Exercise sheet 1

Exercise 1.1. Describe in detail a Turing machine that decides the language

$$L = \{w\#w \mid w \in \{0,1\}^*\}$$

Note that we're allowing # as an input symbol to simplify the task.

Exercise 1.2. *(This exercise assumes some familiarity with deterministic finite automata.)* Give a high-level description of a Turing machine for each of the following decision problems.

(a) $A_{DFA} = \{\langle A, w\rangle \mid A \text{ is a DFA that accepts } w\}$

(b) $E_{DFA} = \{\langle A\rangle \mid A \text{ is a DFA and } L(A) = \varnothing\}$

(c) $EQ_{DFA} = \{\langle A, B\rangle \mid A \text{ and } B \text{ are DFAs and } L(A) = L(B)\}$

Exercise 1.3. Let $L(M)$ be the language *recognized* by the TM M. That is, the set of inputs $w \in \{0,1\}^*$ for which M halts with output 1 (on other inputs it may halt with output 0 or not halt at all). Show that the following problem is undecidable : $E_{TM} = \{\langle M\rangle \mid M \text{ is a TM and } L(M) = \varnothing\}$.

Idea: Show that if E_{TM} were decidable, then the halting problem would be decidable.

Exercise 1.4. A *2-tape* TM is a TM that has 2 tapes, each with an independent tape head. At the beginning of the computation, the input resides on the first tape and the second tape is empty. A transition depends on the symbols under both heads, writes symbols to both tapes, and potentially moves both tape heads (possibly in different directions). If the TM halts, its output is what resides on the first tape starting from the first tape head.

(a) Show that for any 2-tape TM M that computes a function $f\colon \{0,1\}^* \to \{0,1\}$, there exists a 1-tape TM M' that computes f.

(b) Show that for any 2-tape TM M that computes a function $f\colon \{0,1\}^* \to \{0,1\}$ in time $t(n)$, there exists a 1-tape TM M' that computes f in time $O(t(n)^2)$.

Exercise 1.5. A TM is called *oblivious* if its head movements only depend on the *length* of the input. Show that for any TM M that computes a function $f\colon \{0,1\}^* \to \{0,1\}$ there exists a TM M' such that

- M' is oblivious,
- M' computes the same function as M, and
- if M computes f in time $t(n)$ (where $t(n) \geq n$), then M' computes f in time $O(t(n)^2)$.