Université de Limoges **Année universitaire 2014-2015**
Faculté des Sciences et Techniques

Master 1 Informatique - Master 1 Mathématiques
Complexité et calculabilité - Feuille d'exercices n° 6

Exercice 1. Cet exercice a pour but de démontrer que le problème dit «Hitting Set» est NP-complet.

Définition. *Pour n entier strictement positif, on note S_n l'ensemble des entiers compris entre 1 et n et $\mathcal{P}(S_n)$ l'ensemble des parties de S_n. Soit $\mathcal{C} = \{C_1, \ldots, C_m\}$ un sous-ensemble de $\mathcal{P}(S_n)$ (les C_i sont donc des sous-ensembles de S_n). On dit que $K \subset S_n$ est un «hitting set» pour $(S_n, \mathcal{C})$ si chaque ensemble C_i contient au moins un élément de K. Autrement dit, K est un «hitting set» si :*

$$\forall i \in \{1, \ldots, m\}\,,\ C_i \cap K \neq \{\}.$$

Un «hitting set» de $(S_n, \mathcal{C})$ contenant k éléments sera appelé un k-HS.

Problème HITTING SET. *Étant donné $(S_n, \mathcal{C})$ et un entier k, existe-t-il un k-HS pour $(S_n, \mathcal{C})$?*

Dans la suite, on suppose que chaque entier de S_n apparaît au moins une fois dans $\mathcal{C}$. En d'autres termes, on suppose que :

$$S_n = \bigcup_{i=1}^{m} C_i. \tag{1}$$

1. Soit $\mathcal{C} = \{\{1\}, \{2,3,4\}, \{6\}, \{4,1,3,5\}\}$. Existe-t-il des $2, 3$ et 5-HS pour $(S_6, \mathcal{C})$? Si oui, il faut en donner un au moins, et sinon, il faut expliquer brièvement pourquoi.
2. On note $T(E)$ la taille binaire d'un objet E. On suppose que :
   – la taille d'un ensemble est la somme des tailles de ses éléments,
   – un entier de S_n est représenté par $|n| = \lfloor \log_2(n) \rfloor + 1$ bits,
   – on dispose d'une structure de données pour les ensembles telle que l'accès au i-ième élément E_i d'un ensemble E se fait en temps Poly$(T(E))$ et que le cardinal $|E|$ de E peut être déterminé en temps Poly$(T(E))$.

   (a) Donner un algorithme DISJOINTS qui prend en entrée deux sous-ensembles U et V de S_n et qui détermine si $U \cap V = \{\}$ en $O(|U| \times |V|)$ comparaisons d'entiers. Vérifier que cet algorithme s'exécute en temps polynomial Poly(n).

   (b) Donner un algorithme VÉRIFIEUR qui, étant donné n, k et $\mathcal{C}$, $K \subset S_n$, représentés comme ci-dessus, détermine si K est un k-HS pour $(S_n, \mathcal{C})$.

   (c) On pose $r = \sum_{i=1}^{r} |C_i|$. Déterminer T_e, taille de l'entrée T_e de l'algorithme VÉRIFIEUR, en fonction de r, n et $|K|$. On note T_e cette taille. Vérifier que $T_e \geq m$. Montrer que $T_e \geq n$ en utilisant l'hypothèse (1) ci-dessus.

   (d) Montrer que cet algorithme peut s'exécuter en temps Poly(T_e).

   (e) Que pouvez-vous en conclure ?

3. Soit $G = (S_n, A)$ un graphe (sans boucle) dont les sommets sont désignés par les élements de S_n et A est l'ensemble des arêtes (les arêtes sont donc des paires $\{i, j\}$ d'entiers distincts). On remarque que $A \subset \mathcal{P}(S_n)$.
   (a) Montrer que G admet une couverture par k sommets (problème **VERTEX COVER**, voir TD 5) si et seulement si le graphe (S_n, A) admet un k-HS.
   (b) Vérifier que la réduction **VERTEX COVER** à **HITTING SET** peut se faire en temps polynomial.
   (c) Que peut-on en déduire pour le problème **HITTING SET** ?

