COMPLEX - TD 10
décembre 2023

Énoncé

Exercice 1 : $\mathcal{P} \subseteq \mathcal{ZPP}$

1.a] Montrer que $\mathcal{P} \subseteq \mathcal{RP}$ et que $\mathcal{P} \subseteq co-\mathcal{RP}$.

1.b] En déduire que $\mathcal{P} \subseteq \mathcal{ZPP}$.

Exercice 2 : Amplification

Soit ε un nombre réel avec $0 < \varepsilon < 1/2$. Un langage L est dans $\mathcal{BPP}(\varepsilon)$ s'il existe une machine de Turing polynomiale M probabiliste telle que :

— si $x \in L$, alors $\Pr(M \text{ accepte } x) \geq \frac{1}{2} + \varepsilon$,

— si $x \notin L$, alors $\Pr(M \text{ accepte } x) < \frac{1}{2} - \varepsilon$.

En cours, nous avons défini $\mathcal{BPP}$ comme $\mathcal{BPP}(\varepsilon)$ avec $\varepsilon = 1/6$. Nous allons notamment voir dans cet exercice que le choix de ε constant ne modifie pas la définition de $\mathcal{BPP}$.

On considère une machine de Turing probabiliste M' qui sur $x \in \{0,1\}^*$ simule $n > 0$ fois M. Ainsi, M' accepte l'entrée x si M accepte $\geq n/2$ fois l'entrée x. Pour étudier M', nous allons utiliser une *inégalité de Chernoff*.

Théorème. *Soient $X_1, \ldots, X_n$ des variables aléatoires binaires indépendantes, $X = \sum_{i=1}^{n} X_i$ et $\mu = \mathbb{E}(X)$. Alors, pour tout $\delta, 0 < \delta \leq 1$: $\Pr\big(X < (1-\delta)\mu\big) < e^{-\mu \cdot \delta^2/2}$.*

2.a] Soit X_i la variable aléatoire binaire correspondant au résultat de la $i$ème exécution de la machine M par la machine M'. Estimer $\mathbb{E}(X_i)$ (suivant si $x \in L$ ou $x \notin L$, on donnera une minoration ou une majoration de $\mathbb{E}(X_i)$).

2.b] Soit $X = \sum_{i=1}^{n} X_i$. Estimer $\mathbb{E}(X)$.

2.c] Soit p_e la probabilité d'erreur de la machine M' sur $x \in L$. Montrer que :

$$p_e = \Pr\big(X < n/2\big) \leq \Pr\big(X < (1-\delta)\mu\big), \text{ avec } \delta = \frac{2\varepsilon}{1+2\varepsilon}.$$

2.d] En déduire une majoration de p_e.

2.e] Montrer que $\mu \cdot \delta^2/2 > \frac{\varepsilon^2 \cdot n}{2}$ et en déduire que $p_e < e^{-\frac{\varepsilon^2 \cdot n}{2}}$.

2.f] Soit ε une constante. Comment choisir n et pour avoir une probabilité d'erreur $< 2^{-p}$, avec p un polynôme en la taille de l'entrée.

2.g] Qu'en déduisez-vous sur la classe $\mathcal{BPP}$.

2.h] Même question avec $\varepsilon = 2^{-|x|}$.

Exercice 3 : Machines de Turing « je-ne-sais-pas » (Examen 2017-2018)

Dans cet exercice, nous considérons une variante des machines de Turing probabilistes. Soit Σ un alphabet fini. Une ?-*machine de Turing probabiliste* sur l'alphabet Σ est une machine de Turing probabiliste sur l'alphabet Σ qui s'arrête sur toute entrée mais qui possède trois états finaux distincts (au lieu de un seul) notés q_{acc}, q_{rej} et $q_?$. L'état q_{acc} est l'état d'acceptation, l'état q_{rej} est l'état de rejet et l'état $q_?$ est l'état d'indécision.

La machine retourne toujours une valeur dans l'ensemble $\{1, 0, ?\}$ correspondant aux états q_{acc}, q_{rej} et $q_?$ respectivement (avec l'interprétation suivante : 1 signifie que la machine accepte le mot en entrée, 0 signifie qu'elle refuse le mot en entrée, et ? signifie que la machine ne sait pas répondre sur le mot en entrée). Pour une ?-machine de Turing probabiliste $\mathcal{M}$, nous notons $\mathcal{M}(x)$ la variable aléatoire correspondant à la valeur que retourne $\mathcal{M}$ à la fin de son exécution.

Nous définissons la classe de complexité $(?)-\mathsf{PP}$ comme l'ensemble des langages L de Σ^* pour lesquels il existe une ?-machine de Turing probabiliste $\mathcal{M}$ qui s'arrête en temps polynomial en la taille de son entrée et telle que

1. pour tout $x \in \Sigma^*$, $\Pr[\mathcal{M}(x) =?] \leq 1/2$;
2. pour tout $x \in L$, $\Pr[\mathcal{M}(x) = 0] = 0$;
3. pour tout $x \notin L$, $\Pr[\mathcal{M}(x) = 1] = 0$.

