M1, 2005–2006

Complexité structurelle – TD 5
Classes probabilistes, 1

Exercice 1
Montrer que RP $\subseteq$ NP.

Exercice 2 On va montrer que RP $\subseteq$ P/poly.

1. Soit $A \in$ RP. Montrer qu'il existe un polynôme $p(n)$ et un langage $B \in$ P tels que
$$x \in A \Longleftrightarrow |\{y \in \{0,1\}^{p(|x|)}; (x,y) \in B\}| \leq 2^{p(|x|)-|x|-1}, \text{ et}$$
$$x \notin A \Longleftrightarrow \{y \in \{0,1\}^{p(|x|)}; (x,y) \notin B\} = \emptyset.$$
2. Considérer le tableau $t[x,y]$, avec $x \in \{0,1\}^n$ et $y \in \{0,1\}^{p(n)}$, tel que $t[x,y] = 1$ ssi ($x \in A$ et $(x,y) \in B$) (i.e. l'algo se trompe). Majorer le nombre de uns dans ce tableau.
3. Conclure.

On rappelle que E = DTIME($2^{O(n)}$) et EXPSPACE = DSPACE($2^{O(n)}$).
Exercice 3

1. On veut montrer que E $\neq$ EXPSPACE $\Longleftrightarrow$ PSPACE $\cap$ P/poly $\neq$ P.
   - 1.a. Montrer l'implication $\Rightarrow$. Les étapes suivantes concernent la réciproque.
   - 1.b. Soit $L \in$ P/poly : il est reconnu par une famille de circuits booléens (C_n) de taille polynomiale. Encoder (C_n) dans un langage unaire T, tel que $L \in \mathrm{P}^T$. On demande de plus qu'il existe une machine polynomiale N à oracle décidant L avec l'oracle T, qui sur l'entrée x, ne demande à son oracle que des mots de la forme 1^k avec $q(|x|) \leq k < q(|x|+1)$, pour un certain polynôme $q(n)$.
   - 1.c. Supposons de plus que $L \in$ PSPACE, reconnu par une machine M en espace polynomial. Se servir de M pour construire à partir de T un nouveau langage unaire $T' \in$ PSPACE tel que $L(N,T) = L(N,T')$. Ainsi, $L \in \mathrm{P}^{T'}$.
   - 1.d. Supposons de plus que $L \notin$ P. Construire à partir de T' un langage dans EXPSPACE $\setminus$ E. Conclure.
2. Montrer que P $\neq$ RP $\Longrightarrow$ E $\neq$ EXPSPACE.

Si on ordonne les mots sur $\{0,1\}$ par ordre lexicographique, on peut parler du i-ème mot binaire, que l'on notera x_i. À un langage L on associe la suite $u \in \{0,1\}^{\mathbf{N}}$ définie par
$$u_i = \begin{cases} 1 & \text{si } x_i \in L \\ 0 & \text{sinon.} \end{cases}$$
Enfin, sur l'ensemble des suites de $\{0,1\}$, on met la mesure de probabilité uniforme : elle est induite par la mesure d'équiprobabilité sur $\{0,1\}$, ou en d'autres

termes, pour tout i, la probabilité que $u_i = 1$ est $1/2$. On peut ainsi parler d'oracle aléatoire.
Exercice 4 On veut montrer que $\mathrm{P}^A = \mathrm{RP}^A$ avec probabilité 1, c'est-à-dire que pour presque tout oracle A, $\mathrm{P}^A = \mathrm{RP}^A$.

1. D'abord, quelques définitions. Le but du jeu est donc de montrer que pour tout $\delta > 0$, la probabilité que $\mathrm{P}^A \neq \mathrm{RP}^A$ est $< \delta$. On considère pour cela une énumération (M_i) des machines de Turing probabilistes fonctionnant en temps polynomial. On dira que la machine M_i est de type RP avec l'oracle A si pour tout x, dans le calcul de M_i^A sur x, soit la proportion de chemins acceptant est d'au moins $3/4$, soit tous les chemins sont rejetant. On notera $p_i(n)$ le temps de calcul de la machine M_i sur des entrées de taille n.
2. L'idée est d'utiliser l'oracle A comme source de bits pseudo-aléatoires. Pour obtenir des bits « aléatoires », on interrogera A sur des mots que la machine M_i ne demandera pas. Par ailleurs, avec ces bits on souhaite simuler M_i plusieurs fois de façon à réduire suffisamment la probabilité d'erreur. Proposer une machine déterministe D_i avec oracle, fonctionnant en temps polynomial, qui calcule des bits pseudo-aléatoires grâce à son oracle, et qui s'en sert pour simuler M_i (avec le même oracle) un nombre de fois k à déterminer plus tard.
3. Si i est un entier et x un mot binaire, on note $E_{i,x}$ l'ensemble
$$E_{i,x} = \{A; M_i \text{ avec oracle } A \text{ est de type RP, et } D_i^A(x) \neq M_i^A(x)\}.$$
Majorer $\mathrm{Prob}(A \in E_{i,x})$ en fonction du nombre k de simulations de M_i dans la question précédente.
4. Supposons que $\mathrm{P}^A \neq \mathrm{RP}^A$ pour un certain ensemble A. A est-il dans un des $E_{i,x}$ avec $|x| > i$?
5. Majorer la probabilité que $\mathrm{P}^A \neq \mathrm{RP}^A$ en fonction de k, choisir k correctement et conclure.

La classe BPP *(bounded-error probabilistic polynomial-time)* autorise des erreurs des deux côtés : un langage A est dans BPP si et seulement s'il existe un langage $B \in \mathrm{P}$ et un polynôme $p(n)$ tel que
– si $x \in A$ alors $\mathrm{Prob}_{y\in\{0,1\}^{p(|x|)}}((x, y) \in B) \geq 3/4$;
– si $x \notin A$ alors $\mathrm{Prob}_{y\in\{0,1\}^{p(|x|)}}((x, y) \in B) \leq 1/4$.
Exercice 5
Montrer que $\mathrm{BPP}^{\mathrm{BPP}} = \mathrm{BPP}$. Réfléchir à la question « $\mathrm{RP}^{\mathrm{RP}} = \mathrm{RP}$? ».

Exercice 6
Reprendre les exercices 2, 3 et 4 avec BPP à la place de RP. Pour le pendant de l'exercice 1, réfléchir à la question « $\mathrm{BPP} \subseteq \mathrm{NP}$? ».