

TD 5 : Codage des entrées

Théorie de la complexité S5.

2022

Exercice 1 — *Faiblement et fortement NP-Complet*

Pour chacun des problèmes NP-Complets du premier TD, déterminez s'il est faiblement ou fortement NP-Complet.

► Correction

(HAM) : les instances sont des graphes G de taille n . On a donc $|x| = n$, $l(x) = n$, $\max(x) = 0$. Ce problème est NP-Complet au sens fort car il n'y a pas d'entier. (On considère que $|x| = n$ mais en vrai, cette taille vaut au moins m , le nombre d'arêtes puisqu'il faut encoder chaque arête. Puisque le nombre d'arêtes est inférieure à n^2 , dans le cadre de notre cours ça n'a pas d'importance.)

(3SAT) : les instances sont des formules booléennes sur n variables avec m clauses. On a donc $|x| = n + 3m$, $l(x) = n + 3m$ et $\max(x) = 0$. Ce problème est NP-Complet au sens fort car il n'y a pas d'entier.

(VERTEX COVER) *Soit G un graphe non orienté et K un entier, existe-t-il un sous-ensemble des nœuds de G de taille inférieure à K et couvrant toutes ses arêtes.* Les instances sont des graphes G de taille n et un entier $K \in \mathbb{N}$. On a donc $|x| = n + \log(K)$, $l(x) = n + 1$, $\max(x) = K$. On peut voir que ce problème est trivial si $K > n$ puisqu'on peut alors prendre tous les nœuds. On peut donc logiquement penser qu'une réduction polynomiale de Karp de 3SAT vers Vertex Cover se restreindra au cas où $K \leq n$, donc au cas où $\max(x) \leq l(x)$. La réduction est là : <http://web.mit.edu/~neboat/www/6.046-fa09/rec8.pdf>. Ce problème est donc NP-Complet au sens fort.

(INS) *Soit G un graphe non orienté et K un entier, existe-t-il un sous-ensemble des nœuds de G de taille inférieure à K et non reliés deux à deux.* Les instances sont des graphes G de taille n et un entier $K \in \mathbb{N}$. On a donc $|x| = n + \log(K)$, $l(x) = n + 1$, $\max(x) = K$. On peut voir que ce problème est trivial si $K > n$ puisque la réponse est nécessairement non. On peut donc logiquement penser qu'une réduction polynomiale de Karp de 3SAT vers INS se restreindra au cas où $K \leq n$, donc au cas où $\max(x) \leq l(x)$. Une idée de réduction est là : <https://stackoverflow.com/questions/47657574/how-can-3-sat-be-reduced-to-independent-set>. Ce problème est donc NP-Complet au sens fort.

(SET COVER) *Soit X un ensemble et S des sous-ensembles de X et K un entier, existe-t-il un sous-ensemble C de S de taille inférieure à K et couvrant X .* Les instances sont des ensembles X de taille n , p sous-ensembles de X et un entier $K \in \mathbb{N}$. On a donc $|x| = n + \log(K)$, $l(x) = n \cdot p + 1$, $\max(x) = K$. On peut voir que ce problème est trivial si $K > p$ ou $n = 0$ puisque la réponse est nécessairement oui. On peut donc logiquement penser qu'une réduction polynomiale de Karp de (3SAT) vers (SET COVER) se restreindra au cas où $K \leq p$ et $n \geq 1$, donc au cas où $\max(x) \leq l(x)$. La réduction dans le TD3 vérifie cette propriété. Ce problème est donc NP-Complet au sens fort.

