Examen de calculabilité et complexité

M1 informatique – 3h

Le 11 janvier 2011

La qualité de la rédaction et de la présentation sera prise en compte dans l'évaluation. Le barème est donné à titre indicatif.

Exercice 1 – Vrai ou faux ? *(4 points)*

Les affirmations suivantes sont-elles vraies ou fausses ? Ne justifiez pas vos réponses. On comptera $+\frac{1}{3}$ point par bonne réponse, $-\frac{1}{3}$ point par mauvaise réponse, et 0 pour une absence de réponse. Si le total est négatif, on donnera 0 à l'exercice.

1. Pour toute fonction $f : \mathbb{N} \to \mathbb{N}$ vérifiant $f(n) \geq n$ et f croissante, il existe un langage $A \in \mathsf{DTIME}(2^{f(n)^2})$ qui ne soit pas dans $\mathsf{NTIME}(f(n))$.
2. On sait simuler une machine de Turing non déterminisite N fonctionnant en temps $t(n)$ par une machine déterministe M fonctionnant en espace $t(n)^2$.
3. Il existe un langage de EXP qui n'est pas dans NL.
4. SAT possède un algorithme fonctionnant en espace polynomial.
5. Tout langage A tel que $A \neq \emptyset$ et $A \neq \Sigma^\star$ est P-difficile pour les réductions $\leq_m^p$.
6. Si SAT $\leq_m^p A$ alors A est NP-complet.
7. Si $A \leq_m^p$ SAT alors $A \in \mathsf{NP}$.
8. Si $A \leq_m^p \{\epsilon\}$ alors $A \in \mathsf{P}$.
9. Si $A \leq_m \emptyset$ alors $A = \emptyset$.
10. Tout langage récursivement énumérable est indécidable.
11. Si A et B sont récursivement énumérables, alors $AB = \{xy \ : \ x \in A \wedge y \in B\}$ est récursivement énumérable.
12. Le langage $A = \{M \ : \ \forall x, M(0x)$ accepte et $M(1x)$ rejette$\}$ est décidable.

Exercice 2 *(4 points)*

1. Montrer que le langage $A = \{\langle M \rangle \ : \ \forall y \in \Sigma^\star, M(y)$ s'arrête en $\leq (|y|+1)^2$ étapes$\}$, où M est une machine de Turing déterminisite, est indécidable.
   Indication : dans l'optique de construire une réduction du problème de l'arrêt, si M est une machine de Turing et x son entrée, on pourra considérer une machine M' qui sur l'entrée y simule $|y|$ étapes de $M(x)$.
2. Montrer que pour tout $k \in \mathbb{N}$, le langage $A_k = \{\langle M \rangle \ : \ \forall y \in \Sigma^\star, M(y)$ s'arrête en $\leq k$ étapes$\}$ est décidable.

Exercice 3 *(3 points)*

Soit G un graphe orienté. Montrer que le problème de décider si G possède un circuit (pas nécessairement simple) de taille impaire est dans NL.

Exercice 4 *(4 points)*

Soit Set Packing (SP) le problème suivant :

- Entrée : un entier n en unaire, m ensembles $S_1, \ldots, S_m \subseteq \{1, \ldots, n\}$, un entier $k \leq m$.
- Question : existe-t-il k ensembles $S_{i_1}, \ldots, S_{i_k}$ deux à deux disjoints ?

et SP2 la variante suivante (où l'on considère cette fois des couples d'entiers) :

- Entrée : un entier n en unaire, m ensembles $S_1, \ldots, S_m \subseteq \{1, \ldots, n\}^2$, un entier $k \leq m$.
- Question : existe-t-il k ensembles $S_{i_1}, \ldots, S_{i_k}$ deux à deux disjoints ?

1. Montrer que SP2 se réduit à SP (pour les réductions $\leq_m^p$).
2. On rappelle que Ensemble Indépendant (EI) est le problème NP-complet suivant :
   - Entrée : un graphe non orienté G et un entier k.
   - Question : existe-t-il k sommets sans aucune arête entre eux (ensemble indépendant de taille k) ?

   Montrer que EI se réduit à SP2 (pour les réductions $\leq_m^p$).
3. Montrer que SP est NP-complet.

Exercice 5 *(3 points)*

1. Montrer que si $\mathsf{DSPACE}(n) \subseteq \mathsf{NP}$, alors $\mathsf{PSPACE} \subseteq \mathsf{NP}$.

   Indication : on pourra utiliser la technique de *padding* (augmentation artificielle de la taille de l'entrée) vue en cours.
2. En déduire que $\mathsf{DSPACE}(n) \neq \mathsf{NP}$.

   Indication : on pourra utiliser le théorème de hiérarchie en espace.

Exercice 6 *(4 points)*

Un langage A est dit *p-sélectif* ($A \in \mathsf{P}$-sel) s'il existe une fonction $f : \Sigma^\star \times \Sigma^\star \to \Sigma^\star$ calculable en temps polynomial telle que $\forall x, y \in \Sigma^\star$:

- $f(x, y) \in \{x, y\}$, et
- si $x \in A$ ou $y \in A$ alors $f(x, y) \in A$.

1. Montrer que $\mathsf{P} \subseteq \mathsf{P}$-sel.
2. Montrer que P-sel est clos par complémentaire.
3. Montrer que s'il existe un langage A NP-difficile dans P-sel, alors $\mathsf{P} = \mathsf{NP}$.

   Indication : en s'aidant de A, on pourra donner un algorithme polynomial pour SAT qui utilise la propriété $\phi(x_1, \ldots, x_n) \in \text{SAT} \iff [\phi(0, x_2, \ldots, x_n) \in \text{SAT}$ ou $\phi(1, x_2 \ldots, x_n) \in \text{SAT}]$.