TD Théorie n°2

NP-difficulté par réduction

Introduction.

Le but de ce TD est de réaliser des réductions polynomiales pour montrer la **NP**-difficulté d'un problème. Les réductions du TD Théorie n°1 étaient généralement plus simples, et se ressemblaient beaucoup. Dans ce TD, les réductions sont assez différentes. C'est pour cela que des indications sont données pour les réductions, et sur « comment encoder le problème Q dans une entrée du problème R ».

Une réduction polynomiale de Q à R sera notée $Q \preccurlyeq_{\mathrm{p}} R$. En particulier, si Q est **NP**-difficile, alors R l'est aussi.

On admettra, comme dans le cours, la **NP**-difficulté du problème SAT. On admettra également la **NP**-difficulté du problème 3SAT, et du problème CLIQUE.

3SAT :
Entrée. Une formule φ sous forme 3CNF
Sortie. La formule φ est-elle satisfiable ?

CLIQUE :
Entrée. Un graphe $G = (V, E)$ et un entier k
Sortie. Existe-t-il k sommets dans V qui sont deux à deux connectés ?

Attention. Pour les problèmes de graphes (CLIQUE, et STABLE mais pas COUV.SOMMET) définis dans la feuille d'exercice n°13, les problèmes se ressemble à ceux définis plus tard, mais ils sont ***différents*** : on ne demande pas un stable de taille inférieure à k, mais de taille égale à k.

Tout comme dans le TD Théorie n°1, les questions sont indépendantes. Il y a au total 24 réductions.

Si cela n'est pas précisé, les graphes ne seront pas orientés, et les ensembles sont *finis*.

Pour chaque question, on pourra se baser sur la **NP**-difficulté du problème dans le titre de la section, et les problèmes dans la section que vous avez déjà démontrés (pas de boucle de réduction).

I. Réductions depuis 3SAT.

Une valuation « *pas tout égal* » d'une CNF est une valuation[1] telle que toutes les clauses contiennent au moins un littéral ayant pour valuation ***V*** et un littéral ayant pour valuation ***F***.

Q1. Démontrer la **NP**-complétude du problème suivant.

PAS-TOUT-ÉGAL-3SAT :
Entrée. Une formule φ sous forme d'une 3CNF
Sortie. Est-ce que φ a une valuation « pas tout égal » ?

> **Indication.** Soit y une nouvelle variable ($y \notin \mathrm{vars}\ \varphi$). Commencer par réduire 3SAT à PAS-TOUT-ÉGAL-4SAT en ajoutant y à chaque clause.

[1] *i.e.* un environnement propositionnel $\rho \in \mathbb{B}^{\mathrm{vars}\ \varphi}$.

Q2. Démontrer la **NP**-complétude du problème suivant.

1-PARMI-3-SAT :
Entrée. Une formule φ sous forme d'une 3CNF
Sortie. Est ce que φ a une valuation telle que chaque clause contient exactement un littéral ayant pour valuation $\boldsymbol{V}$?

Indication. Réduire depuis 3SAT. Puis, pour chaque clause, définir 8 variables $y_{000}, \dots, y_{111}$ et réduire 3SAT au problème 1-PARMI-AU-PLUS-7 SAT (le nom devrait être suffisamment explicite). L'idée est que *précisément* une de ces variables a une valuation valant $\boldsymbol{V}$ pour chaque clause.

Q3. Démontrer la **NP**-complétude du problème suivant.

PROG-ENTIÈRE-0-1 :
Entrée. Une matrice $\boldsymbol{A} \in \mathscr{M}_{m,n}(\{0,1\})$ et un vecteur $\boldsymbol{b} \in \mathscr{M}_{1,m}(\{0,1\})$
Sortie. Est-ce qu'il existe $\boldsymbol{x} \in \mathscr{M}_{1,n}(\{0,1\})$ tel que $\boldsymbol{Ax} \leq \boldsymbol{b}$[2] ?

Indication. Il s'agit d'une simple réduction à 3SAT.

Q4. Démontrer la **NP**-complétude du problème suivant.

