Computational Complexity

Handout – Lecture 7

Definition 1 (NP)**.** A language $L \subseteq \Sigma^*$ is in the complexity class NP if there is a polynomial $q : \mathbb{N} \to \mathbb{N}$ and a polynomial-time Turing machine $\mathbb{M}$ such that for every $x \in \Sigma^*$:

$$x \in L \quad \text{if and only if} \quad \text{there exists some } u \in \{0,1\}^{q(|x|)} \text{ such that } \mathbb{M}(x,u) = 1.$$

Definition 2 (coNP)**.** A language $L \subseteq \Sigma^*$ is in the complexity class coNP if there is a polynomial $q : \mathbb{N} \to \mathbb{N}$ and a polynomial-time Turing machine $\mathbb{M}$ such that for every $x \in \Sigma^*$:

$$x \in L \quad \text{if and only if} \quad \text{for all } u \in \{0,1\}^{q(|x|)} \text{ it holds that } \mathbb{M}(x,u) = 1.$$

Definition 3 (Σ_2^{p})**.** A language $L \subseteq \Sigma^*$ is in the complexity class Σ_2^{p} if there is a polynomial $q : \mathbb{N} \to \mathbb{N}$ and a polynomial-time Turing machine $\mathbb{M}$ such that for every $x \in \Sigma^*$:

$$\begin{array}{lll} x \in L & \text{if and only if} & \text{there exists } u_1 \in \{0,1\}^{q(|x|)} \text{ such that} \\ & & \text{for all } u_2 \in \{0,1\}^{q(|x|)} \text{ it holds that } \mathbb{M}(x,u_1,u_2) = 1. \end{array}$$

Definition 4 (Σ_i^{p})**.** Let $i \geq 1$. A language $L \subseteq \Sigma^*$ is in the complexity class Σ_i^{p} if there is a polynomial $q : \mathbb{N} \to \mathbb{N}$ and a polynomial-time Turing machine $\mathbb{M}$ such that for every $x \in \Sigma^*$:

$$\begin{array}{llll} x \in L & \text{if and only if} & \text{there exists } u_1 \in \{0,1\}^{q(|x|)} \text{ such that} & \\ & & \text{for all } u_2 \in \{0,1\}^{q(|x|)} & \\ & & \vdots & \\ & & \text{for all } u_i \in \{0,1\}^{q(|x|)} & \\ & & \text{it holds that } \mathbb{M}(x,u_1,\ldots,u_i) = 1. & \text{if } i \text{ is even,} \\ & & & \\ x \in L & \text{if and only if} & \text{there exists } u_1 \in \{0,1\}^{q(|x|)} \text{ such that} & \\ & & \text{for all } u_2 \in \{0,1\}^{q(|x|)} & \\ & & \vdots & \\ & & \text{there exists } u_i \in \{0,1\}^{q(|x|)} & \\ & & \text{such that } \mathbb{M}(x,u_1,\ldots,u_i) = 1. & \text{if } i \text{ is odd.} \end{array}$$

Definition 5 (Π_i^{p})**.** Let $i \geq 1$. The complexity class Π_i^{p} contains all languages that are the complement of a language in Σ_i^{p}:

$$\Pi_i^{\mathrm{p}} = \{ \ \overline{L} \mid L \in \Sigma_i^{\mathrm{p}} \ \}.$$

Definition 6. *Alternating Turing machines* are variants of (deterministic) Turing machines, where a few elements are modified.

- Instead of one halting state q_{halt}, there are two halting states q_{acc} (the *accept state*) and q_{rej} (the *reject state*).
- Instead of a single transition function δ, there are two transition functions δ_1, δ_2.
- The set $Q \setminus \{q_{\text{acc}}, q_{\text{rej}}\}$ is partitioned into $Q_\exists$ and $Q_\forall$.
- Executions of alternating TMs are defined using a labeling procedure on the configuration graph—see Section 4.1.1. in the book [1]. Repeatedly apply the following rules until they cannot be applied anymore.
  - Label each configuration with q_{acc} with "accept."
  - If a configuration c with $q \in Q_\exists$ has an edge to a configuration c' that is labeled with "accept," then label c with "accept."
  - If a configuration c has a state $q \in Q_\forall$ and both configurations c', c'' that are reachable from it in the graph are labeled with "accept," then label c with "accept."

  We say that the TM accepts the input if at the end of this process the starting configuration is labeled with "accept."
- Let $T : \mathbb{N} \to \mathbb{N}$. The TM runs in time $T(n)$ if for every input x and for every possible sequence of transition function choices, the machine halts after at most $T(|x|)$ steps.

Definition 7 (ATIME)**.** We define $\text{ATIME}(T(n))$ to be the set of languages that are accepted by an alternating Turing machine that runs in time $O(T(n))$.

Definition 8 (Σ_iTIME)**.** Let $i \geq 1$. We define $\Sigma_i\text{TIME}(T(n))$ to be the set of languages that are accepted by an alternating Turing machine that runs in time $O(T(n))$, whose initial state is in $Q_\exists$, and that on every input and on every path in the configuration graph alternates at most $i-1$ times between $Q_\exists$ and $Q_\forall$.

Definition 9 (Π_iTIME)**.** Let $i \geq 1$. We define $\Sigma_i\text{TIME}(T(n))$ to be the set of languages that are accepted by an alternating Turing machine that runs in time $O(T(n))$, whose initial state is in $Q_\forall$, and that on every input and on every path in the configuration graph alternates at most $i-1$ times between $Q_\exists$ and $Q_\forall$.

References

[1] Sanjeev Arora and Boaz Barak. *Computational Complexity – A Modern Approach.* Cambridge University Press, 2009.