Advanced Complexity

TD n°5

Charlie Jacomme

October 11, 2017

Exercise 1 : Unary Languages

1. Prove that if a unary language is NP-complete, then P = NP.
   Hint : consider a reduction from SAT to this unary language and exhibit a polynomial time recursive algorithm for SAT
2. Prove that if every unary language in NP is actually in P, then EXP = NEXP.

Exercise 2 : On the existence of one-way functions

A one-way function is a bijection f from k-bit intergers to k-bit intergers such that f is computable in polynomial time, but f^{-1} is not. Prove that if there exists one-way functions, then

$$A = \{(x, y) \mid f^{-1}(x) < y\} \in (\mathsf{NP} \cap \mathsf{coNP})\backslash\mathsf{P}$$

Exercise 3 : Prime Numbers

1. Show that `UNARY-PRIME` $= \{1^n \mid n$ is a prime number $\}$ is in P.
2. Show that `PRIME` $= \{p|p$ is a prime number encoded in binary $\}$ is in coNP.
3. We want to prove that `PRIME` is in NP. Use the following characterization of prime numbers to formulate a non-deterministic algorithm runing in polynomial time.
   A number p is prime if and only if there exists $a \in [2, p-1]$ such that :
   (a) $a^{p-1} \equiv 1[p]$, and
   (b) for all q prime divisor of $p-1$, $a^{\frac{p-1}{q}} \not\equiv 1[p]$
   To prove that your algorithm runs in polynomial time, you can admit that all common arithmetical operations on $\mathbb{Z}/p\mathbb{Z}$ can be performed in polynomial time.

Exercise 4 : Some P-complete problems

Show the following problems to be P-complete :

1. — INPUT : A set X, a binary operator $*$ defined on X, a subset $S \subset X$ and $x \in X$
   — QUESTION : Does x belongs to the closure of S with respect to $*$?
   Hint : for the hardness, reduce from Monotone Circuit Value
2. — INPUT : G a context-free grammar, and w a word
   — QUESTION : $w \in \mathcal{L}(G)$?
   Hint : for the hardness, reduce from the previous problem

Exercise 5 : 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 $\mathsf{P}p$

Exercise 6 : 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 Π_k`QBF` such that Π_k`QBF` is Π_k^P-complete.

Exercise 7 : 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 $\mathsf{NP} = \mathsf{coNP}$.

Exercise 8 : 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 9 : Relativization

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