MAST31213 Complexity theory
Master's Programme in Mathematics and Statistics
Fall 2019
Exercise set 3

Read chapters 1.1–1.3 of the book.

Exercise 1. Given a (1-tape) Turing machine M with halting states q_{halt}, q_{accept} and q_{reject}, describe a (1-tape) Turing machine M' with only one halting state q_{halt} that simulates M such for each calculation on some input

- M' halts if and only if M halts,
- if M ends in state q_{accept} then M' halts with a 1 as its first and only non-blank symbol on the tape,
- if M ends in state q_{reject} then M' halts with a 0 as its first and only non-blank symbol on the tape,
- if M ends in state q_{halt} then M' halts with neither a 0 or a 1 as its first symbol on the tape.

If M halts in $F(n)$ steps for an input of length n, how many steps does M' need for the simulating calculation?

Exercise 2. Assume L is a language in some alphabet , and M is a (1-tape) Turing machine that *decides* L (i.e., calculates the characteristic function of L) in time $T(n)$. Show that there are (1-tape) Turing machines M' and M'' such that

- M' decides $^* - L$ (i.e., the complement of L)
- M'' decides $\{\sigma \in^*: \sigma^R \in L\}$, where σ^R is σ reversed.

In what times do your M' and M'' run?

Exercise 3. Define a *two-dimensional* Turing machine to be a Turing machine where each of its tapes is an infinite grid, and the machine can move not only left and right, but also up and down. Show that for every (time-constructible) $T : \mathbb{N} \to \mathbb{N}$ and every Boolean function f, if f can be computed in time $T(n)$ using a two-dimensional Turing machine, then f can be computed in time $c \cdot T(n)^2$, for some positive constant c, on a standard Turing machine.

Exercise 4. Define a Turing machine M to be *oblivious* if its head movements do not depend on the input but only on the input length, i.e. for every input $x \in^*$ and $i \in \mathbb{N}$, the location of each of M's heads at the ith step is only a function of $|x|$ and i. Show that for every time-constructible $T : \mathbb{N} \to \mathbb{N}$, if a language $L \subseteq^*$ is decidable (on a k-tape Turing Machine) in time $O(T(n))$, then there is an oblivious Turing machine that decides L in time $O(T(n)^2)$. Furthermore, the oblivious machine may be chosen to have only one input tape and one work/output tape.