Informatique I

Les questions sont indépendantes.

1. Soit L un langage calculable en temps polynomial, c'est à dire qu'il existe une machine de Turing déterministe M et un polynôme $p(n)$ tels que

   - M accepte L,
   - sur chaque mot de longueur n, la machine s'arrête en $p(n)$ pas de calcul.

   Montrer que L^* est également calculable en temps polynomial, où par définition

   $$L^* = \{w_1 \dots w_l : l \geq 0 \text{ et } w_i \in L \text{ pour } 1 \leq i \leq l\}.$$

2. Une séquence d'entiers $(a_0, a_1, \dots, a_{n-1})$ est appelée *unimodale* s'il existe un entier $0 \leq t \leq n-1$ tel que la séquence $(a_t, a_{t+1}, \dots, a_{t+n-1})$ est d'abord strictement croissante et après strictement décroissante, où les calculs dans les indices sont calculés modulo n. Donner un algorithme en temps $O(\log n)$ qui détermine la valeur maximale dans une séquence unimodale.
   Suggestion: Considérer d'abord le cas $t = 0$.

3. On rappelle la définition de l'ensemble des formules booléennes à n variables qu'on notera $\mathcal{F}_n$:

   - $x_i \in \mathcal{F}_n$ for $i = 1, \dots, n$.
   - Si $\phi \in \mathcal{F}_n$ et $\psi \in \mathcal{F}_n$ alors $\neg\phi \in \mathcal{F}_n$, $\phi \vee \psi \in \mathcal{F}_n$ et $\phi \wedge \psi \in \mathcal{F}_n$.

   Soit $\mathbb{N}$ l'ensemble des entiers naturels. On définit le langage $MAJSAT$ et la fonction $\#SAT : \mathcal{F}_n \to \mathbb{N}$ de la façon suivante:

   $$\begin{aligned} MAJSAT = \{\phi \in \mathcal{F}_n : \phi & \text{ est satisfaite par} \\ & \text{la majorité (plus de } 2^{n-1}\text{) assignations}\} \end{aligned}$$

   et

   $$\#SAT(\phi) = \text{nombre d'assignations satisfaisantes pour } \phi.$$

   (a) Montrer que pour tout $n > 0$, pour tout $0 < j \leq 2^n$, il existe une formule ψ à n variables et de taille polynomiale telle que le nombre d'assignations satisfaisantes pour ψ est j.

   (b) En utilisant ce résultat, montrer que si $MAJSAT$ est calculable en temps polynomial alors $\#SAT$ est aussi calculable en temps polynomial.

4. Donner un algorithme de coût linéaire qui prend en entrée un graphe biparti (non orienté) $G = (X \cup Y, E)$ décrit par listes d'adjacences, et un couplage parfait M dans ce graphe et qui décide si ce graphe possède d'autres couplages parfaits.
   Suggestion: Orienter les arêtes du couplage de X vers Y et les autres arêtes de Y vers X. Appeler G' le graphe résultant.
   Que peut-on dire des composantes fortement connexes de G' si G admet (respectivement n'admet pas) au moins deux couplages parfaits distincts?

5. Donner un algorithme linéaire qui décide si une formule 2-SAT ϕ est satisfiable. On rappelle qu'une 2-clause est la disjonction de 2 littéraux. Une instance de 2-SAT est une conjonction de 2-clauses.
   Suggestion: On pourra associer le graphe suivant G_ϕ à une formule 2-SAT ϕ : à chaque littéral l défini sur l'ensemble des variables, on associe un sommet v_l. A la clause $l \wedge l'$, on associe les arêtes $(v_{\neg l}, v_{l'})$ et $(v_{\neg l'}, v_l)$.

   (a) Que peut-on dire sur ϕ s'il existe une variable p telle que v_p et $v_{\neg p}$ figurent dans une même composante fortement connexe de G_ϕ ?

   (b) Si ϕ est satisfiable, en notant H_ϕ le graphe dont les sommets correspondent aux composantes fortement connexes de G_ϕ, et où deux sommets sont reliés si les composantes connexes correspondantes le sont, que peut-on dire des sommets associés à p et $\neg p$?