IFT503/711 – Théorie du calcul

Université de Sherbrooke

Devoir 3

Enseignant:	Manuel Lafond
Date de remise:	mardi 3 mars 2020 avant 11h59 AM
À réaliser:	individuellement ou à deux au 1^{er} cycle individuellement aux cycles supérieurs
Modalités:	à remettre en classe ou en personne, en copie imprimée ou copie manuscrite lisible
Pointage:	sur 40 points au 1^{er} cycle (+ 5pts bonus pour ★) sur 50 points aux cycles supérieurs

Question 1.

Répondez à ces questions d'échauffement:

(a) Montrez que le langage UPREM $= \{1^n : n$ est un nombre premier$\}$ est dans P. 1.5 pts

(b) Montrez que le langage COMP $= \{n : n$ encode un nombre non-premier en binaire$\}$ est dans NP. 1.5 pts

(c) Montrez que le langage PREM $= \{n : n$ encode un nombre premier en binaire$\}$ est dans co-NP. 1 pt

(d) Donnez un exemple de langage qui n'est pas dans NP $\cup$ co-NP. Justifiez votre réponse. 2 pts

Question 2.

Montrez que les deux langages décrits ci-bas sont NP-complet.

(a) Dans le problème du NO-BAD-SUM, on reçoit un ensemble I d'entiers et un ensemble S de sommes à éviter, puis on cherche un sous-ensemble $I' \subseteq I$ de taille maximum tel que toute paire de I' ne somme pas à un élément de S. Tous les entiers de I et S sont encodés en binaire. En terme de langage, on a: 6 pts

$$\text{NO-BAD-SUM} = \{\langle I, S, k\rangle : \text{il existe } I' \subseteq I \text{ avec } |I'| \geq k \text{ tel que pour tout } i, j \in I',\ i + j \notin S\}$$

Montrez que NO-BAD-SUM est NP-complet.
Suggestion: IND-SET.

(b) Une clause 2-XOR a la forme $x_1 \oplus x_2$, où x_1 et x_2 sont des variables booléennes. La clause est satisfaite si x_1 est vraie et x_2 est fausse, ou si x_2 est vraie et x_1 est fausse. Dans le problème MAX-2-XOR, on reçoit un ensemble de clauses 2-XOR sur les variables $x_1, \ldots, x_n$, et on cherche une assignation qui satisfait un maximum de clauses. En terme de langage, on a: 6 pts

$$\text{MAX-2-XOR} = \{\langle X, k\rangle : X \text{ est un ensemble de clauses 2-XOR, et il existe une assignation qui satisfait au moins } k \text{ de ces clauses}\}$$

Montrez que MAX-2-XOR est NP-complet.
Suggestion: MAX-CUT.

Question 3.

Rappelons qu'un langage L est dans PSPACE s'il existe une MT qui décide L en accédant à un nombre polynomial de cellules distinctes sur le ruban. Montrez que si PSPACE $\subseteq$ P, alors P = NP. 8 pts

Question 4.

On écrit $\langle \phi, n \rangle$ pour dénoter une formule booléenne qui utilise n variables $x_1, \ldots, x_n$. On dénote par $\#sat(\phi, n)$ le nombre d'assignations de ces n variables qui satisfont ϕ.

(a) Considérez le langage UNIQUESAT = $\{\langle \phi, n \rangle : \#sat(\phi, n) = 1\}$. Dites pourquoi l'argument suivant qui prétend que UNIQUESAT $\in$ NP est erroné. 2 pts

Soit $\langle \phi, n \rangle$ une formule à n variables. En guise de certificat vérifiant que $\langle \phi, n \rangle \in$ UNIQUESAT, on prend une assignation A qui satisfait ϕ, ce qui est facile à vérifier en temps polynomial. Il existe donc un vérificateur pour UNIQUESAT, et le langage est dans NP.

(b) Montrez que si vous avez un oracle pour UNIQUESAT, alors vous pouvez décider SAT en temps polynomial. 4 pts

(c) Soit le langage MAJSAT = $\{\langle \phi, n \rangle : \#sat(\phi, n) \geq 2^n/2\}$, i.e. les formules satisfaites par au moins la moitié des assignations. Dites pourquoi l'argument suivant qui prétend que MAJSAT $\in$ NP est erroné. 2 pts

Soit $\langle \phi, n \rangle$ une formule à n variables. Soient $A_1, A_2, \ldots, A_k$ la liste des assignations qui satisfont ϕ, qui nous serviront de certificat. Pour chaque A_i, $1 \leq i \leq k$, on peut vérifier en temps polynomial que A_i satisfait bel et bien ϕ. Une fois que c'est fait, il suffit de vérifier que $k \geq 2^n/2$. Il existe donc un vérificateur pour MAJSAT, et le langage est dans NP.

(d) Montrez que MAJSAT est NP-difficile. 6 pts

Indice: étant donné une formule booléenne ϕ sur variables $x_1, \ldots, x_n$, considérez

$$(\overline{x_0} \wedge \phi) \vee (x_0 \wedge (x_1 \vee \ldots \vee x_n))$$

où x_0 est une nouvelle variable.

★ Question 5. (cycles supérieurs)

Soient EXPTIME = $\bigcup_{k \in \mathbb{N}} DTIME(2^{n^k})$ et NEXPTIME = $\bigcup_{k \in \mathbb{N}} NTIME(2^{n^k})$ les classes des langages décidables en temps exponentiel, respectivement avec une MT déterministe et une MT non-déterministe. 10 pts

Montrez que si NEXPTIME $\neq$ EXPTIME, alors P $\neq$ NP.

Suggestion: Prenez un langage L dans NEXPTIME, puis montrez que $\{pad(w, f(w)) : w \in L\}$ est dans NP pour un certain k et une fonction $f(w)$ appropriée, où $pad(w, f(w))$ est un mot formé de w suivi de $1^{f(w)}$. Faites ensuite une preuve par contraposition.