# IFT503/711 - Exercices sur P et NP

Manuel Lafond

**Exercice 0**

On a définit le temps d'une MT non-déterministe $M$ comme la longueur minimum d'un chemin acceptant. Montrez que si on remplace "minimum" par "maximum" dans cette définition, NP reste inchangée.

Pour une version plus détaillée du problème, supposons qu'on définit le *temps-max* de $M$ comme la longueur *maximum* d'un chemin acceptant. On définit ensuite NP-max comme l'ensemble des langages $L$ tels qu'il existe une MT non-déterministe $M$ dont les mots acceptés sont $L$, et qui accepte tous ses mots en temps-max dans $O(n^k)$ où $k \in O(1)$.

Montrez que NP = NP-max.

**Exercice 1**

On dénote par co-NP l'ensemble des langages dont le complément est dans NP (le complément d'un langage $L$ est l'ensemble $\overline{L}$ des mots qui ne sont *pas* dans $L$).

Montrez que P $\subseteq$ NP $\cap$ co-NP.

**Exercice 2**

Soit $L \in$ NP. Montrez qu'il existe une constante $k$ telle que $L \in DTIME(2^{n^k})$, où $n$ représente la taille de l'entrée.

**Exercice 3**

Un langage $L$ est co-NP-complet si $L \in$ co-NP, et si, pour chaque langage $A \in$ co-NP, $A \leq_P L$.

Montrez qu'un langage $L$ est NP-complet si et seulement si $\overline{L}$ est co-NP-complet.

**Exercice 4**

Montrez que $A_{NTM} = \{\langle M, x\rangle : M$ encode une MT non-déterministe qui accepte $x\}$ est NP-difficile.

**Exercice 5**

Soit EXPTIME $= \bigcup_{k\in\mathbb{N}} DTIME(2^{n^k})$. Montrez que NP $\subseteq$ EXPTIME, possiblement en utilisant la notion de certificat.

**Exercice 6**

Dans le problème de la couverture minimum, on reçoit des ensembles, et on veut choisir un minimum de ces ensembles pour couvrir tous les éléments. Plus formellement, soit $\mathcal{S} = \{S_1, \ldots, S_n\}$ une collection d'ensembles, et soit $U$ un ensemble qu'on appelle *l'univers*. Le langage correspondant au problème de la couverture minimum est le suivant :

$$\text{SET-COVER} = \{\langle \mathcal{S}, U, k\rangle : \text{il existe } \mathcal{S}' \subseteq \mathcal{S} \text{ tel que } |\mathcal{S}'| \leq k \text{ et } \bigcup_{S\in\mathcal{S}'} S = U\}$$

Montrez que SET-COVER est NP-complet.

**Exercice 7**

Dans un graphe, un sous-ensemble de sommets $X \subseteq V(G)$ est *dominant* si chaque sommet de $V(G)\setminus X$ a au moins un voisin dans $X$. Considérez le langage

$$\begin{aligned}\text{DOM-SET} = \{\langle G, k\rangle : & G \text{ est un graphe dans lequel il existe} \\ & \text{un ensemble dominant } X \subseteq V(G) \text{ de taille } k \text{ ou moins}\}\end{aligned}$$

Montrez que DOM-SET est NP-complet.

**Exercice 8**

Une collection d'ensembles $\{S_1, \ldots, S_\ell\}$ est dite *disjointe* si, pour tout $i, j$ avec $1 \leq i < j \leq \ell$, on a $S_i \cap S_j = \emptyset$. Dans le problème SET-PACKING, on reçoit une collection d'ensembles $\mathcal{S}$ et on doit choisir une sous-collection disjointe de taille maximum. En terme de langage, on a :

$$\text{SET-PACKING} = \{\langle \mathcal{S}, k\rangle : \text{il existe une sous-collection } \mathcal{S}' \subseteq \mathcal{S} \text{ disjointe telle que } |\mathcal{S}'| \geq k\}$$

Montrez que SET-PACKING est NP-complet.

**Exercice 9**

Dans le problème HAMPATH, on veut savoir si $G$ contient un chemin Hamiltonien, qui est un chemin qui contient chaque sommet exactement une fois. En terme de langage, HAMPATH contient les graphes qui ont un chemin Hamiltonien. C'est un langage NP-complet (ce qui n'est pas le but de l'exercice, mais vous pouvez essayer).

Soit $G$ un graphe. Un *arbre couvrant* de $G$ est un sous-graphe $G'$ de $G$ tel que $G'$ est un arbre connectant tous les sommets de $G$. Dans le problème MINDEG-AC, on veut savoir s'il existe un arbre couvrant dont le degré maximum ne dépasse pas $k$ :

$$\text{MINDEG-AC} = \{\langle G, k\rangle : G \text{ est un graphe dans lequel il existe un arbre couvrant tel que chaque sommet a un maximum de } k \text{ voisins}\}$$

Montrez que MINDEG-AC est NP-complet.
*Suggestion :* utiliser la NP-complétude de HAMPATH.

**Exercice 10**

Dans le problème 2-SAT, on reçoit une formule $\phi$ en forme CNF dans laquelle chaque clause a deux variables. On veut savoir si $\phi$ est satisfaisable. Montrez que 2-SAT est dans P.

**Exercice 11**

(Défi !) Dans le problème MAX-2-SAT, on reçoits un ensemble de clauses CNF avec chacune deux variables, et on veut en satisfaire un maximum. En terme de langage, on a

$$\text{MAX-2-SAT} = \{\langle C, k\rangle : C \text{ est un ensemble de clauses CNF, et il existe une assignation qui en satisfait au moins } k\}$$

Montrez que MAX-2SAT est NP-complet.
*Note :* ceci n'est pas facile. Ma réduction passe par le problème MAX-CUT (voir définition sur le web, vous pouvez utiliser le fait qu'il est NP-complet).