Partiel. Complexité algorithmique. 5 novembre 2010.

Durée : 2h.
Notes de cours et documents non autorisés.

Le sujet comprend 6 exercices indépendants. Les exercices 1 et 2 concernent des questions de cours. Vous pouvez écrire en Français ou en anglais / You can write in either French or English. Penser à bien justifier les réponses ; la notation tiendra compte de la rédaction.

Exercice 1.
Rappeler la définition des classes P, NP et coNP, ainsi que des langages NP-complets et coNP-complets. Donner un langage NP-complet et un langage coNP-complet "naturels".

Exercice 2.
Énoncer le théorème de hiérarchie en temps déterministe et le démontrer.

Exercice 3.
La *réductibilité en temps non déterministe polynomial* est définie de la manière suivante :
$L_1 \leq_{\mathsf{NP}} L_2$ ssi il existe une machine non déterministe de temps polynomial $\mathcal{M}$ avec ruban de sortie, telle que :
pour tout x de $\{0,1\}^*$, $x \in L_1$ ssi il existe une exécution de $\mathcal{M}$ sur l'entrée x qui produit une sortie y avec $y \in L_2$.
Montrer que la relation $\leq_{\mathsf{NP}}$ est transitive. Montrer que si $L_1 \leq_{\mathsf{NP}} L_2$ et $L_2 \in \mathsf{NP}$, alors $L_1 \in \mathsf{NP}$.

Exercice 4.
On définit la classe P-Sel de la manière suivante :
un langage $\mathcal{L}$ est dans P-Sel (pour P sélectif) s'il existe $f : \{0;1\}^* \times \{0;1\}^* \to \{0;1\}^*$ calculable en temps polynomial telle que pour tout (x, y) :

$$\begin{cases} f(x,y) = x \text{ ou } f(x,y) = y, \\ \text{si } x \in \mathcal{L} \text{ ou } y \in \mathcal{L}, \text{ alors } f(x,y) \in \mathcal{L} \end{cases}$$

On dit alors que f est une *fonction de sélection* pour $\mathcal{L}$.

1. Montrer que $\mathsf{P} \subseteq$ P-Sel.
2. Montrer que si $\mathcal{L} \in$ P-Sel alors son complémentaire l'est aussi.
3. Montrer que $\mathcal{L} \in$ P-Sel ss'il existe L' dans P tel que : $\mathcal{L} \times \overline{\mathcal{L}} \subseteq L'$ et $\overline{\mathcal{L}} \times \mathcal{L} \subseteq \overline{L'}$, où $A \times B = \{\langle x, y\rangle \ : \ x \in A \text{ et } y \in B\}$, et $\overline{\mathcal{L}}$ désigne le complémentaire de $\mathcal{L}$.
4. Montrer que s'il existe un langage dans P-Sel qui est NP-dur, alors $\mathsf{P} = \mathsf{NP}$.

Exercice 5.

1. Soit $c > 0$. Montrer que $\mathsf{SPACE}(n^c) \subseteq \mathsf{NP}$ implique $\mathsf{NP} = \mathsf{PSPACE}$.
   Indication : on pourra utiliser une technique de padding.
2. En déduire que pour tout $c > 0$, $\mathsf{SPACE}(n^c) \neq \mathsf{NP}$.
3. De la même manière, montrer que $\mathsf{DTIME}(2^{cn}) \neq \mathsf{NP}$ pour tout $c > 0$.

Exercice 6.
On considère le langage suivant :

$$\begin{aligned} IN\_PLACE\_DIVERGE \quad = \quad & \{\alpha \in \{0,1\}^* \ / \ \text{sur l'entrée vide, la machine } \mathcal{M}_\alpha \text{ diverge et} \\ & \text{n'utilise pas plus de } |\alpha| \text{ cases}\}. \end{aligned}$$

où $\mathcal{M}_\alpha$ est la machine déterministe définie par le mot α, et on dit qu'une machine $\mathcal{M}$ *diverge sur une entrée x* si son exécution sur cette entrée ne termine pas.

1. Montrer que ce langage appartient à PSPACE.
2. Montrer qu'il est PSPACE-complet.