# Tutorat de Complexité - 2

La totalité des problèmes posés sont tirés des références suivantes :

- Le cours *Complexité* de Pascal Vanier,
- Le cours *Introduction à la complexité* de Paul Rozière,
- Le cours *Complexity Theory* de Chris Umans,
- Le cours *Computationnal complexity* d'Abhijit Das,
- Le cours *Langages formels, calculabilité et complexité* d'Olivier Carton,
- Le cours *Langages formels, calculabilité et complexité* de Damien Vergnaud.

## 1 Réduction vers `SAT`

*Définition* 1.1 (`CH`). On s'intéresse au problème suivant :

- Entrée : Soit $\mathcal{G}$ un graphe orienté,
- Question : Existe-il un *cycle hamiltonien* dans $\mathcal{G}$, c'est-à-dire un cycle passant une et une seule fois par tous les nœuds de $\mathcal{G}$ ?

**Question 1.** Sans utiliser le théorème de Cook-Levin, montrer que le problème `CH` se réduit en temps polynomial au problème `SAT`.

Pour un graphe à $n$ nœuds, on peut introduire, pour coder une suite de nœuds de longueur $n$ : $(u_1, \dots, u_n)$, une relation binaire $R$ où $R_{i,j}$ signifie $u_i = j$ et axiomatiser propositionnellement le fait que la suite de nœuds représentée par $R$ est un chemin Hamiltonien.

> Commençons par remarquer qu'il n'est pas standard de faire une réduction vers `SAT` puisque la démarche générale des cours de complexité est de montrer que `SAT` est NP-complet, puis ensuite de réduire `SAT` à d'autres problèmes pour montrer qu'ils sont NP-durs.
>
> Notons $n$ le nombre de nœuds de $\mathcal{G}$. On va encoder un cycle comme étant une permutation de $\{1, \dots, n\}$ telle que pour tout $i$, les images de $i$ et $(i + 1 \mod n)$ sont reliées dans $\mathcal{G}$, en utilisant l'indication. On va vouloir encoder différentes choses :

- La fonctionnalité : tout $i$ a au plus une image par $u$ :

$$\Phi_1 = \bigwedge_{i,j} \bigwedge_{k \neq j} \neg R_{i,j} \vee \neg R_{i,k}$$

- La totalité : tout $i$ a au moins une image par $u$ :

$$\Phi_2 = \bigwedge_{i} (\bigvee_{j} R_{i,j})$$

- L'injectivité : tout $j$ a au plus un antécédent par $u$ :

$$\Phi_3 = \bigwedge_{i,j} \bigwedge_{k \neq i} \neg R_{i,j} \vee \neg R_{k,j}$$

- La surjectivité : tout $j$ a au moins un antécédent par $u$ :

$$\Phi_4 = \bigwedge_{j} (\bigvee_{i} R_{i,j})$$

Notons que ces quatre formules sont de taille quadratique ou cubique en la taille du graphe et indépendant de celui-ci, donc constructibles en temps polynomial.

Il reste à exprimer que deux images d'entiers consécutifs modulo $n$ sont reliées par un arc dans $\mathcal{G}$. La formule est

$$\Phi_5 = \bigwedge_{i,j,k} \neg R_{i,j} \vee \neg R_{(i+1 \bmod n),k} \vee E_{j,k}$$

où les $E_{j,k}$ sont des constantes exprimant le fait que les nœuds $i$ et $j$ sont reliés dans $\mathcal{G}$. Cette formule contient $n^3$ clauses qui nécessitent chacune de savoir si une arête appartient à $\mathcal{G}$, donc elle est constructible en temps polynomial.

$\bigwedge_{i=1}^{5} \Phi_i$ dont les variables sont les $R_{i,j}$ est satisfaisable si et seulement s'il existe une permutation des nœuds de $\mathcal{G}$ qui est un cycle, donc on a bien une réduction de `CH` à `SAT`.

Notons que dans l'énoncé usuel du problème `SAT` les clauses ne contiennent pas de constantes booléennes, il serait donc plus conforme à la spécification du problème d'écrire

$$\Phi_5 = \bigwedge_{i,j} \bigwedge_{\substack{k \\ (j,k) \notin E_{\mathcal{G}}}} \neg R_{i,j} \vee \neg R_{(i+1 \bmod n),k}$$

ce qui ne change pas le reste du travail qui a été fait.