3.a] Montrer que $\mathsf{P} \subseteq (?)-\mathsf{PP}$.

3.b] Montrer que $(?)-\mathsf{PP} \subseteq \mathsf{RP} \cap \text{co}-\mathsf{RP}$.

3.c] Montrer que $\mathsf{RP} \cap \text{co}-\mathsf{RP} \subseteq (?)-\mathsf{PP}$ et donc que $\mathsf{RP} \cap \text{co}-\mathsf{RP} = (?)-\mathsf{PP}$.

3.d] Que peut-on en déduire sur $(?)-\mathsf{PP}$?
Donner un argument permettant de prouver directement ce résultat (les détails de la démonstration ne sont pas demandés).

Exercice 4 : Stabilité des classes de complexité probabilistes

4.a] Soient L_1 et L_2 deux langages de la classe de complexité RP.
Montrer que les langages $L_1 \cap L_2$ et $L_1 \cup L_2$ appartiennent à la classe de complexité RP.

4.b] Soient L_1 et L_2 deux langages de la classe de complexité BPP.
Montrer que les langages $L_1 \cap L_2$ et $L_1 \cup L_2$ appartiennent à la classe de complexité BPP.

4.c] Montrer les résultats analogues pour les classes de complexité co-RP et ZPP.

Compléments

Exercice 5 : Atlantic City

Soient $T : \mathbb{N} \to \mathbb{N}$ une fonction et $L \subset \{0,1\}^*$ un langage. Nous dirons que L appartient à la classe $\widetilde{\mathcal{BPTIME}}(\mathcal{T})$, s'il existe une machine de Turing probabiliste M qui termine en temps espéré $T(|x|)$ pour tout mot $x \in \{0,1\}^*$ et telle que $\Pr[M(x) = 1] \geq 2/3$ si $x \in L$ et $\Pr[M(x) = 0] \geq 2/3$ si $x \notin L$ (où $\Pr[M(x) = b]$ pour $b \in \{0,1\}$ désigne la proportion des calculs de M qui retournent le résultat b sur l'entrée x). Nous notons $\widetilde{\mathsf{BPP}} = \bigcup_{k\geq 0} \widetilde{\mathcal{BPTIME}}(n \mapsto n^k)$.

5.a] Montrer que $\widetilde{\mathsf{BPP}} = \mathsf{BPP}$.

Exercice 6 : NP ⊆ BPP ⇒ NP = RP

6.a] Montrer que RP ⊆ NP

Le but de l'exercice est de montrer que si NP ⊆ BPP alors NP ⊆ RP (et donc NP = RP).

6.b] Rappelons que, dans le cadre des langages formels pour les problèmes de décision sur un alphabet Σ, on dit qu'un langage $\mathcal{L}_1 \subset \Sigma^*$ est *réductible en temps polynomial* à un langage $\mathcal{L}_2 \subset \Sigma^*$ (ce qui est généralement noté $\mathcal{L}_1 \leq_P \mathcal{L}_2$) s'il existe une fonction calculable en temps polynomial $f : \Sigma^* \to \Sigma^*$ telle que pour tout $w \in \Sigma^*$, $x \in \mathcal{L}_1$ si et seulement si $f(x) \in \mathcal{L}_2$.

Soient $\mathcal{L}_1$ et $\mathcal{L}_2$ deux langages sur un alphabet Σ tels que $\mathcal{L}_1 \leq_P \mathcal{L}_2$. Montrer que si $\mathcal{L}_2 \in$ RP alors $\mathcal{L}_1 \in$ RP.

6.c] Nous supposons que SAT ∈ BPP. Plus précisément, avec les techniques d'amplification vues en cours et en TD, nous supposons qu'il existe une machine de Turing probabiliste $\mathcal{M}$ qui prenant en entrée une formule booléenne Φ en n variables $(x_1, \ldots, x_n)$ sous forme normale conjonctive de m clauses retourne un bit de sorte que :

— si Φ est satisfiable, $\Pr[\mathcal{M}(\Phi) = 1] \geq 1 - 4^{-n}$;

— si Φ n'est pas satisfiable, $\Pr[\mathcal{M}(\Phi) = 0] \geq 1 - 4^{-n}$.

Considérons la machine de Turing probabiliste $\mathcal{N}$ qui exécute l'algorithme suivant :

1. initialiser une formule booléenne Ψ à Φ
2. pour i de 1 à $n - 1$
   (a) construire la formule booléenne $\Psi_{x_i=0}$ obtenue en remplaçant chaque clause par la clause obtenue en fixant la valeur x_i à 0 (c'est-à-dire les clauses où x_i apparaît sous la forme d'un littéral positif $x_i \vee \ell_1 \vee \cdots \vee \ell_t$ sont remplacées par $\ell_1 \vee \cdots \vee \ell_t$ et les clauses où x_i apparaît sous la forme d'un littéral négatif $\neg x_i \vee \ell_1 \vee \cdots \vee \ell_t$ sont supprimées).

