Complexité avancée - TD 4

Guillaume Scerri

October 22, 2026

Exercise 1 (Oracles)**.** Consider a language A. A Turing machine with oracle A is a Turing machine with a special additional read/write tape, called the oracle tape, and three special states: $q_{query}, q_{yes}, q_{no}$. Whenever the machine enters the state q_{query}, with some word w written on the oracle tape, it moves **in one step** to the state q_{yes} or q_{no} depending on whether $w \in A$.

We denote by P^A (resp. NP^A) the class of languages decided in by a deterministic (resp. non-deterministic) Turing machine running in polynomial time with oracle A. Given a complexity class $\mathcal{C}$, we define $\mathsf{P}^{\mathcal{C}} = \bigcup_{A \in \mathcal{C}} \mathsf{P}^A$ (and similarly for NP).

1. Prove that for any $\mathcal{C}$-complete language A (for logspace reductions), $\mathsf{P}^{\mathcal{C}} = \mathsf{P}^A$ and $\mathsf{NP}^{\mathcal{C}} = \mathsf{NP}^A$.
2. Show that for any language A, $\mathsf{P}^A = \mathsf{P}^{\bar{A}}$ and $\mathsf{NP}^A = \mathsf{NP}^{\bar{A}}$.
3. Prove that if $\mathsf{NP} = \mathsf{P}^{\mathsf{SAT}}$ then $\mathsf{NP} = \mathsf{coNP}$.
4. Show that there exists a language A such that $\mathsf{P}^A = \mathsf{NP}^A$.
5. We define inductively the classes $\mathsf{NP}_0 = \mathsf{P}$ and $\mathsf{NP}_{k+1} = \mathsf{NP}^{\mathsf{NP}_k}$. Show that $\mathsf{NP}_k = \Sigma_k^p$ for all $k \geq 0$.

Exercise 2 (Σ_2^p and Π_2^p membership)**.**

1. Let $\mathsf{ONE-VAL}$ be the problem of deciding whether a boolean formula is satisfied by exactly one valuation. Show that $\mathsf{ONE-VAL} \in \Sigma_2^p$;
2. A boolean formula is minimal if it has no equivalent shorter formula – where the length of the formula is the number of symbols it contains. Let $\mathsf{MIN-FORMULA}$ be the problem of deciding whether a boolean formula is minimal. Show that $\mathsf{MIN-FORMULA} \in \Pi_2^p$.

Exercise 3 (Σ_k^p completeness)**.** We define Σ_k-QBF as the class of tautologies of the form $\exists \bar{x}_1 . \forall x_2 . \cdots \varphi$ where the alternation of quantifiers has length k. Show that Σ_k-QBF is Σ_k^p complete.

Exercise 4 (Σ_2^p and Π_2^p completeness)**.**

1. The classical Σ_2^p-complete problem is Σ_2^p-SAT (note that it can be assumed that the Boolean formula is in DNF). Consider now a different version of SAT denoted $\exists\exists! - \mathsf{SAT}$:

   - Input: a CNF-formula $\varphi(x, y)$ depending on the variables in x and y;
   - Outout: yes iff there exists x such that there exists a unique y satisfying $\varphi(x, \cdot)$.

   Show that $\exists\exists! - \mathsf{SAT}$ is also Σ_2^p-complete.

2. Similarly, the classical Π_2^p-complete problem is Π_2^p-SAT (the Boolean formula can be assumed in 3-CNF). Consider now a new notion of satisfiability: we say that a valuation ν nae-satisfies (for *not all equal*) a 3-CNF formula ϕ, if in all clauses (with at least two literals) of ϕ, ν both sets a literal to true and a literal to false. The clauses with only one literal only need to be satisfied. We consider now this new version of SAT denoted nae-$\Pi_2^p - \mathsf{SAT}$:

   - Input: a Π_2^p-SAT formula $\forall x, \exists y,\ \varphi(x, y)$ with φ a 3-CNF;
   - Outout: yes iff for all x, there exists y nae-satisfying φ.

   Show that nae-$\Pi_2^p - \mathsf{SAT}$ is Π_2^p complete.

Exercise 5 (Collapse of PH)**.** 1. Prove that if $\Sigma_k^P = \Sigma_{k+1}^P$ for some $k \geq 0$ then $\mathsf{PH} = \Sigma_k^P$. (Remark that this is implied by $\mathsf{P} = \mathsf{NP}$).

2. Show that if $\Sigma_k^P = \Pi_k^P$ for some k then $\mathsf{PH} = \Sigma_k^P$.

3. Show that if $\mathsf{PH} = \mathsf{PSPACE}$ then PH collapses.

4. Do you think there is a polynomial time procedure to convert any QBF formula into a QBF formula with at most 10 variables ?