Complexité avancée - TD 5

Benjamin Bordais

October 21, 2020

Exercise 1 Family of circuits

Definition 1 *A* boolean circuit with n inputs *is an acylic graph where the n inputs $x_1, \ldots, x_n$ are part of the vertices. The internal vertices are labeled with $\wedge$, $\vee$ (with 2 incoming edges) or $\neg$ (with 1 incoming edge), with an additional distinguished vertex o that is the output (with no exiting edge). The size $|C|$ of a circuit C is its number of vertices (excluding the input ones). For a word $x \in \{0,1\}^*$, the notation $C(x)$ refers to the output of the circuit C if the input vertices of C are valued with the bits of x.*

Definition 2 *For a function $t : \mathbb{N} \to \mathbb{N}$, a* family of circuit of size $t(n)$ *is a sequence $(C_n)_{n\in\mathbb{N}}$ such that: C_n is an n-input circuit and $|C_n| \leq t(n)$.*

Definition 3 *A language $L \subseteq \{0,1\}^*$ is* decided by a family of circuit $(C_n)_{n\in\mathbb{N}}$ *if for all $n \in \mathbb{N}$, for all $w \in \{0,1\}^n$, we have: $C_n(w) = 1 \Leftrightarrow w \in L$.*

Definition 4 *For a function $t : \mathbb{N} \to \mathbb{N}$, we define $\mathsf{SIZE}(t) := \{L \subseteq \{0,1\}^* \mid L$ is decided by a family of circuits of size $O(t(n))\}$.*

Definition 5

$$\mathsf{P}/poly := \cup_{k\in\mathbb{N}}\mathsf{SIZE}(n^k)$$

1. Show that any language $L \subseteq \{0,1\}^*$ is in size $\mathsf{SIZE}(n \cdot 2^n)$.
2. Show that for all function $t(n) = 2^{o(n)}$, there exists $L \notin \mathsf{SIZE}(t(n))$.
3. Show that every unary language is in $\mathsf{P}/poly$.
4. Exhibit a undecidable language that is in $\mathsf{P}/poly$.
5. Show that $\mathsf{P}/poly$ is not countable.

Exercise 2 Some alternation

1. Exhibit a polynomial time alternating algorithm that solves QBF.
2. Let $\mathsf{ONE-VAL}$ be the problem of deciding whether a boolean formula is satisfied by exactly one valuation. Show that $\mathsf{ONE-VAL} \in \Sigma_2^p$.
3. 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 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 $\mathsf{PH} = \Sigma_k^P$ (*i.e.* PH collapses).

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 4 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 $\mathsf{P}^A = \mathsf{NP}^A$.[1]

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 $k \geq 0$.

[1] In fact, there also exists a language B such that $\mathsf{P}^B \neq \mathsf{NP}^B$, which does not prove that $\mathsf{P} \neq \mathsf{NP}$.