

IFT503/711 – Théorie du calcul
Université de Sherbrooke

Devoir 3

Enseignant:	Manuel Lafond
Date de remise:	jeudi 17 février 2022 avant 10h29 AM
À réaliser:	individuellement ou à deux au 1 ^{er} cycle individuellement aux cycles supérieurs
Modalités:	à remettre via turnin
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 = $\{1^n : n \text{ est un nombre premier}\}$ est dans P. | 2 pts |
| (b) Montrez que le langage COMP = $\{n : n \text{ encode un nombre non-premier en binaire}\}$ est dans NP. | 2 pts |
| (c) Montrez que le langage PREM = $\{n : n \text{ encode un nombre premier en binaire}\}$ est dans co-NP. | 2 pts |
| (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: | 8 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) Dans le problème de <i>l'intersection bornée</i> , on reçoit des ensembles A_1, \dots, A_n et B_1, \dots, B_m . On veut savoir s'il existe un ensemble X tel que $ X \cap A_i \geq 1$ pour tout $i \in \{1, \dots, n\}$, et $ X \cap B_i \leq 1$ pour tout $i \in \{1, \dots, m\}$. | 8 pts |
|---|-------|

$$\text{INTER-BORNE} = \{\langle A_1, \dots, A_n, B_1, \dots, B_m \rangle : \exists X \ (\forall i \in \{1, \dots, n\}, |X \cap A_i| \geq 1 \wedge \forall i \in \{1, \dots, m\}, |X \cap B_i| \leq 1)\}$$

Montrez que INTER-BORNE est NP-complet.

Suggestion: 3-SAT.

Question 3.

On écrit $\langle \phi, n \rangle$ pour dénoter une formule booléenne qui utilise n variables x_1, \dots, 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 ∈ NP est erroné. 3 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 ∈ NP est erroné. 3 pts
- Soit $\langle \phi, n \rangle$ une formule à n variables. Soient A_1, A_2, \dots, 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, \dots, x_n , considérez

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

où x_0 est une nouvelle variable.

★ Question 4. (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 EXPTIME ≠ NEXPTIME, alors P ≠ 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.