Université de Nîmes 2011/2012

Techniques mathématiques pour l'informatique
TD – Séance n° 3

Exercice 1 – On considère le problème suivant :

> PLUS GRAND ÉLÉMENT
> **Entrée :** un tableau T d'entiers
> **Question :** existe-t-il un élément de T strictement plus grand que tous les autres ?

Question 1 : Montrer que PLUS GRAND ÉLÉMENT $\in NP$

Question 2 : Montrer que PLUS GRAND ÉLÉMENT $\in P$

Correction exercice 1

Un certificat pour le problème est l'indice du plus grand élément. On peut vérifier la solution en parcourant tout le tableau.
Le problème est polynômial car il suffit pour chaque élément de vérifier si c'est un plus grand élément : algorithme naïf en $O(n^2)$. On peut faire mieux en triant le tableau, puis en testant si les deux derniers éléments sont différents ou non : $O(n\log(n))$

Fin correction exercice 1

Exercice 2 – On considère le problème suivant :

> SUBSET SUM
> **Entrée :** un multi-ensemble d'entiers E, un entier s
> **Question :** Existe-t-il un sous-ensemble non vide de E dont la somme vaut s ?

Question 1 : Résoudre le problème pour l'instance suivante : $E = \{1, -3, 2, 7, -5, -6\}, s = 0$

Question 2 : Montrer que SUBSET SUM $\in NP$.

Question 3 : Montrer que si l'on remplace la question du problème par "Existe-t-il un sous-ensemble de taille 3 dont la somme vaut s ?", alors le problème est polynomial.

Question 4 : En déduire un algorithme naïf pour résoudre SUBSET SUM, et évaluer sa complexité.

On considère maintenant le problème suivant :

PARTITION
Entrée : un multi-ensemble d'entiers E
Question : peut-on partitionner E en deux sous-ensembles de somme égale ?

Question 5 : Montrer que PARTITION $\in NP$.

Question 6 : Montrer qu'il existe une réduction polynomiale de PARTITION vers SUBSET SUM.

Correction exercice 2

1) instance positive : $\{-3, 1, 2\}$ la somme vaut bien 0.
2) un certificat est un sous ensemble. C'est bien de taille polynomiale car de taille inférieure à l'ensemble d'entrée. Pour vérifier, il suffit de calculer la somme des éléments de ce sous-ensemble, et de la comparer à s.
3) un algo naïf consiste à énumérer tous les triplets d'éléments, d'où une complexité en $O(\binom{n}{3}) = O(n^3)$.
4) algo naïf pour résoudre le problème : énumérer tous les sous-ensembles et tester si la somme de leurs éléments vaut s. L'énumération coûte $O(2^{|E|})$, le test est polynomial, d'où une complexité en $O(2^{|E|})$.
5) un certificat est la donnée de deux sous ensembles, c'est bien polynomial en la taille de l'entrée. Pour vérifier on calcule la somme des éléments des deux sous-ensembles et on compare, tout ça en temps polynomial.
6) Soit E une instance de PARTITION. On construit l'instance de SUBSET SUM composée du même ensemble E, et on prend $s = \frac{1}{2}\sum_{x \in E} x$. L'instance se construit en temps polynomial. Si on a un sous ensemble dont la somme vaut s, alors forcément l'autre sous-ensemble a aussi pour somme s et l'instance est positive pour PARTITION. Inversement, si l'instance est positive pour partition alors puisque les deux sous-ensembles sont de somme égale, ils sont bien de somme s.

Fin correction exercice 2

Exercice 3 – Problèmes de satisfaction

Le problème de satisfaction (SAT) est défini de la façon suivante : soit $V = \{x_1, ..., x_n\}$ un ensemble de variables. Une formule Φ sur V est une expression contenant des opérateurs de conjonction $\wedge$, des opérateurs de disjoinction $\vee$, et des litéraux de V (une variable x_i ou sa négation $\neg x_i$). Exemple : $\Phi = x_1 \wedge \neg x_3 \wedge (\neg x_2 \vee x_4 \vee x_5) \wedge \neg x_6$. Le but du problème est, étant donné une formule Φ sur un ensemble de variables V, de trouver une affectation $m : V \rightarrow \{True, False\}$, telle que la formule Φ soit vraie (on dira dans ce cas que l'affectation satisfait Φ). Pour la formule précédente, on remarque que l'affectation suivante fonctionne : $m(x_1) = True, m(x_2) = False, m(x_3) = False, m(x_4) = True, m(x_5) = False, m(x_6) = True$.