(W-VERTEX COVER) *Soit G un graphe non orienté, des poids $\omega : V \rightarrow \mathbb{N}$ et K un entier, existe-t-il un sous-ensemble des nœuds de G de poids total inférieur à K et couvrant toutes ses arêtes.* Les instances sont des graphes G de taille n , des poids ω et un entier $K \in \mathbb{N}$. On a donc $|x| = n + \sum_{v \in V} \log(\omega(v)) + \log(K)$, $l(x) = 2n + 1$, $\max(x) = \max_{v \in V}(\log(\omega(v)), K)$. On peut voir que

ce problème reste NP-Complet si on se restreint au cas où $\omega(v) = 1$ pour tout v , puisqu'on a alors le problème (VERTEX COVER). Considérons la réduction de (VERTEX COVER) vers (W-VERTEX COVER) qui, connaissant une instance x de (VERTEX COVER) fixe tous les poids à 1 pour former une instance y de (W-VERTEX COVER). On a alors $\max(y) = K$ qui est borné polynomialement par rapport à $\max(x)$ et $l(x)$. Puisque (VERTEX COVER) est NP-Complet au sens fort, on peut appliquer le dernier théorème du cours pour prouver que (W-VERTEX COVER) est NP-Complet au sens fort.

De même on peut prouver que (WINS) et (W-SET COVER) sont NP Complets au sens fort.

(LOP) : soit un graphe orienté $G = (V, A)$ de taille n , des poids sur les arêtes $\omega(a)$, un noeud s , un noeud t , et un entier K , existe-t-il un chemin élémentaire de s vers t de poids supérieur à K ? On a donc $|x| = n + \sum_{a \in A} (\log(\omega(a))) + 2 + \log(K)$, $l(x) = n + |A| + 2 + 1$ et $\max(x) = \max_{a \in A}(\omega(a), K)$.

Il existe une réduction de (HAM) vers (LOP) qui consiste à garder le graphe G de l'instance de (HAM) de mettre des poids de 1 sur toutes les arêtes et poser $K = n - 1$, rajouter un noeud s , et un noeud t , relier s à tous les noeuds sauf t avec des arcs de poids 0, relier tous les noeuds sauf s à t avec des arcs de poids 0. Sur ces instances, $\max(x) = n - 1 \leq l(x)$, donc (LOP) est NP Complet au sens fort.

(SUBSET SUM) : un ensemble X de k entiers et un entier B . Existe-t-il un sous-ensemble de X dont la somme fait B . On a donc $|x| = \log(B) + \sum_{x \in X} (\log(x))$, $l(x) = k + 1$, $\max(x) = \max(B, x)$.

Il existe un algorithme en $O(kB)$ pour résoudre (SUBSET SUM) (cf Chapitre 1 du cours de RO). Cet algo est pseudo polynomial car $k \leq k + 1 = l(x)$ et $B \leq \max(B, xi) = \max(x)$ donc $O(kB) = O(l(x)\max(x))$ donc polynomial en $l(x)$ et $\max(x)$.

Or (SUBSET SUM) est NP Complet donc il est NP Complet au sens faible.

Alternative : si B est codé en unaire, $O(kB)$ est polynomial, donc SUBSET SUM est NP Complet mais polynomial si on code ses entiers en unaire. Donc NP Complet au sens faible.

(PARTITION) : un ensemble X de k entiers. Existe-t-il une partition de X dont la somme des deux parties est égale. On a donc $|x| = \sum_{x \in X} (\log(x))$, $l(x) = k$, $\max(x) = \max(X)$. Il existe un algorithme en $O(k \sum_{x \in X} \lim X)$ pour résoudre (PARTITION) (Cet algo est le même que pour SUBSET SUM). Cet algo est pseudo polynomial car $k \leq k + 1 = l(x)$ et $\sum_{x \in X} \lim \leq k \cdot \max(X) = l(x) \max(x)$ donc $O(kB) = O(l(x)^2 \max(x))$ donc polynomial en $l(x)$ et $\max(x)$.

Or (PARTITION) est NP Complet donc il est NP Complet au sens faible.

Alternative : si X est codé en unaire, $O(k \sum_{x \in X} \lim X)$ est polynomial, donc (PARTITION) est NP Complet mais polynomial si on code ses entiers en unaire. Donc NP Complet au sens faible.

