

Classes Probabilistes & Non-Uniformes PC 7.

Correction

1 Simulation de lois aléatoires

1. Montrer que l'on peut simuler un tirage aléatoire dans $\{1, 2, \dots, N, ?\}$ avec une pièce : i.e. pour tout N et $\delta > 0$, il existe un algorithme probabiliste A , qui fonctionne en temps polynomiale en $\log N \log(1/\delta)$, qui renvoie un élément de $\{1, 2, \dots, N, ?\}$ tel que
 - lorsqu'il ne renvoie pas ?, la sortie de A est uniformément distribuée dans $\{1, 2, \dots, N\}$
 - la probabilité que A renvoie ? est au plus δ .

Correction: *Supposons que $N - 1$ s'écrive en binaire sur k bits. On tire k bits uniformément et indépendamment pour obtenir un nombre n en binaire $a_1 a_2 \dots a_k$. Si $n+1$ n'est pas entre 1 et N , on renvoie ?. Si l'on fait ainsi, on a une probabilité p de renvoyer ?. En itérant k fois l'opération tant qu'on renvoie ?, on rend cette probabilité $\leq p^k$. En prenant $k = \mathcal{O}(\log(1/\delta))$, puisque p est une constante, cette probabilité sera inférieure à δ .*

2. Soit X une variable aléatoire de fonction de répartition F strictement croissante. Rappel : la fonction F est définie par $F(x) = \Pr(X \leq x)$ strictement croissante. Montrer que $F(X)$ se comporte comme une loi uniforme sur $[0, 1]$.

Correction: *Posons $u = F(x)$, et donc $x = F^{-1}(u)$. Par définition, $F(x) = \Pr(X \leq x)$, et donc $F(F^{-1}(u)) = \Pr(X \leq F^{-1}(u))$. Or $F(F^{-1}(u)) = u$, et $\Pr(X \leq F^{-1}(u)) = \Pr(F(X) \leq u)$ puisque F est strictement croissante. On a donc $u = \Pr(F(X) \leq u)$, et on reconnaît la fonction de répartition de la loi uniforme.*

3. En déduire un moyen de simuler une loi lorsqu'on connaît F^{-1} : par exemple, une loi exponentielle.

Correction: *Il suffit de tirer U selon une loi uniforme sur $[0, 1]$, puis calculer $X = F^{-1}(U)$.*

Par exemple pour la loi exponentielle, $F(x) = 1 - e^{-\lambda x}$, et donc $F^{-1}(u) = -\frac{1}{\lambda} \ln(1 - u)$. On pourrait poser $X = -\ln(1 - U)/\lambda$, mais si l'on remarque

que si U suit une loi uniforme sur $[0, 1]$, $1 - U$ aussi, en pratique il suffit de poser $X = -\frac{\ln U}{\lambda}$ où U est une loi uniforme sur $[0, 1]$.

2 Problèmes complets pour PP

1. Montrer que PP est close par réduction polynomiale.

Correction: *Trivial.*

2. Le problème de décision MAJ consiste, étant donnée une formule booléenne à décider si elle est satisfaite par plus de la moitié des affectations de variables. Prouver que MAJ \in PP.

Correction: *On tire au hasard la valeur des variables de la formule, et on calcule la valeur de la formule. On accepte si et seulement si la formule est vraie. Cela donne un algorithme polynomial qui accepte avec probabilité $> 1/2$ si et seulement si la formule est dans MAJ.*

3. Le problème de décision #SAT consiste, étant donnée une formule booléenne F , et un entier i , à déterminer si F est satisfaite par plus de i affectations de variables. Montrer que #SAT est PP-difficile.
4. En déduire que MAJ et #SAT sont PP-complets.
5. Existe-t'il des problèmes BPP-complets ?

3 Classe \mathbb{P}

On rappelle que \mathbb{P} correspond aux langages reconnus par une famille de circuits booléens de taille polynomiale.

1. Montrer que \mathbb{P} contient des langages indécidables.

Correction: *Considérons $L \subset \mathbf{1}^*$ un langage indécidable. Par exemple, en considérant $L = \{1^{<A,w>} \mid \text{l'algorithme } A \text{ termine sur l'entrée } w\}$. Pour chaque n , il y a zéro ou un mot de longueur n . On peut donc construire un circuit (réduit à la porte 0 ou 1) qui reconnaît le langage restreint au mots de longueur n . Cela donne bien une famille de taille polynomiale qui reconnaît le langage L .*

2. Un langage L est dit *creux* lorsqu'il existe un polynôme p tel que, pour tout n , il y a moins de $p(n)$ mots dans L de longueur n .
Montrer que tout langage creux est dans \mathbb{P} .

4 $\mathbb{P} = \text{P}/poly$

Soit \mathcal{C} une classe de problèmes sur l'alphabet fini M . Soit *poly* la classe des fonctions f de \mathbb{N} dans M^* telle que pour un certain polynôme p , $|f(n)| \leq p(n)$.

La classe $\mathcal{C}/poly$ est la classe des problèmes B tels qu'il existe un problème $A \in \mathcal{C}$ et une fonction $f \in poly$ telle que $x \in B$ si et seulement si $\langle x, f(|x|) \rangle \in A$.

1. Prouver que $P/poly \subset \mathbb{P}$.
2. Prouver que $\mathbb{P} \subset P/poly$.
3. En déduire que

$$\mathbb{P} = P/poly.$$