Clause : une clause d'ordre k est une expression de la forme $v_1 \vee v_2 \vee ... \vee v_k$ avec chaque v_i un literal (positif ou négatif), i.e. une disjonction de litéraux.

Forme normale conjonctive (CNF) : une formule Φ est sous forme normale conjonctive si elle est écrite comme une conjonction de clauses.

Remarque : Tout formule peut s'écrire de manière équivalente sous une forme normale conjonctive.

Formellement, le problème est défini ainsi :

> SAT
> **Entrée :** une formule Φ sur n variables et m clauses
> **Question :** Existe-t-il une affectation de V satisfaisant Φ ?

Question 1 : Résoudre le problème pour la formule Φ suivante :

$$\begin{array}{rc} & (x_1 \vee \neg x_2) \\ \wedge & (\neg x_3 \vee \neg x_1 \vee x_4) \\ \wedge & (x_2) \\ \wedge & (\neg x_4 \vee \neg x_1) \end{array}$$

Question 2 : Montrer que SAT $\in NP$.

Question 3 : En déduire un algorithme naïf pour résoudre SAT et évaluer sa complexité.

On dit qu'une clause est *Horn* si elle contient au plus un litéral positif. On appelle Horn-SAT le problème SAT restreint aux clauses Horn.

Question 4 : Montrer que Horn-SAT est polynomial.

Correction exercice 3

1) x_2 doit être vraie, donc x_1 doit être vraie, donc x_4 doit être fausse, et donc x_3 doit être fausse. C'est bien une instance positive.
2) un certificat est une affectation de chaque variable. On la vérifie facilement en temps polynomial.
3) on essaie toutes les affectations possibles, puis on les vérifie en temps polynomial, d'où une complexité en $O(2^n)$ avec n le nombre de variables.
4) on traite les clauses d'arité 1, on propage, puis on met toutes les variables restantes à faux.
5) **Fin correction exercice 3**

Exercice 4 – Problème d'ensemble indépendant
Le problème INDEPENDENT SET est défini de la façon suivante :

INDEPENDENT SET
Entrée : un ensemble de variables $E = \{x_1, ..., x_n\}$
un ensemble $\mathcal{S} = \{S_1, ..., S_m\}$ avec $S_i \subseteq E$ et $|S_i| = 2$ pour tout $i \in \{1, ..., m\}$
un entier k
Question : Existe-t-il un sous ensemble $I \subseteq E$ de taille au moins k tel que pour tout $i, j \in I$, $\{i, j\} \notin \mathcal{S}$?

Question 1 : Résoudre le problème pour l'instance suivante : $E = \{x_1, x_2, x_3, x_4, x_5\}$, $\mathcal{S} = \{\{1, 5\}, \{2, 5\}, \{3, 5\}, \{4, 5\}, \{2, 3\}, \{3, 4\}\}$, et $k = 3$

Question 2 : Montrer que INDEPENDENT SET $\in NP$

Question 3 : En déduire un algorithme naïf pour le résoudre et évaluer sa complexité.

On suppose vrai le théorème suivant :

Théorème 1 *Le problème* 3-SAT *(où chaque clause est de taille* 3*) est* NP*-complet.*

Question 4 : Montrer qu'il existe une réduction polynomiale de 3-SAT vers INDEPENDENT SET. En déduire que le problème INDEPENDENT SET est NP-complet.

Correction exercice 4

1) Une solution est $\{x_1, x_2, x_4\}$
2) Un certificat est un sous-ensemble de E de taille k, c'est bien de taille polynomiale. On vérifie en énumérant toutes les paires, et en parcourant $\mathcal{S}$: $O(k^2 + |\mathcal{S}|)$.
3) On énumère tous les sous-ensembles de taille k en temps $O(n^k)$, puis on teste en temps polynomial.

