TD 8 - complexité en temps réductions et NP-complétude

Réductions

Réduction Many One. Un problème (ou langage) A est *many-one réductible* à un problème (ou langage) B s'il existe une fonction calculable totale $f : \Sigma^* \to \Sigma^*$ telle que $\forall x,\ x \in A \iff f(x) \in B$. On note alors $A \leqslant_m B$.

Réduction polynomiale. Si l'on tient compte de l'efficacité du calcul de f, on peut établir la notion de *réduction polynomiale*. Une fonction $f : \Sigma^* \to \Sigma^*$ est calculable en temps polynomial s'il existe une machine de Turing $\mathcal{M}$ qui s'exécute en temps polynomial, et qui s'arête avec $f(x)$ sur sa bande, pour n'importe quelle entrée x écrit initialement sur la bande. On note alors la réduction par $\leqslant_m^P$.

Exercice 1 *Quelques propriétés*
Montrez les assertions suivantes :
- « Si $A \leqslant_m^P B$ et $B \in \mathsf{P}$, alors $A \in \mathsf{P}$. »
- « Si $A \leqslant_m^P B$ et $B \leqslant_m^P C$, alors $A \leqslant_m^P C$. »

Conséquences :
- Si $A \leqslant_m^P B$ et B peut être résolu en temps polynomial, alors A peut aussi être résolu en temps polynomial.
- Si $A \leqslant_m^P B$ et A ne peut pas être résolu en temps polynomial, alors B ne peut pas être résolu en temps polynomial.

Exercice 2 *Décision vs Optimisation ; équivalence de problèmes*
Problème **Ensemble stable**

> *entrée* : un graphe $G = (V, E)$.
> *sortie* : un plus grand sous-ensemble de sommets $S \subseteq V$ tel que S soit un *stable* de G, c'est-à-dire que pour tout sommet $u, v \in S$, on a $\{u, v\} \notin E$.

- Formulez ce problème comme un problème de décision.
- Expliquez pourquoi, du point de vue de la résolution en temps polynomial, il n'y a pas de différence significative entre *sa version d'optimisation* et *sa version de décision.*

Problème **Couverture de sommets (vertex cover)**

> *entrée* : un graphe $G = (V, E)$; un entier $k \in \mathbb{N}$.
> *sortie* : existe-t-il un sous-ensemble de sommets $S \subseteq V$ de cardinalité au plus k, et tel que S soit une *couverture de sommets* de G, c'est-à-dire que pour chaque arête $\{u, v\} \in E$, soit $u \in S$, soit $v \in S$?

- Montrez que ces deux problèmes sont aussi difficiles l'un que l'autre (formellement, ils se réduisent l'un à l'autre).

NP-complétude

Difficulté et complétude Soit A un problème :
— A est dit NP-*difficile* si pour tout $B \in \mathsf{NP}$, $B \leqslant_m^P A$.
— A est dit NP-*complet* si pour tout $B \in \mathsf{NP}$, $B \leqslant_m^P A$ et $A \in \mathsf{NP}$.

Preuve de NP-complétude. Si l'on suppose que $\mathsf{P} \neq \mathsf{NP}$, la NP-complétude d'un problème peut être vue comme une indication qu'il ne puisse pas être résolu temps polynomial.

Pour montrer qu'un problème X est NP-complet, on adoptera la stratégie suivante. (Il existe différents types de réductions: *Karp réduction*, *Cook réduction*, *Turing réduction*. L'étudiant intéressé pourra se diriger vers la bibliothèque universitaire pour enrichir sa culture $\ddot\smile$).

1. On montre que $X \in \mathsf{NP}$;
2. On choisit un problème Y, connu comme étant NP-complet;
3. On considère une instance arbitraire s_Y du problème Y, et on montre comment construire, en temps polynomial, une instance s_X du problème X qui satisfait les propriétés :
   - Si s_Y est une instance positive de Y alors s_X est une instance positive de X.
   - Si s_X est une instance positive de X alors s_Y est une instance positive de Y.

Exercice 3 *Réductions polynomiales*
Soit Q un problème de décision. Pour chacune des questions suivantes, indiquer ce que l'on peut en déduire de la complexité du problème Q (est-il dans P, dans NP, NP-difficile, NP-complet), et expliquer pourquoi.

1. Soit $P_1 \in \mathsf{P}$. Que peut-on en déduire si $P_1 \leqslant_m^P Q$? et si $Q \leqslant_m^P P_1$?
2. Soit $P_2 \in \mathsf{NP}$. Que peut-on en déduire si $P_2 \leqslant_m^P Q$? et si $Q \leqslant_m^P P_2$?
3. Soit P_3 un problème NP-difficile. Que peut-on en déduire si $P_3 \leqslant_m^P Q$? Si $Q \leqslant_m^P P_3$?
4. Soit P_4 un problème NP-complet. Que peut-on en déduire si $P_4 \leqslant_m^P Q$? Si $Q \leqslant_m^P P_4$?

