TD12 - Randomized Computation

Exercice 1. *Tirage équitable*

1. Soit une pièce (source aléatoire), avec $Pr[Face] = \rho$. Montrer que si le $i$ème bit de ρ est calculable en temps poly(i), alors la pièce peut être simulée en "expected time" $O(1)$.

2. Donner un réel ρ, tel que si l'on se donne une pièce qui renvoie 'face' avec probabilité ρ, on peut construire un machine de Turing qui décide un langage indécidable en temps polynomial.

3. Inversement, montrer qu'on peut simuler une pièce équitable (1/2-pièce) avec une ρ-pièce en expected time $O(1/\rho.(1-\rho))$.

4. Montrer qu'on peut simuler un tirage aléatoire dans $[1..N]$ avec une pièce ; i.e., Pour tout N et $\delta > 0$, il existe un algorithme probabiliste A, polynomial en $\log N \log(1/\delta)$, qui renvoie un élément de $1, ..., N, ?$ tel que
– Lorsqu'il ne renvoie pas ?, la sortie de A est uniformément distribuée dans $[1..N]$
– La probabilité que A renvoie ? est au plus δ.

Exercice 2. *Marches aléatoires*

1. Donner une définition de "borné en espace" pour les MT probabilistes. Définir BPL et RL

✎ La réduction d'erreur pour les machines probabilistes peut être effectuée en espace logarithmique. (du coup la constante importe peu...)

On définit le problème $UPATH$ sur un graphe non-orienté : Etant donnés un graphe G à n arêtes et deux noeuds s et t, déterminer si s et t sont connectés dans G.

3. Montrer que $UPATH \in RL$.

4. Montrer qu'on ne peut pas utiliser les mêmes arguments pour des graphes orientés : Donnez un graphe à n arêtes, et deux noeuds s, t tels qu'il existe un chemin de s à t mais que le expected time pour aller de s à t soit en $\Omega(2^n)$.

Exercice 3. *Réductions randomisées*

$B \leq_r C$ ssi il existe une PTM M telle que pour tout x, $Pr[C(M(x)) = B(x)] = 2/3$.

1. Cette relation est-elle transitive ?

2. Montrer que si $C \in BPP$ et $B \leq_r C$, alors $B \in BPP$

✎ On peut du coup définir la NP-complétude à l'aide de cette définition...

Exercice 4. *Reduction dans RP*

1. Montrer que pour tout langage L de RP avec PTM M telle que $Pr[M(x) = 1] \geq n^{-c}$ pour $x \in L$ (0 sinon), on peut construire une machine M' qui reconnait L avec $Pr[M(x) = 1] \geq 1 - 2^{-n^d}$ pour $x \in L$ (0 sinon).