Exercice 2. Soit $G = (S, A)$ un graphe non orienté, où S est l'ensemble des sommets et A celui des arêtes. On note s le cardinal de S.
Soit k un entier vérifiant $1 \leq k \leq s$. Une *k-clique* de G est un sous-ensemble K de sommets de S, de cardinal k, et tel que chaque sommet de K est relié à tous les autres sommets de K par une arête de G. En d'autres termes, une k-clique est un sous-graphe de G à k sommets et qui est complet.

Dans cet exercice, on cherche à montrer que le problème suivant est *NP* -complet :

Problème CLIQUE : *étant donné un graphe G et k un entier tel que* $1 \leq k \leq s$*, existe-t-il une* k*-clique dans G ?*

Dans la suite, on considère l'instance du problème 3SAT définie par un ensemble E de k 3-clauses $C_i = \{y_{i1}, y_{i2}, y_{i3}\}$ (avec $1 \leq i \leq k$) où y_{ij} est un littéral associé à une variable prise dans l'ensemble $X = \{x_1, x_2, \ldots, x_n\}$.
On suppose que toute les variables de X apparaissent dans une clause de E, ce qui implique que $n \leq 3k$.

1. On considère le graphe suivant :

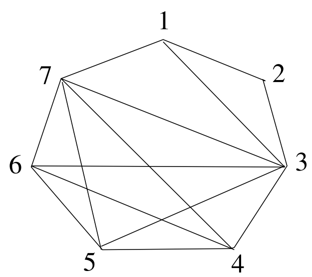

FIG. 1 – Un exemple de graphe ayant des k-cliques pour $k = 3, 4$ et 5.

   Vérifier que ce graphe contient des 3-cliques, des 4-cliques et une unique 5-clique. Vérifier qu'il ne contient pas de 6-clique. *Indication : un sommet d'une k-clique possède au moins $k - 1$ arêtes incidentes.*

2. Soit K un sous-ensemble de S.
   (a) Proposer une représentation pour G et K.
   (b) Donner un algorithme permettant de décider si K est une k-clique de G.
   (c) Montrer que cet algorithme s'exécute en temps polynomial en fonction de la taille de l'entrée.

(d) Que peut-on en conclure ?

3. On associe à l'instance du problème 3SAT ci-dessus un graphe G à $3k$ sommets, construit de la manière suivante :
   - À chaque clause est associé 3 sommets étiquetés par les littéraux de la clause.
   - On relie tous les sommets entre eux par des arêtes, sauf :
     - les sommets d'une même clause,
     - les paires de sommets correspondant à une variable et à sa négation.

   Par exemple, si $E = \{\{x_1, x_2, \bar{x}_3\}, \{\bar{x}_1, \bar{x}_2, \bar{x}_3\}, \{\bar{x}_1, x_2, x_3\}\}$, on obtient le graphe de la figure 2.

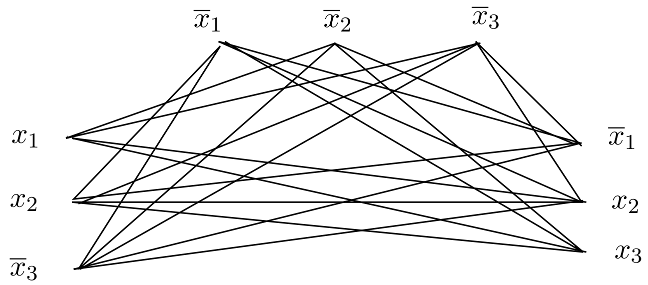

FIG. 2 – Graphe associé à l'exemple de problème $3SAT$

   (a) On se place dans le cadre de l'exemple ci-dessus.
      i. Choisir une 3-clique de ce graphe. Vérifier qu'on peut lui associer une affectation des variables qui satisfait E.
      ii. Choisir une autre affectation des variables satisfaisant E. En déduire une 3-clique.

   (b) On se place dans le cadre général.
      i. Montrer que si G possède une k-clique, alors E est satisfaisable.
      ii. Montrer que si E est satisfaisable, alors le graphe G ainsi construit possède une k-clique.

4. (a) Proposer une représentation pour E et G et montrer que la taille de G est polynomiale en la taille de E.
   (b) Esquisser un algorithme pour construire G à partir de E.
   (c) Montrer que votre algorithme s'exécute en un temps polynomial en la taille de E.

5. Conclure.