Complexité structurelle – Partiel
Lundi 27 mars, 13h30 – 15h30

Exercice 1

On rappelle que $E = \mathrm{DTIME}(2^{O(n)})$.
Soit T unaire. Montrer que $E^T = E \Longleftrightarrow T \in \mathrm{P}$.

Exercice 2

Un langage A est dit *p-sélectif* s'il existe une fonction f calculable en temps polynomial (f est appelée la fonction de sélection), telle que pour tout $x, y \in \{0,1\}^*$, on ait :
- $f(x,y) = x$ ou $f(x,y) = y$;
- si $x \in A$ ou $y \in A$ alors $f(x,y) \in A$.

Question 1 – Soit A un langage p-sélectif avec fonction de sélection f. Montrer que pour tout n, il existe un ensemble $C \subseteq A \cap \{0,1\}^n$ tel que $|C| \leq n+1$ et

$$A \cap \{0,1\}^n = \{x; |x| = n \text{ et } \exists y \in C \text{ tel que } f(x,y) = x \text{ ou } f(y,x) = x\}.$$

Question 2 – Montrer qu'un langage p-sélectif est dans P/poly. (Ko, 1983)

Question 3 – Montrer que s'il existe un langage p-sélectif NP-difficile, alors P = NP.

Problème

Le but de ce problème est de montrer l'implication suivante[1] :

$$\mathrm{coNP} \subseteq \mathrm{NP}/1 \Longrightarrow \mathrm{PH} \subseteq \mathrm{DP}.$$

On rappelle que la classe DP est celle des langages qui sont l'intersection d'un langage NP et d'un langage coNP, ou de manière équivalente, qui sont la différence de deux langages NP. Par ailleurs, NP/1 est la classe des langages NP avec un conseil de longueur 1. L'ordre utilisé sur les mots booléens est l'ordre lexicographique. Enfin, PH désigne la hiérarchie polynomiale : $\mathrm{PH} = \cup_i \Sigma_i$, avec

$$\Sigma_0 = \mathrm{P}, \text{ et } \Sigma_{i+1} = \mathrm{NP}^{\Sigma_i}.$$

Question 1 – Montrer que $\Sigma_2 \subseteq \mathrm{DP} \Longrightarrow \mathrm{PH} \subseteq \mathrm{DP}$.

Pour montrer le résultat annoncé au début du problème, nous allons procéder en deux étapes : d'abord, nous allons montrer que $\mathrm{coNP} \subseteq \mathrm{NP}/1 \Rightarrow \Sigma_2 \subseteq \mathrm{P}^{\mathrm{NP}}$, puis que $\mathrm{coNP} \subseteq \mathrm{NP}/1 \Rightarrow \mathrm{P}^{\mathrm{NP}} \subseteq \mathrm{DP}$.
Question 2 – Justifier que ces deux étapes permettent de conclure.

Nous commençons par la première étape, à savoir, nous allons montrer l'implication $\mathrm{coNP} \subseteq \mathrm{NP}/1 \Rightarrow \Sigma_2 \subseteq \mathrm{P}^{\mathrm{NP}}$.

[1] La réciproque est vraie également mais nous ne nous en occuperons pas ici. Il s'agit d'un résultat de Buhrman, Chang et Fortnow, dans *One bit of advice*, 2002.

Question 3 – Par NP-complétude de SAT, on peut supposer que l'oracle dans P^{NP} est SAT. Montrer que l'on peut de plus supposer que la taille des requêtes à l'oracle SAT sur les entrées de taille n est toujours la même.

Question 4 – Montrer que si coNP $\subseteq$ NP/1, alors $\Sigma_2 \subseteq$ NP/1.

Question 5 – Montrer qu'il existe des langages Σ_2-complets de la forme

$$L = \{(x, y) \ / \ (\exists y')(\forall z)[y' \geq y \wedge (x, y', z) \in A]\},$$

où $A \in \mathrm{P}$ et où les quantifications portent sur des mots de taille polynomiale.

À partir de maintenant, L désigne un langage Σ_2-complet de la forme donnée à la question 5.

Question 6 – À la question 4, nous avons montré que $L \in$ NP/1 sous l'hypothèse coNP $\subseteq$ NP/1. Montrer que dans P^{NP}, et si l'on connaît le bit du conseil, sur l'entrée x on peut trouver, s'il existe, le y' maximum tel que $(\forall z)[(x, y', z) \in A]$.

Question 7 – Montrer que sous l'hypothèse coNP $\subseteq$ NP/1, on a $L \in \mathrm{P}^{\mathrm{NP}}$, et conclure la première étape.

On s'attache maintenant à la seconde étape, à savoir, nous allons montrer l'implication coNP $\subseteq$ NP/1 $\Rightarrow \mathrm{P}^{\mathrm{NP}} \subseteq$ DP. On admettra que le problème LexMaxSat suivant sur les formules booléennes ϕ est P^{NP}-complet :

LexMaxSat =
$\{\phi$ satisfaisable / le plus grand x tel que $\phi(x) = 1$ se termine par $1\}$.

Question 8 – Sous l'hypothèse coNP $\subseteq$ NP/1, construire une machine N non déterministe fonctionnant en temps polynomial, avec un bit de conseil, qui sur la formule booléenne ϕ retourne x_{max}, le plus grand x tel que $\phi(x) = 1$. Plus précisément, lorsqu'on lui fournit le bon conseil, N a au moins un chemin acceptant ssi ϕ est satisfaisable ; et à la fin de tous ses chemins acceptants, x_{max} est inscrit sur le ruban de sortie de N. Si en revanche on lui fournit le mauvais bit de conseil, il peut y avoir plusieurs chemins acceptant ayant des mots différents sur le ruban de sortie. On demande de plus que la machine vérifie à la fin que $\phi(x_{max})$ est vraie, où x_{max} est le mot sur le ruban de sortie à la fin du chemin en cours, et qu'elle rejette si ce n'est pas le cas. Ainsi, dans tous les cas notre machine ne peut avoir de chemin acceptant que si ϕ est satisfaisable (condition nécessaire mais pas suffisante si on donne le mauvais conseil).

S'il existe un chemin acceptant de N tel que le mot final sur le ruban de sortie est x, on dira que x est l'un des mots renvoyés. On note $N(\phi, 0)$ le calcul de N sur ϕ avec le conseil 0, et $N(\phi, 1)$ avec le conseil 1. On définit le langage A_1 suivant :

$$A_1 = \{\phi \text{ satisfaisable } / \ N(\phi, 0) \text{ ou } N(\phi, 1) \text{ renvoie un mot se terminant par } 1\}.$$

Le langage A_2 est celui reconnu par la machine non déterministe N_2 fonctionnant ainsi : sur l'entrée ϕ, N_2 simule $N(\phi, 0)$ et $N(\phi, 1)$. Si tous les mots renvoyés sont les mêmes, ou si l'une des deux simulations rejette, alors N_2 rejette ϕ. Sinon, il existe x_0 renvoyé par $N(\phi, 0)$, et x_1 par $N(\phi, 1)$, avec $x_0 \neq x_1$. Alors N_2 accepte ϕ si et seulement si $\min(x_0, x_1)$ se termine par 1 et $\max(x_0, x_1)$ se termine par 0.

Question 9 – Montrer que A_1 et A_2 sont dans NP. Montrer que LexMaxSat $= A_1 \setminus A_2$ et conclure.