{ does not stop on } w. \end{cases}$

Is $L_{\Sigma,M,n,w}$ decidable?

**Remark:** For some $a \in \Sigma$, $a^n$ denotes the word which repeats $a$ $n$ times.

Exercise 2: Semi-Decidable vs. Recursively Enumerable (4 Points)

Very often people in computer science use both terms equivalently. The following exercise shows in which way they actually are equivalent. We first recall the definition of both terms.

A language $L$ is **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 **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 3: Halting Problem (3+2+2+1 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.

*Hint: Assume that $M$ is a TM which decides $H_s$ and then construct a TM which halts iff $M$ does not halt. 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.

*Hint: What can you say about a language $L$ if $L$ 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?
Exercise 4 (2+3 points)

(a) Show that every finite language is a decidable.

(b) Assume that $\pi$ is a fixed order of the words in $\Sigma^*$ for which a Turing machine $P$ exists that decides in finite time whether $\pi(w) \leq \pi(w')$ for all $w, w' \in \Sigma^*$. Furthermore, assume that for a given language $L$ there is a Turing machine $M$ that enumerates all words of a given language $L$ in order $w_1, w_2, w_3, \ldots$ such that $\pi(w_i) \leq \pi(w_j)$ holds for all $i \leq j$. Show that $L$ is decidable.