Une preuve similaire montre que (KNAPSACK) est NP-Complet au sens faible.

Exercice 2 — Quelques résultats

1. Que prouve une réduction depuis un problème faiblement NP-Complet ?

► Correction

Si Π_1 est faiblement NP-Complet et $\Pi_1 \preceq \Pi_2$, alors on sait que Π_2 est NP-Difficile. Cela ne prouve pas que Π_2 est fortement ou faiblement NP-Complet. (Rappelez vous qu'on peut réduire tous les problèmes de NP à 3SAT qui est fortement NP-Complet.)

Dans tous les cas, il n'est pas possible d'appliquer le dernier théorème du cours.

2. Montrer que démontrer une réduction polynomiale de Karp d'un problème fortement NP-Complet Π_1 vers un problème Π_2 ne démontre pas que Π_2 est fortement NP-Complet.

► **Correction**

Il existe une réduction de (3SAT) vers (SUBSET SUM) et ce dernier est NP-Complet au sens faible. Pour que la NP Complétude soit au sens fort, il faut en plus que l'entier $max(x)$ de l'instance de (SUBSET SUM) produite soit borné polynomialement par rapport à la taille de l'instance de (3SAT).

3. Montrer que, si un problème est fortement NP-Complet (et est donc NP-Complet si on encode ses entiers en unaires), le problème est toujours NP-Complet si on encode ses entiers en binaire.

► **Correction**

Considérons les deux versions Π_u et Π_b du problèmes où les entiers sont respectivement codés en unaire et en binaire.

Il existe une réduction polynomiale de Karp de Π_u vers Π_b qui consiste à réencoder tous les entiers en binaire. La réduction préserve trivialement les instances positives et négatives. Elle est polynomiale (même meilleure que polynomiale) car il faut un temps $O(\log(n))$ pour encoder un entier n en binaire. Puisque, dans Π_u , l'instance est encodée en unaire, la taille de l'entier n est n , on a bien une transformation en temps polynomial.

Exercice 3 — Faiblement et fortement polynomial

1. L'algorithme du simplexe (résolvant des programmes linéaires) est-il fortement polynomial ? Si on se restreint aux programmes n'ayant pas plus de 2 variables, l'algorithme du simplexe est-il fortement polynomial ?

► **Correction**

Un programme linéaire est constitué d'une matrice A , d'un vecteur b et d'un vecteur c et on cherche le vecteur x maximisant cx tel que $Ax \leq b$. L'algorithme du simplexe se déplace de solution de base réalisable en solution de base réalisable jusqu'à trouver une solution optimale.

On note n le nombre de variables (et colonnes de A) et m le nombre de contraintes (et lignes de A).

Aujourd'hui, on ne sais pas si l'algorithme du simple est polynomial, car on ne connaît pas d'algorithme polynomial permettant d'éviter de cycler entre des solutions de bases réalisables. On ne sait donc pas s'il est fortement polynomial. Par contre, on peut voir que chaque itération de l'algorithme du simplexe est linéaire en le nombre de variables. Si on ne cycle pas entre les solutions de bases, on ne peut pas avoir plus d'itérations que le nombre de solutions de bases. Il y a au plus m^n solutions de bases réalisables dans un programme linéaire. Enfin, chaque itération effectue $O(n)$ opérations élémentaires ou arithmétiques.

Donc dans le cas de $n = 2$ variables, en supposant que le simplexe ne boucle pas, il s'exécute en $O(m^2)$ et est donc fortement polynomial. C'est aussi le cas pour n'importe quel nombre fixé de variables.

2. Décrivez un algorithme fortement polynomial pour calculer la moyenne de n nombres.

► **Correction**

L'algorithme naïf suivant est fortement polynomial.

ENTRÉES: n nombres x_1, x_2, \dots, x_n

SORTIES: La moyenne de ces nombres

1: $s \leftarrow 0$

2: **Pour** $i \leq n$ **Faire**

3: $s \leftarrow s + x_i$

4: **Renvoyer** s/n .

