Exercise 1: Designing a Turing Machine

Design a Turing machine which accepts the language \( L = \{ w \# \text{rev}(w) \mid w \in \{0, 1\}^* \} \) where \( \text{rev} \) denotes the reverse complement, i.e., \( \text{rev}(a_1a_2 \ldots a_n) = \overline{a_n} \overline{a_{n-1}} \ldots \overline{a_2} \overline{a_1} \) with \( 0 = 1 \) and \( 1 = 0 \).

Remark: It is sufficient to give a detailed description of the Turing Machine. You do not need to give a formal definition.

Exercise 2: Semi-Decidable vs. Recursively Enumerable

Very often people in computer science use the terms semi-decidable and 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 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

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?

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