Exercice 4 *Domination*
Problème **Ensemble dominant (dominating set)**
> *entrée* : un graphe $G = (V, E)$; un entier $k \in \mathbb{N}$.
> *sortie* : existe-t-il un sous-ensemble de sommets $D \subseteq V$ de cardinalité au plus k, et tel que D soit un *ensemble dominant* de G, c'est-à-dire que pour tout sommet $v \in V$, soit $v \in D$, soit il existe un sommet u voisin de v et tel que $u \in D$?

▲ Montrez que **Ensemble dominant** $\in \mathsf{NP}$.
◆ Montrez que **Couverture de sommets** $\leqslant_m^P$ **Ensemble dominant**.
⬟ En supposant que **Couverture de sommets** est NP-complet, que peut-on en déduire sur **Ensemble dominant** ?

Exercice 5 *Autour de TSP*

Problème **Voyageur de commerce** ***(version d'optimisation)***

entrée : Un graphe complet $G = (V, E)$ et une fonction de poids sur les arêtes $c : E \to \mathbb{N}$.
sortie : Une permutation ordonnées $(v_1, v_2, \dots, v_n)$ des sommets de G qui minimise la valeur $\sum_{1 \leqslant i < n} c(\{v_i, v_{i+1}\}) + c(\{v_1, v_n\})$.

Problème **Cycle Hamiltonien**

entrée : Un graphe orienté $G = (V, E)$.
sortie : Un cycle hamiltonien de G, c'est-à-dire une permutation ordonnée $(v_1, v_2, \dots, v_n)$ des sommets de G telle que $\{v_i, v_{i+1}\} \in E$ pour tout i, $1 \leqslant i < n$, et $\{v_n, v_1\} \in E$.

Problème **3-SAT**

entrée : – un ensemble $X = \{x_1, \dots, x_n\}$ de variables booléennes ;
— une formule propositionnelle $F = c_1 \wedge c_2 \wedge \dots \wedge c_m$ où chaque clause c_i est de la forme $l_1 \vee l_2 \vee \vee l_3$; chaque littéral l_i est soit une variable booléenne x_i, soit sa négation $\overline{x_i}$.
sortie : une affectation de valeurs aux variables booléennes de X, tel que la formule F soit satisfaite.

- Transformez le problème **Voyageur de commerce** sous la forme d'un problème de décision, que nous appelerons **TSP**.
- Montrez que **Cycle Hamiltonien** $\leqslant_m^P$ **TSP**.
- Montrez que **Cycle Hamiltonien** appartient à NP.
- Montrez que **3-SAT** $\leqslant_m^P$ **Cycle Hamiltonien**.
- En supposant **3-SAT** NP-complet, que peut-on en déduire sur **Cycle Hamiltonien** ?
- Que peut-on dire de **TSP** ?
- Expliquez pourquoi **Chemin Hamiltonien** est également NP-complet.

Exercice 6 *Ordonnancement*

Problème **Subset Sum**

entrée : n entiers naturels $(w_1, w_2, \dots, w_n)$ et un entier W.
sortie : existe-t-il un sous-ensemble de $\{w_1, w_2, \dots, w_n\}$ dont la somme est égale à W ?

Problème **Ordonnancement**

entrée : Un ensemble de n tâches devant être traitées sur une machine. Pour chaque tâche i, on dispose d'une *date de disponibilité* r_i indiquant le moment où la tâche est disponible pour être traitée par la machine ; une *date limite* d_i de traitement est aussi donnée, ainsi que la *durée de traitement* t_i de la tâche. (Ainsi, chaque tâche doit être allouée sur un unique créneau de l'intervalle $[r_i, d_i]$.)
sortie : Un ordonnancement des n tâches de sorte à respecter les dates limite.

Problème **Regroupement de Composantes**

entrée : Un graphe G non connexe et un entier k.
sortie : Existe-t-il un sous-ensemble de ses composantes connexes dont l'union contient exactement k sommets ?

- Montrez que **Ordonnancement** appartient à NP.
- Montrez que **Subset Sum** $\leqslant_m^P$ **Ordonnancement**.
- En déduire la NP-complétude de **Ordonnancement**.
- Établir la complexité de **Regroupement de Composantes**.

Exercice 7 *Coloration*

Problème **k-Coloration**

entrée : un graphe $G = (V, E)$.

sortie : existe-t-il une coloration des sommets de G utilisant au plus k couleurs ?

Pour rappel, une coloration des sommets d'un graphe $G = (V, E)$ est une fonction $c : V \to \mathbb{N}$ telle que $\forall u, v \in V \ : \ \{u, v\} \in E \to c(u) \neq c(v)$.

- ▲ Montrez que **3-SAT** $\leqslant_m^P$ **3-Coloration**.
- ◆ Montrez que **3-Coloration** $\leqslant_m^P$ k-**Coloration**, pour $k \geqslant 3$.

Exercice 8 *Les problèmes de la classe* NP

Pour chacune des affirmations suivantes, indiquer si elle est « vraie », « fausse », « vraie sous l'hypothèse P = NP », ou « vraie sous l'hypothèse P ≠ NP ».

1. Un problème NP-difficile est un problème dans NP.
2. Un problème NP-difficile est un problème NP-complet.
3. Un problème NP-complet est un problème NP-difficile.
4. Un problème NP-complet est un problème dans NP.
5. Un problème NP-complet n'est pas dans P.