Faculté des Sciences
Aix*Marseille Université

Année universitaire 2020/2021

Site : ☒ Luminy ☐ St-Charles ☐ St-Jérôme ☐ Cht-Gombert ☐ Aix-Montperrin ☐ Aubagne-SATIS
Sujet de : ☒ 1er semestre ☐ 2ème semestre ☐ Session 2 Durée de l'épreuve : 1h30
Examen de : M1 Nom du diplôme : Master Informatique
Code du module : SINAU03L Libellé du module : Complexité
Calculatrices autorisées : NON Documents autorisés : NON

Préambule : Ce sujet contient 13 questions, ce qui peut sembler considérable pour une épreuve d'une durée de 1h30. Cela étant, certaines questions peuvent facilement être traitées en moins de 3 minutes (le temps de les lire avec quelques secondes seulement pour y répondre). Par ailleurs, il a été constaté lors de la correction du partiel que certaines copies étaient parfois très difficiles à déchiffrer. Aussi, il vous est demandé de veillez à rendre votre copie aussi claire et lisible que possible.

1 Quelques questions de cours

Chaque question peut posséder 0 ou plusieurs réponses possibles. Il vous est juste demandé de préciser "*OUI*" ou "*NON*", ou encore "*Je ne sais pas*", avec les numéros des différents items, ceci sans donner de justification. Mais attention, comme dans certains QCM, la notation prendra en compte (négativement) les réponses erronées. Vous êtes donc invités à ne donner que les réponses dont vous êtes sûrs.

Question 1. Soient A et B deux langages tels que A se réduit à B et B se réduit à A avec deux réductions many-one polynomiales. Supposons que A est **NP**-complet. Que peut-on déduire ?

1. B est **NP**-complet
2. $B \in \mathbf{NP}$
3. $A \in \mathbf{P}$
4. $A \notin \mathbf{P}$

Question 2. Lesquelles des affirmations suivantes impliquent $\mathbf{P} = \mathbf{NP}$?

1. Il existe une machine de Turing déterministe qui reconnaît les mots palindromes en temps $O(n)$
2. Il existe une machine de Turing déterministe qui reconnaît le langage **SAT** en temps $O(2^n)$
3. Il existe une machine de Turing non déterministe qui reconnaît le langage **SAT** en temps $O(n)$
4. Aucune machine de Turing déterministe ne reconnaît un langage **NP**-complet en temps polynomial

2 Appartenance à la classe NP

Nous considérons ici le problème dit du "Sac à dos". Dans ce problème, on dispose d'un sac à dos dont le poids rempli ne peut excéder une certaine valeur P, d'une collection d'objets E dont on connaît le poids de chacun, à savoir un poids $p(e)$ pour un objet $e \in E$, et son utilité $u(e)$ pour la randonnée à réaliser. On veut savoir s'il est possible de remplir le sac avec une partie $F \subseteq E$ des objets disponibles, sans dépasser le poids limite P mais en assurant une utilité globale (somme de l'utilité de tous les objets emportés) supérieure ou égale à U. Le problème de décision correspondant s'énonce de la façon suivante :

> **Sac à dos**
> **Donnée :** Un ensemble de n éléments $E = \{e_1, e_2, \dots e_n\}$, une fonction p définie de E vers $\mathbb{N}$, donc telle que $\forall e \in E$, $p(e) \in \mathbb{N}$, une fonction u définie de E vers $\mathbb{N}$, donc telle que $\forall e \in E$, $u(e) \in \mathbb{N}$, un entier P et un entier U.
> **Question :** Existe-t-il un sous-ensemble $F \subseteq E$ tel que $\Sigma_{e \in F} p(e) \leq P$ et tel que $\Sigma_{e \in F} u(e) \geq U$?

Notez que les fonctions p et u peuvent facilement être représentées par des tableaux d'entiers indicés de 1 à n (ou de 0 à $n-1$ si vous préférez).

Question. Montrez que le problème **Sac à dos** appartient à **NP**. Notez que pour simplifier votre travail, nous ne vous demandons pas de rentrer dans le détail du codage binaire des nombres entiers et on supposera que les opérations arithmétiques sont exécutables en temps constant. Il vous est demandé de fournir un pseudo-code suffisamment détaillé de sorte à ce que vous puissiez fournir une évaluation précise de sa complexité. Vous avez le choix de recourir à l'une des deux techniques classiques vues en cours et en TD pour montrer l'appartenance de ce problème à **NP**.