En considérant les opérations arithmétiques comme élémentaires, la complexité de cet algorithme est $O(n)$, donc polynomial en $l(x)$.

3. On veut calculer le n^{e} décimal de $\sqrt{2}$ avec la méthode de Newton et Raphson.
 - Soit f une fonction strictement convexe et croissante sur $D \subset \mathbb{R}$, on recherche un réel $x \in D$ tel que $f(x) = 0$. (On suppose qu'un tel réel existe). Soit $x_0 \in D$ et $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$, montrez que $(x_n)_{n \in \mathbb{N}}$ est convergente et tend vers x .
 - Montrez que $x_n - x \leq f(x_n)/f'(x)$
 - Soit $f(x) = x^2 - 2$. Décrire un algorithme calculant le n^{e} décimal de $\sqrt{2}$.
 - Montrer que, si $x_0 < \sqrt{2} + 0.3$, l'algorithme s'arrête en n itérations. En déduire la complexité de l'algorithme. Est-il pseudo-polynomial, fortement polynomial ou faiblement polynomial ?

► **Correction**

Pour cette dernière question, la réponse dépend de l'encodage de n . Si n est binaire, cet algorithme est exponentiel. S'il est unaire, alors l'algorithme est faiblement polynomial puisque la complexité dépend de $\max(x)$ et n'est donc pas borné uniquement par rapport à $l(x)$. Par contre il est pseudo-polynomial.

Exercice 4 — Graphe succinct Un circuit booléen C est un graphe orienté sans circuit

contenant n sources représentant les entrées (1 ou 0), un unique puits représentant une sortie (1 ou 0 également). Les autres nœuds sont des portes logiques ET, OU et NON. Pour toute entrée $x \in \{0, 1\}^n$, on note $C(x)$ la sortie associée.

Soit G un graphe non orienté contenant 2^n nœuds numérotés de 1 à 2^n . On définit la *représentation succincte* de G comme étant un circuit booléen C avec $2n$ entrées (dit autrement deux entrées x et y de taille n) tel que, pour tout couple de nœuds numérotés x et y , la sortie $C(x, y) = 1$ si et seulement si x et y sont reliés par une arête. Attention : tous les graphes n'ont pas une représentation succincte.

On veut prouver que le problème suivant est NP-Complexe.

(TRIANGLE-SUCC?) : soit C un circuit booléen représentant un graphe G , est-ce que G contient un triangle (une clique de taille 3) ?

1. Montrer que (TRIANGLE-SUCC?) est dans NP.
2. Soit $G_1 = (V_1, E_1)$ et $G_2 = (V_2, E_2)$ tels que $V_1 \subset V_2$ et tels que ces deux graphes ont une représentation succincte, montrez que $G = (V_2, E_1 \cup E_2)$ possède une représentation succincte.
3. Soit φ une instance de (SAT?) avec n variables x_1, x_2, \dots, x_n . Soit $G_\varphi = (V_\varphi, E_\varphi)$ le graphe suivant. V_φ contient $2^n + 1$ nœuds : $(v_0, v_1, \dots, v_{2^n})$. On relie v_i et v_{2^n} si E_φ si et seulement si, quand on affecte VRAI à x_j si le j^{e} bit de la représentation binaire de i est 1 et FAUX sinon, φ est satisfaite. Montrez que G_φ possède une représentation succincte.
4. Soit $G_2 = (V_2, E_2)$ le graphe contenant $2^n + 2$ nœuds $(v_0, v_1, \dots, v_{2^n+1})$ et où chaque nœud est relié à v_{2^n+1} (excepté v_{2^n+1} lui-même). Montrer que G_2 possède une représentation succincte.
5. En déduire que $G = (V_2, E_\varphi \cup E_2)$ possède une représentation succincte.
6. Montrez G possède un triangle si et seulement si φ est satisfiable.
7. En déduire que (TRIANGLE-SUCC) is NP-Complete.