Complexité - TD 05

Benjamin Bordais

17 Décembre 2020

Exercise 1 Espace Poly-Logarithmique

On rappelle le théorème de hiérarchie en espace :

Théorème 1 *Pour deux fonctions constructibles $f, g \geq \log$ telles que $f(n) = o(g(n))$, on a* $\mathsf{DSPACE}(f) \subsetneq \mathsf{DSPACE}(g)$.

On définit à présent la classe de complexité :

$$\mathsf{PolyLog} = \bigcup_{k>0} \mathsf{SPACE}(\log^k(n))$$

.

1. Montrer que $\mathsf{PolyLog}$ n'a pas de problème complet pour des réductions en espace logarithmique. Que peut-on déduire quant à la comparaison entre les classes P et $\mathsf{PolyLog}$?
2. On rappelle que $\mathsf{PSPACE} = \bigcup_{k>0} \mathsf{SPACE}(n^k)$. Est-ce que PSPACE a un problème complet pour des réductions en espace logarithmique ? Pourquoi est-ce que la preuve de la question précédente ne s'applique pas à PSPACE ?

Exercise 2 Fonction de choix

Un langage L appartient à P-*choice*, écrit $L \in \mathsf{P}_c$, s'il existe une fonction $f : \Sigma^* \times \Sigma^* \to \Sigma^*$, calculable en temps polynomial telle que pour tout $x, y \in \Sigma^*$:
- $f(x, y) \in \{x, y\}$,
- si $x \in L$ ou $y \in L$, alors $f(x, y) \in L$.

Dans ce cas, f est appelé la *fonction de choix* pour L.

1. Montrer que $\mathsf{P} \subseteq \mathsf{P}_c$.
2. Monter que P_c est clos par complémentaire.
3. Monter que s'il existe un problème NP-dur dans P_c alors $\mathsf{P} = \mathsf{NP}$.

Exercise 3 Clôture par morphisme

Étant donné un alphabet fini Σ, une fonction $f : \Sigma^* \to \Sigma^*$ est un morphisme si $f(\Sigma) \subseteq \Sigma$ et pour tout $a = a_1 \cdots a_n \in \Sigma^*$, $f(a) = f(a_1) \cdots f(a_n)$ (f est entièrement déterminée par les valeurs prises sur Σ).

Montrer que $\mathsf{P} = \mathsf{NP}$ si et seulement si P est clos par morphisme (i.e. si $L \in \mathsf{P}$, alors $f(L) \in \mathsf{P}$ pour tout morphisme f).

Exercise 4 Théorème de Ladner

On souhaite montrer que si $\mathsf{P} \neq \mathsf{NP}$ alors il existe $L \in \mathsf{NP} \backslash \mathsf{P}$ tel que L ne soit pas NP-complet.

On considère un codage des machines de Turing dans les entiers tel que à toute machine de Turing correspondent un nombre infini d'entiers, on notera M_i la machine de Turing codé par l'entier i. On définie récursivement la fonction $H : \mathbb{N} \mapsto \mathbb{N}$ et l'ensemble SAT_H comme suit :

$$\mathsf{SAT}_H = \{\psi.0.1^{n^{H(n)}} \mid \psi \in \mathsf{SAT} \wedge n = |\psi|\}$$

$$H(n) = \min \begin{cases} \log(\log(n)) \\ \displaystyle\min_{i < \log(\log(n))} \{i \mid M_i \text{ decide } \mathsf{SAT}_H \text{ en temps } i \cdot |x|^i \text{ pour tout } |x| < \log(\log(n))\} \end{cases}$$

La définition ci-dessus n'est pas valide en 0 et en 1, on prendra donc $H(0) = H(1) = 0$. On va maintenant montrer les résultats suivants :

1. Montrez que H et SAT_H sont bien définies, que H est croissante et qu'elle se calcule en temps polynomial.
2. Montrez que si SAT_H est dans P alors H est bornée.
3. Réciproquement montrez que si H est bornée alors SAT_H est dans P.

Établisser à présent les propriétés suivantes :

(i) $\mathsf{SAT}_H \in \mathsf{NP}$;

(ii) $\mathsf{SAT}_H \notin \mathsf{P}$;

(iii) SAT_H n'est pas NP-complet.