COMPLEX - Partie 2

Énoncé

2020-2021 Numéro d'anonymat :

Une rédaction claire et concise sera appréciée. Toute affirmation devra être justifiée.

Une question non résolue n'empêche pas de faire les suivantes
(dans ce cas indiquez clairement que vous admettez le(s) résultat(s) de la question non faite).

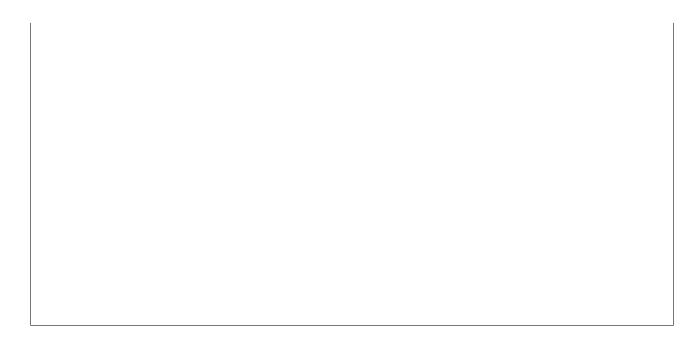
Exercice 1 : Classe de complexité probabiliste \mathcal{PP}

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 définis sur Σ pour lesquels il existe une machine de Turing probabiliste \mathcal{M} telle que :

- (1) \mathcal{M} s'arrête sur toute entrée et s'exécute en temps polynomial;
- (2) pour tout $x \in L$, $\Pr[\mathcal{M}(x) = 1] > 1/2$;
- (3) pour tout $x \notin L$, $\Pr[\mathcal{M}(x) = 0] > 1/2$;
- **1.a**] Nous considérons la classe de complexité \mathcal{PP}^{\geq} définie comme étant l'ensemble des langages L définis sur Σ pour lesquels il existe une machine de Turing probabiliste \mathcal{M} qui vérifie la propriété (1) précédente et
 - (2') pour tout $x \in L$, $\Pr[\mathcal{M}(x) = 1] \ge 1/2$;
 - (3') pour tout $x \notin L$, $\Pr[\mathcal{M}(x) = 0] \ge 1/2$;

Montrer que tout langage défini sur Σ appartient à \mathcal{PP}^{\geq} .

l.b] Nous considérons la classe de complexité $\mathcal{PP}_{1/4}$ définie comme étant l'ensemble des langages L définie sur Σ pour lesquels il existe une machine de Turing probabiliste \mathcal{M} qui vérifie la propriété (1) précédente e
(2") pour tout $x \in L$, $\Pr[\mathcal{M}(x) = 1] > 1/4$;
(3") pour tout $x \notin L$, $\Pr[\mathcal{M}(x) = 0] > 3/4$.
Montrer que $\mathcal{PP} \subseteq \mathcal{PP}_{1/4}$ Indication: Pour un langage L de \mathcal{PP} , on pourra considérer une machine de Turing probabiliste \mathcal{N} vérifiant les propriétés (1) , (2) et $(3')$ et considérer la machine de Turing probabiliste \mathcal{M}' qui — avec probabilité $1/2$, rejette son entrée x (sans faire de calcul); — avec probabilité $1/2$, exécute \mathcal{M} sur son son entrée x et retourne la réponse de \mathcal{M} .
1.c] Montrer que $\mathcal{PP}_{1/4} \subseteq \mathcal{PP}$ (et donc $\mathcal{PP}_{1/4} = \mathcal{PP}$).
Indication: Pour un langage L de $\mathcal{PP}_{1/4}$, on pourra considérer une machine de Turing probabiliste \mathcal{N} vérifiant les propriétés (1) , $(2")$ et $(3")$ et considérer la machine de Turing probabiliste \mathcal{M}' qui — avec probabilité $3/8$, accepte son entrée x (sans faire de calcul); — avec probabilité $1/8$, rejette son entrée x (sans faire de calcul); — avec probabilité $1/2$, exécute \mathcal{M} sur son son entrée x et retourne la réponse de \mathcal{M} .