PROG-QUADRATIQUE :
Entrée. Un tenseur[3] $\boldsymbol{A} \in \mathbb{Z}^{n\times n\times n}$ et un vecteur $\boldsymbol{b} \in \mathbb{Z}^n$
Sortie. Existe-t-il $x \in \mathbb{R}^n$ tel que $\sum_{i=1}^{n}\sum_{j=1}^{n} A_{i,j,k}x_ix_j = b_j \bmod 2$, avec $k \in [\![1, n]\!]$?

Indication. Réduire à 3SAT et utiliser que fait que $u \mathbin{\tilde{\vee}} v = u + v - uv$[4] pour $u, v \in \{0,1\}$ et $u^2 = u$.

Dans le jeu de solitaire suivant (pas exactement celui du TP OCaml n°1), on nous donne un tablier de taille $m \times m$. Sur chacun de ces m^2 positions, on peut placer uniquement une pierre bleue, uniquement une pierre rouge, ou rien du tout. On joue en retirant les pierres du jeu. Vous gagnez si chaque colonne contient uniquement des pierres d'une couleur, et chaque ligne contient au moins une pierre.

Q5. (Plus complexe) Démontrer la **NP**-complétude du problème suivant.

JEU-SOLVABLE :
Entrée. Une configuration initiale de ce jeu du solitaire
Sortie. Est-ce qu'il est possible de gagner depuis cette position ?

Q6. Démontrer la **NP**-complétude du problème suivant.

AUTO-MOT-NON-ACCEPTÉ :
Entrée. Un automate fini $\mathscr{A}$, et un entier $n \in \mathbb{N}$
Sortie. Existe-t-il un mot de taille n qui n'est pas accepté par $\mathscr{A}$?

Remarque. Ce problème n'est pas trivialement dans **NP**.

Indication. Réduire depuis 3SAT. Pour chaque formule φ, construire un automate tel qu'une entrée w de taille n soit acceptée si $[\![\varphi]\!]^\rho = \boldsymbol{F}$, où $n = \mathrm{card}(\mathrm{vars}\ \varphi)$.

[2] On dit ici que $\boldsymbol{a} \leq \boldsymbol{b}$ *ssi* $\forall i \in I, a_i \leq b_i$ où $\boldsymbol{a} = (a_i)_{i\in I}$ et $\boldsymbol{b} = (b_i)_{i\in I}$.
[3] Généralisation d'une matrice, dans ce cas-ci, c'est une matrice 3D.
[4] où $\tilde{\vee}$ est l'opérateur OU sur les entiers $\{0,1\}$, ce n'est pas une formule

Q7. Démontrer la **NP**-complétude du problème suivant.

LANG-REG-FINIS-DIFFÉRENTS :
Entrée. Deux automates finis $\mathscr{A}$ et $\mathscr{B}$
Sortie. Les langages de $\mathscr{A}$ et $\mathscr{B}$ sont-ils finis et sont-ils différents ?

> **Indication.** La partie compliquée est de montrer que ce problème est dans **NP**. Remarquons qu'un automate ayant un langage fini ne contient pas de cycles. Pour la réduction, utiliser le fait que le problème AUTO-MOT-NON-ACCEPTÉ reste **NP**-difficile si on ajoute la contrainte que $\mathscr{A}$ n'accepte pas les mots de longueurs différents de n.

II. Réductions depuis 1-PARMI-3-SAT.

Une *partition* d'un ensemble U est une famille finie d'ensembles $S_1, \dots, S_n \subseteq U$ qui sont deux à deux disjoints, et tels que $\bigcup_{i=1}^{n} S_i = U$.

Q8. Démontrer la **NP**-complétude du problème suivant.

CONTIENT-PARTITION :
Entrée. Un ensemble U et une famille C de sous-ensembles de U
Sortie. Existe-t-il une sous-famille $C' \subseteq C$ qui est une partition de U ?

Q9. Démontrer la **NP**-complétude du problème suivant.

SUBSET-SUM :
Entrée. Un ensemble $X \subseteq \mathbb{Z}$ et $k \in \mathbb{Z}$
Sortie. Existe-t-il un sous-ensemble $I \subseteq X$ tel que $\sum_{i \in I} i = k$?

