Complexité avancée - TD 6

Guillaume Scerri

October 23, 2025

Exercise 1 (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 ?

Exercise 2 (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 3 (Deciding QBF with unknown oracles)**.** Assume that you are given two oracles A and B and one of them decides QBF (but you do not know which one). Exhibit a polynomial time algorithm with access to A and B that decides QBF in polynomial time.

Exercise 4 (Non deterministic Turing machine alternative definition)**.** An *SNDTM* (strong non-deterministic Turing machine) is a non-deterministic Turing machine with three possible outputs: yes, no and ?. An SNDTM decides a language L if, on all inputs x:

- if $x \in L$ then all computations either yield 1 or ? and there is at least one that yields 1.
- if $x \notin L$, then all computations either yield 0 or ? and there is at least one that yields 0.

Show that any language L is decided by an SNDTM in polynomial time if and only if $L \in \mathsf{NP} \cap co\mathsf{NP}$.

Exercise 5 (About PP)**.** Define PP as the class to probabilistic Turing machines such that :

$$x \in L \quad \Rightarrow \quad P[M(x,r) \text{ accepts }] > 1/2$$
$$x \notin L \quad \Rightarrow \quad P[M(x,r) \text{ accepts }] \leq 1/2$$

Define PP' as the class to probabilistic Turing machines such that :

$$x \in L \quad \Rightarrow \quad P[M(x,r) \text{ accepts }] \geq 1/2$$

$$x \notin L \quad \Rightarrow \quad P[M(x,r) \text{ accepts }] < 1/2$$

Prove the following statements:

1. $\mathsf{NP} \subseteq \mathsf{PP}$
2. $\mathsf{PP} = \mathsf{PP}'$
3. PP is closed under complement
4. PP has complete problems

Exercise 6 (PP vs $\sharp\mathsf{P}$)**.** 1. Prove that $f \in \sharp\mathsf{P}$ iff there exists a probabilistic Turing machine runing in polynomial time $t(n)$ with random tape of size $t(n)$ such that for all $x \in \Sigma^*$ we have

$$f(x) = |\{r||r| = t(n) \text{ and } M(x,r) \text{ accepts}|$$

2. Prove that $f \in \sharp\mathsf{P}$ iff there exists a polynomial time computable relation R (i.e. R computable in polynomial time and there exists a polynomial p such that if $(x,y) \in R$ then $|y| \leq p(|x|)$) such that

$$f(x) = |\{y|(x,y) \in R\}|$$

3. Prove that $\mathsf{P}^{\mathsf{PP}} = \mathsf{P}^{\sharp\mathsf{P}}$
4. Prove that $\mathsf{P} = \mathsf{PP} \quad \Leftrightarrow \quad \sharp\mathsf{P} = \mathsf{FP}$

Exercise 7 (Parsimonious reductions)**.** 1. Show that $\sharp 3\mathsf{SAT}$ is $\sharp\mathsf{P}$ complete.

2. If $g \leq f$ and $f \leq h$ then $f \leq h$ under parsimonious reductions.
3. Is $\sharp\mathsf{P}$ closed under parsimonious reductions ?

Exercise 8 (Non deterministic Turing machine alternative definition)**.** An *SNDTM* (strong non-deterministic Turing machine) is a non-deterministic Turing machine with three possible outputs: yes, no and ?. An SNDTM decides a language L if, on all inputs x:

- if $x \in L$ then all computations either yield 1 or ? and there is at least one that yields 1.
- if $x \notin L$, then all computations either yield 0 or ? and there is at least one that yields 0.

Show that any language L is decided by an SNDTM in polynomial time if and only if $L \in \mathsf{NP} \cap co\mathsf{NP}$.

Exercise 9 (A quick come back to the polynomial hierarchy)**.** Show that:

1. $\mathsf{NP} \subseteq \mathsf{P}^{\mathsf{SAT}} \subseteq \Sigma_2^{p'} \cap \Pi_2^{p'}$;
2. $\mathsf{NP}^{\mathsf{NP}} = \Sigma_2^{p'}$;

(Bonus). $\mathsf{NP}^{\Sigma_2^{p'}} = \Sigma_3^{p'}$ and $\mathsf{NP}^{\Sigma_3^{p'}} = \Sigma_4^{p'}$.