(b) exécuter la machine de Turing $\mathcal{M}$ sur $\Psi_{x_i=0}$ et obtenir le bit $b \in \{0,1\}$

(c) si $b = 1$, $\mathcal{N}$ met à jour Ψ avec $\Psi_{x_i=0}$
si $b = 0$, $\mathcal{N}$ met à jour Ψ avec $\Psi_{x_i=1}$ (la formule booléenne obtenue en remplaçant chaque clause de Ψ par la clause obtenue en fixant la valeur x_i à 1).

3. si $\Psi_{x_n=0}$ est vraie ou $\Psi_{x_n=1}$ est vraie, retourner 1
sinon retourner 0

Montrer que si la machine de de Turing probabiliste $\mathcal{N}$ prend en entrée une formule booléenne Φ satisfiable, alors la formule booléenne Ψ obtenue à la fin de la i-ème itération de boucle est insatistifiable avec probabilité inférieure ou égale à

$$\sum_{j=1}^{i} \frac{1}{4^{n-j}} = \frac{1}{4^{n-1}} + \cdots + \frac{1}{4^{n-i}}$$

6.d] En déduire, en utilisant la machine de Turing probabiliste $\mathcal{N}$, que $\mathsf{SAT} \in \mathsf{RP}$.

6.e] Conclure.

Exercice 7 : Classe de complexité probabiliste $\mathcal{PP}$

Soit Σ un alphabet arbitraire fini (avec $\#\Sigma > 1$). Nous considérons la classe de complexité $\mathcal{PP}$ définie comme étant l'ensemble des langages $L \subseteq \Sigma^*$ pour lesquels il existe une machine de Turing probabiliste $\mathcal{M}$ telle que :

(1) $\mathcal{M}$ s'arrête sur toute entrée $x \in \Sigma^*$ et s'exécute en temps polynomial $p(|x|)$ où $|x|$ désigne la longueur de x ;

(2) pour tout $x \in L$, $\Pr[\mathcal{M}(x) = 1] > 1/2$;

(3) pour tout $x \notin L$, $\Pr[\mathcal{M}(x) = 0] > 1/2$.

7.a] Montrer que si L est un langage défini sur Σ dans $\mathcal{PP}$ et si $\mathcal{M}$ est une machine de Turing probabiliste qui vérifie les propriétés (1) et (2) précédentes, alors nous avons

(2') pour tout $x \in L$, $\Pr[\mathcal{M}(x) = 1] \geq \frac{1}{2} + \frac{1}{2^{p(|x|)}}$ où $|x|$ est la longueur de x.

7.b] Nous considérons la classe de complexité $\mathcal{PP}'$ définie comme étant l'ensemble des langages L définis sur Σ pour lesquels il existe une machine de Turing probabiliste $\mathcal{M}$ qui vérifie les propriétés (1), (2) et la propriété (3') suivante :

(3') pour tout $x \notin L$, $\Pr[\mathcal{M}(x) = 0] \geq 1/2$;

7.c] Montrer que $\mathcal{PP} \subseteq \mathcal{PP}'$

7.d] Soient L un langage de Σ^* et une machine de Turing probabiliste $\mathcal{M}$ vérifiant les propriétés (1), (2') et (3'). Considérons la machine de Turing probabiliste $\mathcal{M}'$ qui exécute $\mathcal{M}$ sur son entrée x et :

— si $\mathcal{M}$ rejette x, $\mathcal{M}'$ rejette x ;

— si $\mathcal{M}$ accepte x, $\mathcal{M}'$ rejette x avec probabilité $2^{-(p(|x|)+1)}$ et accepte x avec probabilité $1 - 2^{-(p(|x|)+1)}$.

Montrer que $\mathcal{M}'$ vérifie les propriétés (1), (2) et (3) pour le langage L.

7.e] Conclure.

7.f] En déduire que pour tout langage L de $\mathcal{PP}$, le langage $\overline{L} = \Sigma^* \setminus L$ appartient à $\mathcal{PP}$.

Les deux dernières questions sont indépendantes des précédentes. Nous considérons désormais la classe de complexité $\mathcal{PP}^+$ définie comme étant l'ensemble des langages $L \subseteq \Sigma^*$ pour lesquels il existe une machine de Turing probabiliste $\mathcal{M}$ qui vérifie les propriétés (2), (3) et la propriété (1') suivante :

(1') $\mathcal{M}$ s'arrête sur toute entrée $x \in \Sigma^*$ et s'exécute en temps polynomial **espéré** $p(|x|)$ où $|x|$ désigne la longueur de x ;

7.g] Montrer que tout langage L de $\mathcal{PP}^+$ est décidable (c'est-à-dire qu'il existe une machine de Turing déterministe qui s'arrête sur toute entrée de Σ^*, en temps fini arbitraire, qui accepte tout mot $x \in L$ et rejette tout mot $x \notin L$).

7.h] Montrer que tout langage décidable appartient à $\mathcal{PP}^+$.