Advanced Complexity

TD n°5

Charlie Jacomme

October 11, 2017

Exercise 1 : P-choice

A language L is said P-peek ($L \in \mathsf{P}p$) if there is a function $f : \{0,1\}^* \times 0,1^* \to \{0,1\}^*$ computable in polynomial time such that $\forall x, y \in \{0,1\}^*$:
— $f(x,y) \in \{x,y\}$
— if $x \in L$ or $y \in L$ then $f(x,y) \in L$
f is called the peeking function for L.

1. Show that $\mathsf{P} \subseteq \mathsf{P}p$
2. Show that $\mathsf{P}p$ is closed under complementary
3. Show that if there exist L NP-hard in $\mathsf{P}p$, then $\mathsf{P} = \mathsf{NP}$
4. Let $r \in [0,1]$ a real number, we define L_r as the set of words $b = b_1...b_n \in \{0;1\}^*$ such that $0, b_1...b_n \leq r$. Show that $L_r \in \mathsf{P}p$
5. Deduce that there exist a non-recursive language in $\mathsf{P}p$

Solution:

1. Let there be $A \in \mathsf{P}$. We set $f(x,y) = x$ if $x \in A$, and $f(x,y) = y$ otherwise.
2. Let there be $A \in \mathsf{P}p$ through f. Then, define $f'(x,y) = y$ if $f(x,y) = x$ and $f'(x,y) = x$ otherwise. f' is then a peeking function for A^c :
   — if $x \in A^c$ and $y \in A^c$, then $f'(x,y) = y \in A^c$
   — if $x \in A^c$ and $y \in A$, then $f(x,y) = y$ and $f'(x,y) = x \in A^c$
   — if $x \in A$ and $y \in A^c$, then $f(x,y) = x$ and $f'(x,y) = y \in A^c$ contains a language which is undecidable.
3. Let there be $A \in \mathsf{P}p$ through f and g a reduction from SAT to A. Here is a polynomial algorithm for SAT on input ϕ with n variables, where we denote ϕ_0 (resp. ϕ_1) the formula ϕ in which the first variable is set to 0 (resp. 1).

   ```
   For i from 1 to n do
     if f(g(φ0), g(φ1)) = g(φ0) then φ ← φ0
     else φ ← φ1
   Accept iff φ = ⊤
   ```

4. $f(x,y) = min(x,y)$ is a valid selection function for L_r
5. $\mathsf{P}p$ is not countable as it contains L_r for any $r \in [0,1]$. Thus, $\mathsf{P}p$ contains a language which is not decidable.

Exercise 2 : Complete problems for levels of PH

Show that the following problem is Σ_k^P-complete (under polynomial time reductions).

Σ_k`QBF` : • INPUT : A quantified boolean formula $\psi := \exists X_1 \forall X_2 \exists ... Q_k X_k \phi(X_1, ..., X_k)$, 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 $\Pi_k\texttt{QBF}$ such that $\Pi_k\texttt{QBF}$ is Π_k^P-complete.

Solution:

— If we are given some $X_1, ..., X_k$, we can check in polynomial time if $\phi(X_1, ..., X_k)$ is true. Thus, it is in Σ_k^P.

— Let there be $A \in \Sigma_k^P$. A can be expressed as follows :

$$x \in A \Leftrightarrow \exists y_1 \in \{0,1\}^{p|(x)|} \forall y_2 \in \{0,1\}^{p|(x)|} ... Q_k y_k \in \{0,1\}^{p|(x)|}(x, y_1, ..., y_k) \in B$$

with $B \in P$.

Let us assume that $Q_k = \exists$, the other case can be done in a similar fashion. Now, $\exists y_k \in \{0,1\}^{p|(x)|}(x, y_1, ..., y_k) \in B$ is in NP, so by Cook's theorem, we have ϕ such that :

$$\exists y_k \in \{0,1\}^{p|(x)|}(x, y_1, ..., y_k) \in B \Leftrightarrow \exists z, \phi_{x,y_1,...,y_{k-1}}(z)$$

By inspecting Cook's proof, we can modify ϕ such that the input tape $x, y_1, ..., y_{k-1}$ appear as variables in ϕ. We thus have

$$\exists y_k \in \{0,1\}^{p|(x)|}(x, y_1, ..., y_k) \in B \Leftrightarrow \exists z, \phi(x, y_1, ..., y_{k-1}, z)$$

And finally :

$$x \in A \Leftrightarrow \exists y_1, \forall y_2, ... \forall y_{k-1} \exists z, \phi(x, y_1, ..., y_{k-1}, z)$$

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, $\mathsf{P}^{\mathcal{C}} = \mathsf{P}^L$ 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 $\mathsf{NP} = \mathsf{coNP}$.

