# Partiel – Vendredi 23 novembre 2007, 9h-12h

Les notes de cours et de TD sont interdites. Il sera tenu compte du soin apporté à la rédaction.

**Définition**
On considère des machines de Turing à un ruban d'entrée en lecture seule et avec un ou plusieurs rubans de travail. Une fonction $f : \mathbb{N} \to \mathbb{N}$ est dite *constructible en temps* (respectivement en espace) s'il existe $k$ et une machine à $k$ rubans qui, sur toute entrée de taille $n$, s'arrête en temps exactement $f(n)$ (resp. utilise exactement un espace $f(n)$ sur ses rubans de travail).

**Exercice 1.**

**1.** Montrer que la fonction $f(n) = \lfloor \log_2 n \rfloor + 1$ est constructible en espace.

**2.** Montrer que si les fonctions $f$ et $g$ sont constructibles en temps, alors il en est de même de leur composition $f \circ g$.

**3.** Quelles sont les fonctions $f$ constructibles en temps telles qu'il existe un entier $n$ vérifiant $f(n) < n$ ?

**Définition**
Soient $A$ et $B$ deux langages. On dit que $A$ se réduit à $B$ par réduction Turing, et on note $A \leq_T B$, s'il existe une machine de Turing à oracle $M$ fonctionnant en temps polyomial telle que $A$ soit le langage reconnu par la machine $M$ munie de l'oracle $B$ (en symboles, $A \in \mathsf{P}^B$).

**Exercice 2.**
On suppose que $A$ se réduit à $B$ pour la réduction habituelle (en symboles, $A \leq B$). Montrer que $A \leq_T B$.

**Définitions**
L'ensemble des *termes* sur les variables $x_1, \ldots, x_n$ (aussi appelés formules booléennes) est défini par induction :

1. les constantes 0 et 1, et les variables $x_i$ sont des termes ;
2. si $T_1$ et $T_2$ sont des termes, alors $\neg T_1$, $T_1 \wedge T_2$ et $T_1 \vee T_2$ aussi.

Autrement dit, un terme est un circuit booléen où toutes les portes ont pour degré sortant 1 (il n'y a pas de porte de duplication). Toutefois une même variable peut apparaître plusieurs fois en entrée.
La taille et la profondeur d'un terme sont également définies par induction :

1. les constantes 0 et 1, et les variables $x_i$ sont des termes de taille 1 et de profondeur 0 ;
2. si $T_1$ et $T_2$ sont des termes de tailles respectives $t_1$ et $t_2$ et de profondeurs respectives $p_1$ et $p_2$, alors :
   – $\neg T_1$ est de taille $1 + t_1$ et de profondeur $1 + p_1$ ;
   – $T_1 \wedge T_2$ et $T_1 \vee T_2$ sont de taille $1 + t_1 + t_2$ et de profondeur $1 + \max\{p_1, p_2\}$.

**Exercice 3.** *Théorème de Spira*
Nous allons montrer que tout terme de taille $t$ est équivalent à un terme de profondeur au plus $4 \log_2 t$.

**1.** Montrer le résultat pour $t = 1$.

**2.** On raisonne par récurrence. Soit $T$ un terme de taille $t \geq 2$. Considérez un sous-terme $S$ de taille minimale parmi les sous-termes de taille $> t/2$. On définit $T_0$ (respectivement $T_1$) le terme où l'on a remplacé $S$ par 0 (resp. 1) dans $T$.
Conclure en utilisant une technique de parallélisation du calcul.

---

**Exercice 4.** *Théorème de Ladner*

Nous allons montrer le résultat suivant : si $\mathsf{P} \neq \mathsf{NP}$ alors il existe un langage $\mathcal{L} \in \mathsf{NP}$ qui n'est ni $\mathsf{NP}$-complet, ni décidable en temps polynomial.
On supposera donc dans tout l'exercice que $\mathsf{P} \neq \mathsf{NP}$.

**1.** Montrer qu'il existe une énumération effective $M_i$ des machines de Turing fonctionnant en temps polynomial.

On fixe une telle énumération $M_i$. Note : pour une même machine $M_i$, on s'intéressera suivant les cas au mot $M_i(x)$ écrit sur le ruban de sortie à la fin du calcul, ou simplement à son état de sortie (acceptation ou rejet).

**2.** On considère la fonction suivante, définie par récurrence :
- $f(0) = 1$.
- si $k = 2i + 1$ est impair : on calcule d'abord $f(k-1)$. On initialise alors une horloge qui décomptera le nombre de pas de calcul par la suite. Cette horloge est initialisée à la valeur $f(k-1)$. On énumère ensuite un à un les mots $x$ tels que $|x| \geq f(k-1)$. On s'arrête dès qu'on en trouve un vérifiant l'une des deux conditions suivantes :
  i) $M_i$ rejette $x$ et $x \in \mathsf{SAT}$
  ii) $M_i$ accepte $x$ et $x \notin \mathsf{SAT}$.
  On pose $f(k)$ la valeur de l'horloge à ce moment-là.
- si $k = 2i$ est pair : on pose
$$T = \{x \in \Sigma^\star,\ \text{tel que } \exists j < i, f(2j) \leq |x| < f(2j+1)\}.$$
  On calcule d'abord $f(k-1)$. On initialise alors une horloge qui décomptera le nombre de pas de calcul par la suite. Cette horloge est initialisée à la valeur $f(k-1)$. On énumère alors un à un les mots $x$ tels que $|x| \geq f(k-1)$. On s'arrête dès qu'on en trouve un vérifiant l'une des deux conditions suivantes :
  iii) $M_i(x) \notin \mathsf{SAT} \cap T$ et $x \in \mathsf{SAT}$
  iv) $M_i(x) \in \mathsf{SAT} \cap T$ et $x \notin \mathsf{SAT}$.
  On pose $f(k)$ la valeur de l'horloge à ce moment-là.

Montrez que la suite $(f(k))_{k\in\mathbb{N}}$ est infinie, strictement croissante et calculable.

**3.** Montrer qu'il existe une machine déterministe qui sur l'entrée $k$ calcule $f(0), f(1), \ldots, f(k)$ en temps $O(f(k))$.

**4.** On pose $S = \{x \in \Sigma^\star,\ \text{tel que } \exists i, f(2i) \leq |x| < f(2i+1)\}$. Montrer que $S \in \mathsf{P}$.

**5.** On pose $A = \mathsf{SAT} \cap S$. Montrer que $A$ est dans $\mathsf{NP}$.

**6.** Montrer que $A$ n'est pas dans $\mathsf{P}$.

**7.** Montrer que $A$ n'est pas $\mathsf{NP}$-complet.

---

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

**Exercice 5.** *Théorème de Mahaney*

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

**2.** Nous allons montrer que s'il existe un langage $\mathcal{L}$ creux et NP-dur, alors $\mathsf{P} = \mathsf{NP}$. Soit donc un tel langage $\mathcal{L}$ 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 $G(A) = \{\langle x,\, w\rangle, \text{ tels que } \exists y \in \Sigma^{p(|x|)}, y \geq w \text{ et } \langle x,\, y\rangle \in A\}$.
Montrer que $G(A)$ est dans NP.

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