Complexité avancée - TD 6

Simon Halfon

October 19, 2016

Exercise 1: Complete problems for levels of PH

Give high-level arguments that show the following problem Σ_k^P-complete (under polynomial time reductions).

$\Sigma_k\mathtt{QBF}$:
- INPUT: A quantified boolean formula $\exists X_1 \forall X_2 \exists ... Q_k X_k \phi$, where $X_1, ..X_k$ are k disjoint sets of variables, Q_k is the quantifier $\forall$ if k is even, and the quantifier $\exists$ if k is odd, ϕ is a boolean formula over variables $X_1 \cup \cdots \cup X_k$;
- QUESTION: is the input formula true ?

Define a similar problem $\Pi_k\mathtt{QBF}$ such that $\Pi_k\mathtt{QBF}$ is Π_k^P-complete.

Exercise 2: Collapse of PH

1. Show that if $\Sigma_k^P = \Pi_k^P$ for some k then $\mathsf{PH} = \Sigma_k^P$ (*i.e.* PH collapses).
2. Show that if $\mathsf{P} = \mathsf{NP}$ then $\mathsf{P} = \mathsf{PH}$.
3. Prove that if $\Sigma_k^P = \Sigma_{k+1}^P$ for some $k \geq 0$ then $\mathsf{PH} = \Sigma_k^P$.
4. Show that if $\mathsf{PH} = \mathsf{PSPACE}$ then PH collapses.
5. Do you think there is a polynomial time procedure to convert any QBF formula into a QBF formula with at most 10 variables ?

Exercise 3: Oracle machines

Let O be a language. A Turing machine with oracle O 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 $w \in O$.

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

1. Prove that for any $\mathcal{C}$-complete language L (for polynomial time reductions), $\mathsf{P}^{\mathcal{C}} = \mathsf{P}^L$ and $\mathsf{NP}^{\mathcal{C}} = \mathsf{NP}^L$.
2. Show that for any language L, $\mathsf{P}^L = \mathsf{P}^{\bar{L}}$ and $\mathsf{NP}^L = \mathsf{NP}^{\bar{L}}$.
3. Prove that if $\mathsf{NP} = \mathsf{P}^{SAT}$ then $\mathsf{NP} = \mathsf{coNP}$.
4. Prove that $\Sigma_{k+1}^P = \mathsf{NP}^{\Sigma_k^P}$. Give an oracle characterization of Π_k^P.

This third view of the polynomial hierarchy gives access to the class of languages $\Delta_{k+1}^P = \mathsf{P}^{\Sigma_k^P}$, with $\Delta_0^P = \mathsf{P}$.

Exercise 4: Relativization

Show that there is an oracle O such that $\mathsf{P}^O = \mathsf{NP}^O$.

Exercise 5: Sort your problems

Give complexity upper bounds for the following problems:

1. `MIN-FORMULA`
   - INPUT: a propositional formula ϕ
   - QUESTION: is ϕ minimal, in the sense that there exists no smaller formula equivalent to ϕ ?
2. `MAX-CLIQUE`
   - INPUT: a graph G and a natural number k
   - QUESTION: k is the exact size of a maximal clique in G
3. `USAT`
   - INPUT: a boolean formula ϕ
   - QUESTION: is ϕ satisfiable by only one assignment

Exercise 6: Which one is lying ?

Suppose that a Turing machine has access to two oracles A and B, one of which is an oracle for `QBF`, but you don't know which. Show that `QBF` can still be decided in polynomial time by such a machine.

Exercise 7: Bounded number of queries to the oracle

A k-query oracle Turing machine is an oracle Turing machine that can access the q_{query} state at most k times. Given an oracle O, define $\mathsf{P}^{O,k}$ the class of languages that can be decided in deterministic-polynomial time by a k-query Turing machine with oracle O.

1. Show that $\mathsf{NP} \cup \mathsf{coNP} \subseteq P^{SAT,1}$.
2. Assuming $\mathsf{NP} \neq \mathsf{coNP}$, prove that the first inclusion above is strict.

Exercise 8: The Difference Hierarchy

Let DP be the class of languages of the form $L_1 \cap L_2$, where $L_1 \in \mathsf{NP}$ and $L_2 \in \mathsf{coNP}$. (In other words a language in DP is the difference of two NP languages.)

We consider the problem `EXACT-INDSET`:

- INPUT: a graph G and an integer k
- OUTPUT: does the maximum size of an independent set of G is k ? That is whether G has an independent set of size k, and all other independent sets of G have size at most k. Recall that an independent set of a graph is a set I of vertices such that no two vertices of I are connected by an edge.

.

1. Show that $\mathsf{P}^{SAT,1} \subseteq \mathsf{DP} \subseteq \mathsf{P}^{SAT,2}$.
2. Prove that `EXACT-INDSET`:
   (a) is in $\Sigma_2^P \cap \Pi_2^P$;
   (b) is in DP;
   (c) is DP-complete (under polynomial time reductions).