## 2 Pas facile d'organiser des examens

*Définition* 2.1 (`EDT`). On s'intéresse au problème d'emploi du temps suivant :

- Entrée :
  - $n \in \mathbb{N}$ un nombre d'examens,
  - $p \in \mathbb{N}$ un nombre de créneaux disponibles,
  - $C = \{C_1, \ldots, C_k\}$ un ensemble de sous-ensembles de $\{1, \ldots, n\}$, où $C_i$ représente les examens que doit passer l'élève $i$.
- Question : Est-il possible de trouver un emploi du temps (un fonction de $\{1, \ldots, n\}$ à $\{1, \ldots, p\}$) tel que tous les élèves peuvent aller à tous leurs examens ?

**Question 2.** Montrer que `EDT` est NP-complet.

Commençons par montrer que `EDT` est NP.

Un certificat est un tableau contenant $n$ cases, la case $i$ contenant un entier entre 1 et $p$ indiquant sur quel créneau est organisé l'examen $i$. Un tel tableau est de taille $n \log(p)$ qui est polynomial en la taille de l'entrée étant donné que l'on peut considérer que tous les examens sont passés par au moins un étudiant et donc que $C$ est de taille $n \log(n)$ et que $n \geqslant p$ sinon le problème n'a aucun intérêt.

Pour chaque $C_i$, pour chaque paire d'entiers dans $C_i$, on doit accéder aux deux cases correspondantes du tableau et vérifier qu'elles contiennent des entiers différents. Ceci peut être fait en espace logarithmique.

Montrons maintenant que `EDT` est NP-dur.

On sait que le problème `COLORIAGE` est NP-complet, on va donc le réduire à `EDT`. Soit $f$ la fonction qui à $\mathcal{G} = (V, E)$ et $k \in \mathbb{N}$ associe $(|V|, k, E)$, celle-ci est calculable en temps polynomial et transforme une instance du problème `COLORIAGE` en une instance de `EDT` (on considère ici que le graphe $\mathcal{G}$ étant non orienté, $E$ est un ensemble de paires). Cette fonction associe les nœuds aux examens, les arêtes aux étudiants et les couleurs aux créneaux.

Il y a équivalence entre la $k$-colorabilité de $\mathcal{G}$ et la compatibilité des examens dans l'instance $f(\mathcal{G}, k)$ : Si le graphe est $k$ coloriable, il suffit d'affecter au même créneau les nœuds de la même couleur, et réciproquement.

## 3 Problèmes de sous-graphes

*Définition* 3.1 (`TRIANGLE FREE`). On s'intéresse au problème suivant :

- Entrée : $\mathcal{G}$ un graphe,
- Question : $\mathcal{G}$ contient-il un triangle (un graphe complet à 3 éléments) ?

**Question 3.** Montrer que `TRIANGLE FREE` est dans L.

On énumère tous les triplets de nœuds dans $\mathcal{G}$ et on va lire sur l'entrée s'il y a une arête entre chacun d'entre-eux, ce qui ne nécessite qu'un espace polynomial.

*Définition* 3.2 (Graphes isomorphes). Deux graphes $\mathcal{G} = (V, E)$ et $\mathcal{G}' = (V', E')$ sont *isomorphes* s'il existe une fonction bijective $h : V \to V'$ telle que $\{v_1, v_2\} \in E$ si et seulement si $\{h(v_1), h(v_2)\} \in E'$.

*Définition* 3.3 (`SGI`). On s'intéresse au problème suivant, appelé *Subgraph Isomorphism* :

- Entrée : Deux graphes $\mathcal{G}$ et $\mathcal{G}'$,
- Question : Existe-t-il un sous-graphe de $\mathcal{G}$ isomorphe à $\mathcal{G}'$ ?

**Question 4.** Montrer que `SGI` est NP-complet.

Commençons par montrer que `SGI` est dans NP.

Un certificat est une fonction totale qui a chaque nœud de $\mathcal{G}'$ associe un nœud de $\mathcal{G}$. Un tel certificat est de taille $|\mathcal{G}'|\log(|\mathcal{G}'|)$ qui est plus petit que le plus grand des graphes en entrée, donc polynomial en l'entrée. Pour chaque arête dans $\mathcal{G}'$ on vérifie qu'il y a une arête entre les images dans $\mathcal{G}$, ce qui se fait en espace logarithmique. On vérifie aussi en espace logarithmique que la fonction est injective.

