Tutorial 03 – Respect the hierarchy!

Exercise 1.

1. Prove that coP = P.
2. Prove that $L \in$ coNP iff there exists a polyomial P and a deterministic TM M working in polynomial time s.t.

$$x \in L \iff \forall u \in \{0,1\}^{p(|x|)}, M(x,u) = 1.$$

3. Prove that if there exists a NP language which is coNP-hard, then NP = coNP.
4. Prove that a language L is NP-complete iff $\bar{L}$ is coNP-complete.

Exercise 2.

1. Prove that $\mathsf{DTIME}(2^{n+k}) = \mathsf{DTIME}(2^{n+l})$ for all $l > k > 0$.
2. Prove that $\mathsf{DTIME}(2^{n^k}) \subsetneq \mathsf{DTIME}(2^{n^l})$ for all $l > k > 0$.

Exercise 3. *The H in Ladner*
In the proof of Ladner's theorem, the H-function is defined as follows: $H(n)$ is the smallest integer $i < \log\log n$ s.t. for all $x \in \{0,1\}^*$ with $|x| \leq \log n$, the TM M with code i decides whether $x \in \textsc{Sat}_H$ within $i|x|^i$ steps, or $\log\log n$ if no such i exists. The language $\textsc{Sat}_H$ is defined by $\left\{\psi 01^{n^{H(n)}} : \psi \in \textsc{Sat} \text{ and } |\psi| = n\right\}$.

✎ Prove that H is polynomial-time (in n) computable.

⋆ **Exercise 4.** *Mahaney's theorem (1982)*
Definition. A language L is said sparse if there exists a polynomial p s.t., for all n, $L \cap \Sigma^n$ has cardinality at most $p(n)$.

1. Let L be a sparse language. What can you say about the cardinality of $L \cap \Sigma^{\leq n}$?

We will show that if there exists a sparse NP-hard language L, then P = NP. Let L be such a language, and let X be in NP:

$$x \in X \text{ iff } \exists w \in \Sigma^{p(|x|)}, \langle x, w \rangle \in A$$

with p a polynomial and $A \in$ P. The aim is to prove that X is polynomial-time decidable.
Let $G(A) = \{\langle x, w \rangle : \exists y \in \Sigma^{p(|x|)}, y \geq w \text{ and } \langle x, y \rangle \in A\}$.

2. Prove that $G(A)$ is in NP.

3. Using a reduction from $G(A)$ to L, prove that X is polynomial-time decidable. **Hint.** One can find a polynomial-time algorithm which, on input x, find the longest w such that $\langle x, w \rangle \in A$ if it exists.

** **Exercise 5.** *Nondeterministic Time Hierarchy Theorem (Cook 1972)*

Theorem. *Let f and g be two time-constructible functions s.t. $f(n+1) = o(g(n))$. Then*

$$\mathsf{NTIME}(f(n)) \subsetneq \mathsf{NTIME}(g(n)).$$

In the sequel, suppose that $f(n+1) = o(g(n))$. We will prove this theorem.

1. Remind the idea behind the proof of the Deterministic Time Hierarchy Theorem, and explain why this proof cannot be adapted here.

2. Explain how effectively enumerate the NDTM working in time $O(f(n))$.

We will use a *lazy diagonalization*. Habitually, to diagonalize, one tries to "eliminate" the machine M_i on input i. In this lazy version, one tries to eliminate M_i not on a precise input, but on one of the inputs of a set I_i.
To each machine M_i in the previous enumeration is associated a tally set $I_i = \{1^k : \alpha_i \leq \beta_i\}$ where α_i and β_i have to be defined later. Let N the following NDTM: on input x, N finds i s.t. $x \in I_i$, then

1. If $x \in I_i \setminus \{1^{\beta_i}\}$, N emulates $M_i(x \cdot 1)$ in a nondeterministic way, stopping within $g(|x|)$ steps, and accepts iff M_i stopped and accepted within this time;

2. If $x = 1^{\beta_i}$, N emulates $M_i(1^{\alpha_i})$ **in a deterministic way**, and answers the contrary of M_i.

3. How to choose α_i et β_i so that $L(N) \in \mathsf{NTIME}(g(n))$? **Hint.** Find i s.t. $x \in I_i$ has to be *fast enough* and step **(b)** as well.

4. Suppose that $L(N) \in \mathsf{NTIME}(f(n))$, through a NDTM M. Prive that there exists i s.t. $M = M_i$ and s.t. at step **(a)**, M_i is always emulated until it stops.

5. Conclude.