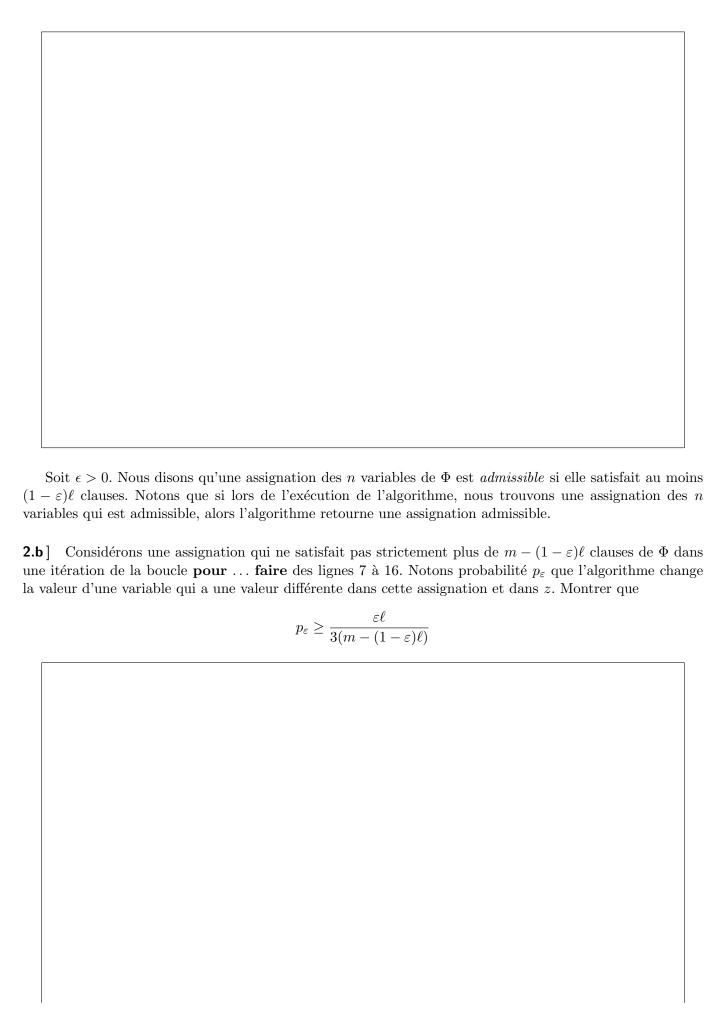
Exercice 2: Algorithme de Hirsch pour MAX-3-SAT

```
Algorithme 1: Algorithme de Hirsch pour MAX-3-SAT
    Entrée : Formule \Phi de m clauses C_1, \ldots, C_m en n variables x_1, \ldots, x_n; T \in \mathbb{N}
    Sortie: (y, \mu) \in \{0, 1\}^n \times \{0, \dots, m\} tel que \Phi(y) satisfait \mu clauses
 1 \mu^* \leftarrow -1; y^* \leftarrow (0, \dots, 0)
                                                                                                             > solution optimale
 2 pour j de 1 à T faire
        y = (y_1, \dots, y_n) \xleftarrow{\text{Constant}} \{0, 1\}^n
\mu \leftarrow \#\{i \in \{1, \dots, m\} | C_i(y) = 1\}
 3
                                                                                \triangleright nombre de clauses satisfaites par y
 4
         si \mu > \mu^{\star} alors
 5
          \mu^{\star} \leftarrow \mu \; ; \; y^{\star} \leftarrow y
 6
                                                                                pour i de 1 à n faire
 7
             \mathbf{si}\ \Phi(y) = 0\ \mathbf{alors}
 8
                  considérer une clause C_t (arbitraire) non satisfaite de \Phi
 9
                  j \stackrel{::}{\longleftarrow} VARIABLES(C_t)
                                                                                         \triangleright tirage d'une variable dans C_t
10
                  y_j \leftarrow 1 - y_j
11
                                                                                          \triangleright changement de la valeur de y_i
                  \mu \leftarrow \#\{i \in \{1, \dots, m\} | C_i(y) = 1\}
                                                                                \triangleright nombre de clauses satisfaites par y
12
                  si \mu > \mu^{\star} alors
                      \mu^{\star} \leftarrow \mu \; ; \; y^{\star} \leftarrow y
                                                                                ⊳ mise à jour de la solution optimale
14
             sinon
15
                  retourner (y^*, \mu^*)
16
17 retourner (y^*, \mu^*)
```

Dans cet exercice, nous considérons un algorithme probabiliste du à E. A. Hirsch pour résoudre le problème MAX-3-SAT de façon approchée. L'algorithme est inspiré de l'algorithme de Schöning pour le problème 3-SAT vu en cours.

Soit Φ une formule booléenne en forme normale conjonctive formée de m clauses contenant toutes au plus 3 littéraux. Soit $\ell \leq m$ le nombre maximal de clauses simultanément satisfiables pour Φ et soit $z=(z_1,\ldots,z_n)$ une assignation (optimale) des variables telle que $\Phi(z_1,\ldots,z_n)$ satisfait ℓ clauses.

2.a] Montrer que $\ell \geq m/2$.



En déduire que		$n > -\varepsilon$	_		
		$p_{\varepsilon} \ge \frac{\varepsilon}{3(1+\varepsilon)}$	(
Supposons que, lors de de celle de z en exactune telle initialisation, là p_{ε}^{t} .	ement t varia	bles (ce qui se	produit avec p	robabilité $\binom{n}{t}/2$	2^{-n}). Montrei

En som \mathbf{r} \mathbf{faire} gale à $q =$	mant pour tou e des lignes 2 à $\epsilon \left(\frac{1}{2} \left(1 + \frac{\varepsilon}{3(1+\varepsilon)}\right)\right)$	tes les valeurs p $16, l'algorithme $ $\binom{n}{2}^{n}.$	ossibles de t , is obtient une a	montrer que lo ssignation adr	rs de chaque ité nissible avec pro	ration de la b babilité supér
Montrer ale à exp	r que pour $T = 0$, 0.368 (1/q, l'algorithmet dans ce cas T	the obtient une at $C = c^n$ avec $c = c^n$	assignation adı = $(2 - 2\varepsilon/(3 +$	missible avec pro (4ε)) < 2).	babilité supér