Computational Complexity

Handout – Lecture 1

Definition 1 (Turing machines)**.** A *Turing machine* $\mathbb{M}$ is a tuple (Γ, Q, δ), where:

- Γ is the *alphabet*: a finite set of symbols, including 0, 1, $\square$ (the blank symbol), and $\triangleright$ (the start symbol);
- Q is a finite set of *states*, including a designated start state q_{start} and a designated halting state q_{halt}; and
- $\delta : Q \times \Gamma^k \to Q \times \Gamma^{k-1} \times \{\text{L}, \text{R}, \text{S}\}^k$ is a *transition function*, for some $k \geq 2$ (the number of tapes of the machine).

The machine operates on $k \geq 2$ *tapes*, that each consist of a countably infinite number of *tape cells*: cell 1 (leftmost), cell 2, etc. For each of these tapes, the machine has a tape head that is positioned at one of the cells of that tape (and that can move to different cells).

When we execute the machine $\mathbb{M}$ on some input string $\sigma_1\sigma_2 \dots \sigma_n$, we initialize the machine and tapes as follows. (This is called the *start configuration* of $\mathbb{M}$ on input $\sigma_1\sigma_2 \dots \sigma_n$.) The input tape (the first tape) contains the start symbol $\triangleright$ on its first cell, the symbols $\sigma_1, \sigma_2, \dots, \sigma_n$ on the next n cells, and a blank symbol $\square$ on each of the remaining cells. The other tapes contain the start symbol $\triangleright$ on their first cell, and a blank symbol $\square$ on each of the remaining cells. For each tape, the tape head of the machine is positioned at the first cell of the tape. The machine $\mathbb{M}$ is in the starting state q_{start}.

The machine operates in discrete time steps. At each step, if the machine is in state q and $\tau_1, \dots, \tau_k$ are the symbols that are in the cells that are currently being read (the cells where the k tape heads are located), then $\delta(q, \tau_1, \dots, \tau_k) = (q', \tau_2', \dots, \tau_k', d_1, \dots, d_k)$ describes the configuration of the machine in the next step:

- the machine will be in state q',
- the symbols $\tau_2, \dots, \tau_k$ under the tape heads of tapes $2, \dots, k$ will be replaced by $\tau_2', \dots, \tau_k'$ (the input tape is read-only), and
- for each $1 \leq i \leq k$, the tape head on the i-th tape will move one cell to the left if $d_i = \text{L}$, will move one cell to the right if $d_i = \text{R}$, and will stay at the same position if $d_i = \text{S}$. (If the tape head is at the leftmost cell and it should move to the left, it stays in place.)

We iteratively apply the transition function in this way. Whenever the machine reaches the halting state q_{halt}, we stop (then we say that the machine has *halted*). We then consider the contents of the k-th tape (the output tape) as *output* of the computation.

The machine $\mathbb{M}$ computes the following (partial) function f, where for each $x \in \Sigma^*$, $f(x) = y$ if $\mathbb{M}$ halts on input x with output y, and $f(x) = \text{undefined}$ if $\mathbb{M}$ does not halt on input x.

Definition 2 (Decision problems)**.** A *decision problem* is a function $f : \Sigma^* \to \{0, 1\}$ where for each input $x \in \Sigma^*$ the correct output $f(x)$ is either 0 or 1.

Alternatively, one can view a decision problem as a formal language $L \subseteq \Sigma^*$, where for each $x \in \Sigma^*$ it holds that $x \in L$ if and only if $f(x) = 1$.

Definition 3. Let $f, g : \mathbb{N} \to \mathbb{R}$ be functions.

$$f \in O(g) \quad \text{means} \quad (\exists c, n_0 \in \mathbb{N})(\forall n \geq n_0)(|f(n)| \leq c|g(n)|) \tag{$\leq$}$$

$$f \in o(g) \quad \text{means} \quad (\forall \varepsilon > 0)(\exists n_0 \in \mathbb{N})(\forall n \geq n_0)(|f(n)| < \varepsilon|g(n)|) \tag{$<$}$$

$$\text{or equivalently} \quad \lim_{n\to\infty} \frac{|f(n)|}{|g(n)|} = 0$$

$$f \in \Omega(g) \quad \text{means} \quad g \in O(f) \tag{$\geq$}$$

$$f \in \omega(g) \quad \text{means} \quad g \in o(f) \tag{$>$}$$

$$f \in \Theta(g) \quad \text{means} \quad f \in O(g) \cap \Omega(g) \tag{$=$}$$

Definition 4 (Polynomial-time computability)**.** A function $f : \Sigma^* \to \Sigma^*$ is *polynomial-time computable* (or *computable in polynomial time*) if there exist a Turing machine $\mathbb{M}$ and a constant $c \in \mathbb{N}$ such that $\mathbb{M}$ computes f and for each $x \in \Sigma^*$ it takes $O(|x|^c)$ steps.