Montrons maintenant que `SGI` est NP-dur.

On sait que `CLIQUE` est NP-dur et c'est manifestement un cas particulier de `SGI` donc `SGI` est NP-dur. La fonction de réduction est $(\mathcal{G}, k) \mapsto (\mathcal{G}, C_k)$ où $C_k$ désigne la clique à $k$ éléments. Elle se calcule en temps polynomial et il y a bien équivalence entre la présence d'une $k$-clique dans $\mathcal{G}$ et le fait que $C_k$ soit isomorphe à un sous-graphe de $\mathcal{G}$.

*Définition* 3.4 (`BIG CYCLE`). On s'intéresse au problème suivant :

- Entrée : Un graphe $\mathcal{G}$,
- Question : Existe-t-il un cycle de taille supérieure à $\frac{\lfloor \mathcal{G} \rfloor}{2}$ dans $\mathcal{G}$ ?

**Question 5.** Montrer que `BIG CYCLE` est NP-complet.

Commençons par montrer que `SGI` est dans NP.

Un certificat est une suite d'entiers entre 1 et $|\mathcal{G}|$. La suite en question doit être d'une taille comprise entre $\frac{\lfloor \mathcal{G} \rfloor}{2}$ et $|\mathcal{G}|$, donc d'une taille inférieure à $|\mathcal{G}|\log(|\mathcal{G}|)$ donc polynomiale en la taille de l'entrée. Il faut vérifier que la suite ne contient pas deux fois le même nœud et que deux entiers adjacents dans la suite correspondent à deux nœuds reliés dans le graphe, ce qui se fait en espace polynomial.

Montrons maintenant que `CSI` est NP-dur.

On va réduire `CH` à `CSI`, car on sait que `CH` est NP-complet. La fonction de réduction est $f : \mathcal{G} \mapsto \mathcal{G}+\mathcal{G}$ où $+$ est l'opération qui juxtapose deux graphes de façon disjointe. Cette fonction $f$ se calcule en temps polynomial. Si $\mathcal{G}$ contient un cycle hamiltonien, il y a un grand cycle dans $\mathcal{G} + \mathcal{G}$ et réciproquement.

## 4 Voyage, voyage

*Définition* 4.1 (Graphe fortement connexe). Un graphe est *fortement connexe* si pour tout nœuds $a$ et $b$, il existe un chemin permettant d'aller de $a$ à $b$.

*Définition* 4.2 (STRONGLY CONNECTED). On s'intéresse au problème suivant :

- Entrée : Un graphe $\mathcal{G}$,
- Question : $\mathcal{G}$ est-il fortement connexe ?

**Question 6.** Montrer que STRONGLY CONNECTED est NL-complet.

Commençons par montrer que STRONGLY CONNECTED est dans NL.

On considère que $\mathcal{G} = (n, E)$ où $n \in \mathbb{N}$ et $E \in \{1...n\}^2$.

Une machine vérifie STRONGLY CONNECTED avec 5 compteurs.

```
for a=1 to n
 for b=1 to n
  p=a
  nb=0
  while p!=b and nb<=n
   guess p' such that p and p' are linked in G
   p:=p'
   nb+=1
  if p==b
  then continue
  else return False
return True
```

Montrons maintenant que STRONGLY CONNECTED est NP-dur.

On va réduire ACCESSIBILITY à STRONGLY CONNECTED, car on sait que ACCESSIBILITY est NL-complet. Soit $\mathcal{G} = (n, E)$ un graphe et $u$ et $v$ deux nœuds de $\mathcal{G}$. La fonction $f : (\mathcal{G}, u, v) \mapsto \mathcal{G}' = (n, E \cup \{(v, x) | x \leqslant n\} \cup \{(x, u) | x \leqslant n\})$ est calculable en temps polynomial. S'il y a un chemin entre $u$ et $v$ dans $\mathcal{G}$, $\mathcal{G}'$ est fortement connexe puisqu'on peut aller de tout point en $u$, puis de $u$ à $v$ et enfin de $v$ en tout point. S'il n'y a pas de chemin entre $u$ et $v$ dans $\mathcal{G}$, il n'y en a pas non plus dans $\mathcal{G}'$, donc $\mathcal{G}'$ n'est pas fortement connexe.