TD 03 – Respect de la hiérarchie

Exercice 1. *(Le) cours, Forest ! (Le) cours !*

1. Montrer que $\mathsf{coP} = \mathsf{P}$.
2. Montrer que $L \in \mathsf{coNP}$ ssi il existe un polynôme P et une MT M déterministe fonctionnant en temps polynomial tels que
$$x \in L \iff \forall u \in \{0,1\}^{p(|x|)}, M(x,u) = 1.$$
3. Montrer que s'il existe un langage de NP qui est coNP-difficile, alors $\mathsf{NP} = \mathsf{coNP}$.
4. Montrer qu'un langage L est NP-complet ssi $\bar{L}$ est coNP-complet.

Exercice 2. *Hier ? Archi !* [1]

1. Montrer que $\mathsf{DTIME}(2^{n+k}) = \mathsf{DTIME}(2^{n+l})$ pour tous $l > k > 0$.
2. Montrer que $\mathsf{DTIME}(2^{n^k}) \subsetneq \mathsf{DTIME}(2^{n^l})$ pour tous $l > k > 0$.

Exercice 3. *Le H de Ladner*

Dans la preuve du théorème de Ladner, la fonction H est définie comme suit : $H(n)$ est le plus petit entier $i < \log\log n$ tel que pour tout $x \in \{0,1\}^*$ vérifiant $|x| \leq \log n$, la machine de Turing M de code i décide si $x \in \textsc{Sat}_H$ en au plus $i|x|^i$ étapes, ou $\log\log n$ si aucun tel i n'existe. Le langage $\textsc{Sat}_H$ est lui défini par $\left\{\psi 01^{n^{H(n)}} : \psi \in \textsc{Sat} \text{ and } |\psi| = n\right\}$.

✎ Montrer que H est calculable en temps polynomial en n.

⋆ **Exercice 4.** *Théorème de Mahaney (1982)*

Définition. Un langage L est dit creux lorsqu'il existe un polynôme p tel que, pour tout n, $L \cap \Sigma^n$ est de cardinal au plus $p(n)$.

1. Soit un langage creux L. Que pouvez-vous dire du cardinal de $L \cap \Sigma^{\leq n}$?

Nous allons montrer que s'il existe un langage L creux et NP-difficile, alors $\mathsf{P} = \mathsf{NP}$. Soit donc un tel langage L et soit X dans NP :
$$x \in X \text{ ssi } \exists w \in \Sigma^{p(|x|)}, \langle x, w\rangle \in A$$
avec p un polynôme et $A \in \mathsf{P}$. On veut montrer que X est décidable en temps polynomial. Soit $G(A) = \{\langle x, w\rangle : \exists y \in \Sigma^{p(|x|)}, y \geq w \text{ et } \langle x, y\rangle \in A\}$.

2. Montrer que $G(A)$ est dans NP.

1. Si vous trouvez mieux, je prends…

3. En utilisant une réduction de $G(A)$ à L, montrer qu'on peut décider X en temps polynomial. **Indication.** On pourra déterminer un algo polynomial qui, sur l'entrée x, trouve le plus grand w tel que $\langle x, w\rangle \in A$ lorsqu'il existe.

** **Exercice 5.** *Théorème de hiérarchie en temps non déterministe (Cook 1972)*

Théorème. *Soit f et g deux fonctions constructibles en temps telles que $f(n+1) = o(g(n))$. Alors*

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

Dans la suite on suppose donc que $f(n+1) = o(g(n))$, et on veut montrer ce théorème.

1. Rappeler l'idée de la preuve du théorème de hiérarchie en temps déterministe et expliquer pourquoi cette preuve ne peut pas simplement être adaptée au cas présent.
2. Expliquer comment effectivement énumérer les MTND fonctionnant en temps $O(f(n))$.

On va utiliser une *diagonalisation paresseuse*. D'habitude pour diagonaliser, on cherche à « éliminer » la machine M_i sur l'entrée i. Dans la version paresseuse, on cherche à éliminer M_i non pas sur une entrée bien précise mais sur l'une des entrées d'un ensemble I_i d'entrées.
A chaque machine M_i de l'énumération précédente, on associe un ensemble unaire $I_i = \{1^k : \alpha_i \le \beta_i\}$ où α_i et β_i seront définis plus tard. Soit N la MTND suivante : sur l'entrée x, N détermine i tel que $x \in I_i$, puis

1. si $x \in I_i \setminus \{1^{\beta_i}\}$, N simule $M_i(x \cdot 1)$ de manière non déterministe en s'arrêtant au bout d'au plus $g(|x|)$ étapes, et accepte ssi M_i s'est arrêtée en ce temps-là et a accepté ;
2. si $x = 1^{\beta_i}$, N simule $M_i(1^{\alpha_i})$ **de manière déterministe**, et répond le contraire de M_i.

3. Comment choisir α_i et β_i tels que $L(N) \in \mathsf{NTIME}(g(n))$? **Indication.** Trouver i tel que $x \in I_i$ doit être *suffisamment rapide* et l'étape **(b)** également.
4. Supposons que $L(N) \in \mathsf{NTIME}(f(n))$, grâce à une MTND M. Montrer qu'il existe i tel que $M = M_i$ et tel qu'à l'étape **(a)**, M_i est toujours simulée jusqu'au bout.
5. Conclure.