4) Soit une instance de 3-SAT composée d'un ensemble de variables $V = \{v_1, ..., v_n\}$ et d'un ensemble de clauses $\mathcal{C} = \{C_1, ..., C_m\}$, avec pour tout $i \in \{1, ..., m\}$ $C_i = l_i^1 \vee l_i^2 \vee l_i^3$, où chaque l_i^j correspond à une variable v_k ou sa négation $\neg v_k$. Nous construisons l'instance d'INDEPENDENT SET SUIVANTE, composée d'un ensemble de variables E et d'un ensemble d'ensembles de variables $\mathcal{S}$. Pour tout $i \in \{1, ..., m\}$, on ajoute à E trois variables x_i^1, x_i^2 et x_i^3 correspondant aux trois literaux l_i^1, l_i^2 et l_i^3 de C_i. On dira que x_i^1, x_i^2 et x_i^3 est le gadget de la clause C_i. On ajoute à $\mathcal{S}$ les ensembles $\{x_i^1, x_i^2\}, \{x_i^1, x_i^3\}$ et $\{x_i^2, x_i^3\}$. Puis, pour tout $i, j \in \{1, ..., k\}$ avec $i \neq j$, et tout $k, l \in \{1, 2, 3\}$, si x_i^k et x_i^l correspondent à des litéraux opposés d'une même variable (v_u et $\neq v_u$ pour un certain $u \in \{1, ..., n\}$), alors on ajoute à $\mathcal{S}$ l'ensemble $\{x_i^k, x_i^l\}$. Enfin, on pose $k = m$. La construction de E, $\mathcal{S}$ est k peut clairement se réaliser en temps polynomial. Ainsi, il reste à montrer que l'instance $(E, \mathcal{S}, k)$ est positive pour INDEPENDENT SET si et seulement si l'instance $(V, \mathcal{C})$ est positive pour 3-SAT :
$\Rightarrow$: supposons qu'il existe un ensemble $I \subseteq E$ de taille $k = m$ tel que pour tout $x, y \in I$, $\{x, y\} \notin \mathcal{S}$. Puisque pour tout $i \in \{1, ..., m\}$ chaque paire de variables du gadget de C_i est dans $\mathcal{S}$, I contient au plus un élément parmi $\{x_i^1, x_i^2, x_i^3\}$. Puisque $|I| = m$ et qu'il y a m gadgets, I contient exactement un élément parmi $\{x_i^1, x_i^2, x_i^3\}$. De plus, pour pour toute paire d'élément (x, y) de I, et par définition de I, x ne correspond pas à la négation de la variable correspondant à y, et inversement. En effet, autrement on aurait $\{x, y\} \in \mathcal{S}$ ce qui contredirai la définition de I. On en déduit que pour tout

élément x de I, on peut affecter à la variable lui correspondant la valeur telle que le litéral soit évalué à *vrai*. Ainsi, chaque clause est satisfaite et l'instance $(V, \mathcal{C})$ de 3-SAT est positive.
$\Leftarrow$: inversement, supposons qu'il existe une affectation des variables V telle que l'instance de 3-SAT soit positive et construisons un ensemble indépendant $I \subseteq E$ de taille k. Pour chaque clause C_i avec $i \in \{1, ...m\}$, il existe $k \in \{1, 2, 3\}$ tel que le litéral l_i^k soit évalué à *vrai*. On ajoute à I la variable x_i^k. Puisque $k = m$, on a bien $|I| = k$. De plus, pour toute paire $\{x, y\}$ d'éléments de I, $\{x, y\} \notin \mathcal{S}$. En effet, puisque on a pris une variable par gadget, la seule possibilité pour avoir $\{x, y\} \in \mathcal{S}$ serait que x et y correspondent à une même variable, et que le litéral associé à x soit la négation de celui associé à y. Cela est impossible puisque on a uniquement choisit des variables dont les litéraux sont évalués à *vrai*.

Fin correction exercice 4