Tutorial 04 – Oracles

Exercise 1. *Mme de Pompadour* [1]

1. Show that $\textsc{Tautology} \in \mathsf{P}^{\textsc{Sat}}$.
2. Show that for every $A \in \mathsf{P}$, $\mathsf{P}^A = \mathsf{P}$.
3. Let $E = \{\langle \alpha, x, 1^n \rangle : M_\alpha \text{ outputs 1 on } x \text{ within } 2^n \text{ steps}\}$. Show that $\mathsf{NP}^E \subseteq \mathsf{EXP}$ (this concludes the proof that $\mathsf{P}^E = \mathsf{NP}^E = \mathsf{EXP}$, sketched in the lecture).

Exercise 2. *Turing jump*

For a language X, let us define $X' = \{\alpha : L(M_\alpha^X) = \emptyset\}$.

1. Let $A = \{\langle \alpha, w \rangle : M_\alpha(w) = 1\}$ and $B = \{\alpha : L(M_\alpha) = \emptyset\}$. Show that A is decidable by an Oracle Turing Machine with oracle B, and conversely.
2. Show that X' is not decidable by an OTM with oracle X.

Exercise 3. *Karp vs. Cook-Turing*

The *Cook-Turing polynomial-time reduction* is defined by $A \leq_T^p B$ if $A \in \mathsf{P}^B$. In the sequel, $\leq_m^p$ denotes the usual polynomial-time reduction, called *many-one reduction* or *Karp reduction*.

1. Show that $A \leq_T^p B$ and $B \leq_T^p C$ implies $A \leq_T^p C$.
2. Show that for every A, $\bar{A} \leq_T^p A$.
3. Show that if $\textsc{Tautology} \leq_m^p \textsc{Sat}$, then $\mathsf{NP} = \mathsf{coNP}$.
4. Show that $\mathsf{NP} = \mathsf{coNP}$ iff NP is closed under $\leq_T^p$. ($A \leq_T^p B$ and $B \in \mathsf{NP} \implies A \in \mathsf{NP}$)
5. How are $\leq_m^p$ and $\leq_T^p$ related to each other ?

** **Exercise 4.** *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. Why did I choose this title ?

   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.