> **Indication.** Réduire depuis CONTIENT-PARTITION en considérant le développement des entiers dans une certaine base.

Q10. Démontrer la **NP**-complétude du problème suivant. Il est appelé PARTITION dans la feuille d'exercice n°13.

SET-PARTITION :
Entrée. Une famille finie $(x_1, \dots, x_n)$ d'entiers
Sortie. Existe-t-il un ensemble $I \subseteq [\![1, n]\!]$ tel que $\sum_{i \in I} x_i = \sum_{i \notin I} x_i$?

Q11. Démontrer la **NP**-complétude du problème suivant.

ENSEMBLES-DISJOINTS :
Entrée. Une famille C d'ensembles et un entier k
Sortie. Y a-t-il k ensembles dans C qui sont deux à deux disjoints ?

> **Indication.** Réduire depuis 1-PARMI-3-SAT de la même manière que CONTIENT-PARTITION.

III. Réductions depuis CLIQUE.

Q12. Démontrer la **NP**-complétude du problème suivant.

ENSEMBLE-INDÉPENDANT :
Entrée. Un graphe $G = (V, E)$ et un entier k
Sortie. Y a-t-il un sous-ensemble $U \subseteq V$ de k sommets qui sont deux à deux non connectés, *i.e.* $(U \times U) \cap E = \emptyset$?

Une *couverture par sommets* d'un graphe G est un sous-ensemble S' des sommets de G tel que chaque arête de G a au moins une extrémité dans S'.

Q13. Démontrer la **NP**-complétude du problème suivant.

COUVERTURE-SOMMETS :
Entrée. Un graphe $G = (V, E)$ et un entier k
Sortie. Le graphe G a-t-il une couverture par sommets de taille inférieure à k ?

Q14. Démontrer la **NP**-complétude du problème suivant.

SET-COVER :
Entrée. Un ensemble E, une famille $(S_1, \dots, S_m)$ de sous-ensembles de E et un entier k
Sortie. Y a-t-il k ensembles dans la famille dont l'union vaut E ?

Q15. Démontrer la **NP**-complétude du problème suivant.

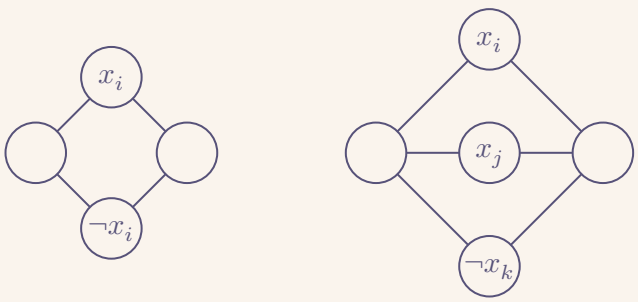
SET-PACKING :
Entrée. Un ensemble E, une famille $(S_1, \dots, S_m)$ de sous-ensembles de E et un entier k
Sortie. Y a-t-il k ensembles dans la famille qui sont deux à deux disjoints ?

IV. Réductions depuis 3SAT, problèmes de graphes.

Q16. Démontrer la **NP**-complétude du problème suivant.

PAIRES-INTERDITES :
Entrée. Un graphe $G = (V, E)$, deux sommets s et $t \in V$, et un n-uplet de couples $((u_1, v_1), \dots, (u_n, v_n)) \in (V \times V)^n$
Sortie. Existe-t-il un chemin de s à t qui visite *au plus* un sommet de chaque paire (u_i, v_i) ?

Indication.



Variable x_i Clause $x_i \wedge x_j \wedge \neg x_k$

Figure 1. Indications pour la réduction

Q17. Démontrer la **NP**-complétude du problème suivant.

CHEMIN-HAM.-ORIENTÉ :
Entrée. Un graphe orienté $G = (V, E)$, deux sommets s et $t \in V$.
Sortie. Existe-t-il un chemin de s à t qui visite chacun des sommets de V *exactement* une fois ?

Q18. Démontrer la **NP**-complétude du problème suivant.