Solution:

1. We do the proof for NP. Let $B \in \mathsf{NP}^{\mathcal{C}}$, we have N a polynomial NTM for B with an oracle C, $C \in \mathcal{C}$. We also have a polynomial reduction f such that : $x \in \mathcal{C} \Leftrightarrow f(x) \in A$. We build N' for B with oracle A, by simulating N and replacing a call $u \in C$? with a call $f(u) \in A$?. f is polynomial, so we are still in NP, which concludes the proof.
2. We simply have to swap the states q_{yes} and q_{no} in the computation.
3. P^{SAT} is a deterministic class, so it is closed by complementation, so if $\mathsf{NP} = \mathsf{P}^{SAT}$, $\mathsf{coNP} = \mathsf{NP}$
4.

Exercise 4 : 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 ?

Solution:

1. We assume that $\Sigma_k^P = \Sigma_{k+1}^P$ for some $k \geq 0$, we prove by induction that $\forall t \geq k, \Sigma_k^P = \Sigma_j^P$, For $j = i$, it is directly correcT. For $j > i$, $\Sigma_j^P = \mathsf{NP}^{\Sigma_{j-1}^P} = \mathsf{NP}^{\Sigma_i^P}$ by induction, and thus $\Sigma_j^P = \Sigma_{i+1}^P$. By hypothesis, we then have $\Sigma_j^P = \Sigma_i^P$
2. With the previous question, we just have to prove that $\Sigma_k^P = \Sigma_{k+1}^P$.
   Let there be $A \in \Sigma_{k+1}^P$. A can be expressed as follows :
   $$x \in A \Leftrightarrow \exists y_1 \in \{0,1\}^{p|(x)|} \forall ... Q_{k+1} y_{k+1} \in \{0,1\}^{p|(x)|} (x, y_1, ..., y_{k+1}) \in B$$
   with $B \in \mathsf{P}$.
   On input, (x, y_1), decide if $\forall y_2 \in \{0,1\}^{p|(x)|} ... Q_{k+1} y_{k+1} \in \{0,1\}^{p|(x)|} (x, y_1, ..., y_{k+1}) \in B$ is a problem in $\Pi_k^P = \Sigma_k^P$. We can thus rewrite it as, with $C \in \mathsf{P}$:
   $$\exists y_2 \in \{0,1\}^{p|(x)|} ... \overline{Q}_{k+1} y_{k+1} \in \{0,1\}^{p|(x)|} (x, y_1, ..., y_{k+1}) \in C$$
   Finally :
   $$x \in A \Leftrightarrow \exists y_1, y_2 \in \{0,1\}^{p|(x)|} \forall ... \overline{Q}_{k+1} y_{k+1} \in \{0,1\}^{p|(x)|} (x, y_1, ..., y_{k+1}) \in B$$
   with $B \in \mathsf{P}$. And this is the expression of a problem in Σ_k^P. Finally, $\Sigma_k^P = \Sigma_{k+1}^P$.
3. If $\mathsf{PH} = \mathsf{PSPACE}$, then QBF is in Σ_k^P for some k. But QBF is a complete problem for PSPACE, and thus PH. Let there be $B \in \mathsf{PH}$, it can be reduced to $QBF \in \Sigma_k^P$, so $B \in \Sigma_k^P$, and $\mathsf{PH} = \Sigma_k^P$
4. It is unlikely that PH collapses, and the statement would imply the previous question.

Exercise 5 : Relativization

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

Solution:

We have $\mathsf{PSPACE} \subseteq \mathsf{NP}^{\mathsf{PSPACE}}$, we must show the converse. Let there be N a polynomial NTM with oracle $A \in \mathsf{PSPACE}$. We can simulate N in PSPACE on input x by :

- enumerate all possible path of $N^A(x)$
- For each of them, compute the oracle calls
- accept if one of the path accepts.

Each path is in polynomial size, thus the enumeration is, and the oracle calls are PSPACE. We do have $\mathsf{NP}^{\mathsf{PSPACE}} \subseteq \mathsf{PSPACE}$.