3 Réduction

Nous rappelons les deux problèmes suivants :

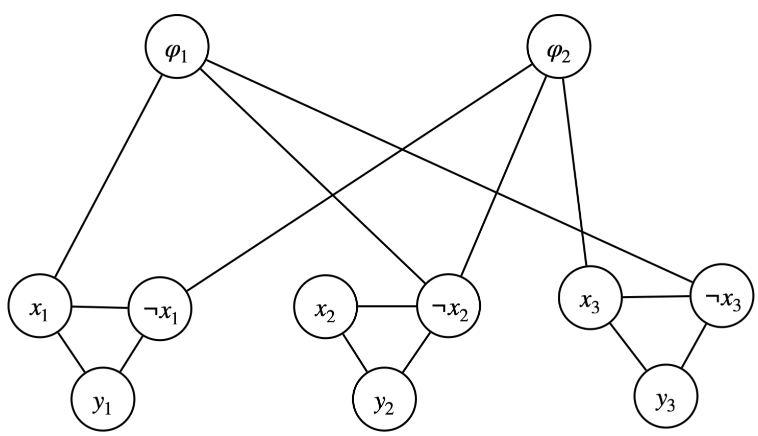
> **3-SAT**
> **Donnée :** Une formule φ en forme normale conjonctive avec exactement trois littéraux par clause.
> **Question :** Existe-t-il une affectation des variables qui satisfait φ ?

> **Ensemble Dominant**
> **Donnée :** Un graphe non orienté $G = (S, A)$ et un entier naturel k.
> **Question :** Existe-t-il un sous-ensemble $D \subseteq S$ de k sommets tel que, pour chaque sommet $s \in S$, soit $s \in D$, soit il existe une arête $\{s, t\} \in A$ telle que $t \in D$ (c'est-à-dire, le sommet s est soit dans l'ensemble D, soit relié par une arête à un sommet qui est dans l'ensemble D) ?

On veut réduire le problème **3-SAT** à **Ensemble Dominant** avec une réduction *many-one* en temps polynomial. Nous donnons ci-dessous un exemple de transformation pour la formule φ définie avec $n = 3$ variables (x_1, x_2 et x_3) et $m = 2$ clauses (φ_1 et φ_2) telles que :

$$\varphi = \underbrace{(x_1 \vee \neg x_2 \vee \neg x_3)}_{\varphi_1} \wedge \underbrace{(\neg x_1 \vee \neg x_2 \vee x_3)}_{\varphi_2}$$

Cette formule φ se transforme ainsi en une instance (G, k) du problème **Ensemble Dominant**, où $k = 3$ et $G = (S, A)$ est le graphe suivant :



Question 1. Trouvez une affectation des variables x_1, x_2, x_3 de φ qui satisfait la formule et indiquez un ensemble de sommets de G qui représente cette affectation en montrant en quoi cet ensemble de sommet constitue un ensemble dominant de G de taille 3.

Question 2. Trouvez un ensemble dominant de taille 3 dans G, différent de celui de la question précédente, et une affectation des variables x_1, x_2, x_3 de φ qui représente cet ensemble dominant.

Question 3. Il est possible de trouver des ensembles dominants de taille 3 dans G qui contiennent le sommet y_2. Donnez un tel ensemble et justifiez qu'il s'agit bien d'un ensemble dominant. À quoi correspond cet ensemble en termes d'affectation des variables de la formule φ ?

Question 4. Tout ensemble dominant de G doit contenir au moins 3 sommets. Indiquez parmi quels sommets de G il faut sélectionner ces 3 sommets et précisez pourquoi.

Question 5. Généralisez la transformation de la formule précédente φ en un graphe G à des formules **3-SAT** quelconques définies sur n variables et par m clauses, en décrivant (de forme précise, mais sans nécessairement écrire du pseudo-code) comment construire le graphe. Expliquez pourquoi cette transformation peut être calculée en temps polynomial.

Question 6. La transformation donnée dans la question précédente transforme une formule **3-SAT** quelconque en un graphe G. Il s'agit maintenant de montrer comment il est possible de transformer une formule **3-SAT** quelconque, donc une instance de **3-SAT**, en une instance de **Ensemble Dominant**, c'est-à-dire en un couple de la forme (G, k). Il faut donc préciser comment calculer la valeur de k (en justifiant que ce calcul est réalisable en temps polynomial).

Question 7. Expliquez pourquoi si une formule est satisfaisable, alors elle est transformée en un graphe qui admet un ensemble dominant de taille k.

Question 8. Expliquez pourquoi si une formule est transformée en un graphe qui admet un ensemble dominant de taille k, alors elle est satisfaisable.

Question 9. À partir des réponses aux questions précédentes, que peut-on conclure concernant la transformation proposée ? Justifiez votre réponse.

Question 10. En sachant que **3-SAT** est **NP**-complet, et que **Ensemble Dominant** appartient à **NP**, comment peut-on classer le problème **Ensemble Dominant** en utilisant la transformation proposée ? Justifiez votre réponse.