CHEMIN-HAMILTONIEN :
Entrée. Un graphe non-orienté $G = (V, E)$, deux sommets s et $t \in V$.
Sortie. Existe-t-il un chemin de s à t qui visite chacun des sommets de V *exactement* une fois ?

Un *cycle* dans un graphe est un chemin ayant même sommet de départ et d'arrivée.

Q19. Démontrer la **NP**-complétude du problème suivant.

CYCLE-HAM.-ORIENTÉ :
Entrée. Un graphe orienté $G = (V, E)$
Sortie. Existe-t-il un cycle qui visite chaque sommet *exactement* une fois ?

Q20. Démontrer la **NP**-complétude du problème suivant.

CYCLE-HAMILTONIEN :
Entrée. Un graphe non-orienté $G = (V, E)$
Sortie. Existe-t-il un cycle qui visite chaque sommet *exactement* une fois ?

Q21. Démontrer la **NP**-complétude du problème suivant.

TRAVELLING-SALES-MAN :
Entrée. Une liste de distances deux à deux entre n points, et un entier k
Sortie. Existe-t-il un cycle de longueur au plus k qui contient tous les points ? [5]

V. Réductions depuis 3COLORIABLE.

Le *coloriage* d'un graphe $G = (V, E)$ est une fonction $f : E \to C$, où C est l'ensemble des couleurs tels que deux sommets connectés n'aient jamais la même couleur.

Q22. Démontrer la **NP**-complétude du problème suivant.

3COLORIABLE :
Entrée. Un graphe G
Sortie. Le graphe G est-il coloriable avec 3 couleurs ?

> **Indication.** Réduisons ce problème depuis PAS-TOUT-ÉGAL-3SAT. Notons les couleurs $C = \{\mathrm{V}, \mathrm{F}, \mathrm{N}\}$. On interprète les couleurs V comme ***V*** et F comme ***F***, et on les représente sur une palette.

[5] Autrement dit, existe-t-il une liste $p_1, \ldots, p_m$ qui contient chaque point au moins une fois, telle que l'inégalité soit vérifiée $d(p_1, p_2) + \cdots + d(p_{m-1}, p_m) + d(p_n, p_1) \leq k$?

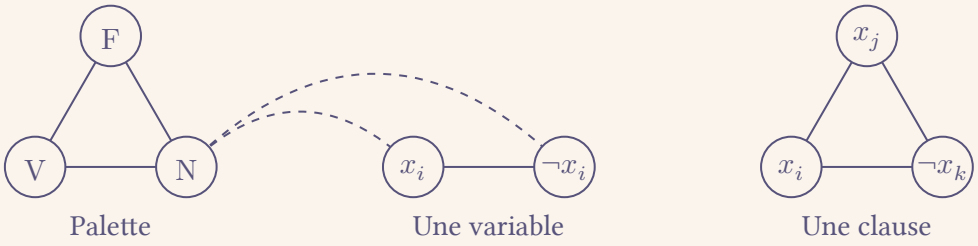


Figure 2. Indications pour la réduction

Q23. Démontrer la **NP**-complétude du problème suivant.

CLIQUE-RECOUVREMENT :
Entrée. Un graphe G et un entier k
Sortie. Peut-on partitionner les sommets en k ensembles tel que chaque partie forme une clique dans G ?

Indication. Réduire depuis 3COLORIABLE. Choisir $k = 3$.

Q24. On considère le problème de planification suivant. On nous donne une liste de DSs[6] $F_1, \dots, F_k$ à planifier, et une liste d'étudiants $\acute{E}_1, \dots, \acute{E}_\ell$. Chaque étudiant passe un sous-ensemble connu des DSs. Vous devez planifier chaque DS sur un créneau tel qu'aucun étudiant doit passer deux DSs en même temps sur le même créneau. Formuler clairement ce problème (sous forme entrée/sortie), et démontrer qu'il est **NP**-complet.

VI. *Masterclass*

On va démontrer la **NP**-complétude du problème suivant *sans* utiliser une réduction depuis un problème **NP**-complet. Même s'il ressemble au problème ARRÊT, ce problème est totalement décidable :

ACCEPTE-RAPIDEMENT :
Entrée. Une machine $\mathscr{M}$ et un entier t
Sortie. Est-ce que $\mathscr{M}$ accepte une entrée en moins de t étapes élémentaires ?

Cette démonstration est là uniquement pour montrer qu'une réduction à un problème **NP**-difficile n'est pas nécessaire pour démontrer la **NP**-difficulté.

Pour montrer que le problème ACCEPTE-RAPIDEMENT est **NP**-complet, il faut montrer qu'il est dans **NP** et qu'il est **NP**-difficile.

- Le problème ACCEPTE-RAPIDEMENT est dans **NP**. En effet, pour une entrée $(\mathscr{M}, t)$, on définit le certificat comme une entrée (un mot w de $\Sigma^{\leq t}$[7]) sur lequel la machine $\mathscr{M}$ termine en moins de t étapes élémentaires. On peut considérer que c'est un mot de $\Sigma^{\leq t}$ car l'entrée w doit être lue, ce qui demande au moins $|w|$ étapes élémentaires. La vérification prend t étapes élémentaires, en simulant $\mathscr{M}$ sur w, ce qui est polynomial en t. On en déduit que le problème ACCEPTE-RAPIDEMENT est dans **NP**.
- Montrons que ACCEPTE-RAPIDEMENT est **NP**-difficile. Pour cela, on passe par la définition de **NP**-difficulté. Soit R un problème quelconque dans **NP**. Montrons que R se réduit en temps polynomial à ACCEPTE-RAPIDEMENT. Comme R est un problème de **NP**, il existe une machine $\mathscr{V}$ qui vérifie

[6] Ceci correspond plus à des partiels en université qu'à des DSs en prépa : les DS ne sont pas les mêmes pour tous, et dépendent des cours choisis.

un certificat de R pour une entrée donnée en temps polynomial. Comme la vérification par la machine $\mathscr{V}$ a lieu en temps polynomial, soit $q(X)$ un polynôme bornant le temps de calcul en fonction de la taille de l'entrée $|w|$. On définit la fonction de réduction :

$$f : w \longmapsto (\mathscr{M}_w, q(|w|))$$

où la machine $\mathscr{M}_w$ prend en entrée un certificat c et simule $\mathscr{V}(w, c)$. Les machines $\mathscr{M}_w$ terminent en temps polynomial par rapport à l'entrée : le temps de calcul est borné par $q(|w|)$. Cette réduction est donc bien polynomiale. De plus,

$$\begin{aligned} w \in R^+ &\iff \exists c \text{ un certificat}, (w, c) \xrightarrow[\mathscr{V}]{} \textbf{\textit{V}} \\ &\iff \begin{array}{l} \mathscr{M}_w \text{ accepte une entrée } c \\ \text{de taille inférieure à } q(|w|) \\ \text{avec } q(|w|) \text{ opérations élémentaires} \end{array} \\ &\iff (\mathscr{M}_w, q(|w|)) \in \textsc{accepte-rapidement}^+ \end{aligned}$$

On a donc démontré que $R \preccurlyeq_{\mathrm{p}} \textsc{accepte-rapidement}$ et ce, quel que soit $R \in \mathbf{NP}$. On en déduit donc que ACCEPTE-RAPIDEMENT est **NP**-difficile.

Ceci permet de démontrer la **NP**-complétude de ACCEPTE-RAPIDEMENT.

Pourquoi avoir placer cette démonstration ici ? Démontrer qu'un problème est **NP**-difficile se fait *majoritairement* par réduction à un autre problème **NP**-difficile. L'intérêt de cette démonstration, c'est de montrer qu'on peut également démontrer de la **NP**-difficulté sans passer par des réductions à d'autres problèmes. Au programme de MPI$^{(\varepsilon|\star)}$, les réductions polynomiales habituelles sont largement privilégiées.

Avec le résultat de **NP**-difficulté sur ACCEPTE-RAPIDEMENT, on peut prouver la **NP**-difficulté du problème SAT.

[7] On note $\Sigma^{\leq t}$ l'ensemble $\cup\,(i \leq t)\Sigma^i$ des mots de taille inférieure ou égale à t.