TD 10 – NP-complétude

1 Variations autour de SAT

Une *formule booléenne* φ est construite à partir d'un ensemble de variables $x_1, \ldots, x_n$ (pour $n \in \mathbb{N}_{>0}$) et des connecteurs logiques $\wedge$ (ET), $\vee$ (OU) et $\neg$ (NON). Elle est satisfaisable lorsqu'il existe une assignation des variables $x_1, \ldots, x_n$ à vrai ($\top$) ou faux ($\bot$) telle que φ s'évalue à vrai en utilisant les règles habituelles d'évaluation des connecteurs logiques.

On rappelle que

$$\mathrm{SAT} = \{\langle\varphi\rangle \mid \varphi \text{ est une formule booléenne satisfaisable}\}$$

est NP-complet.

Une formule *CNF* (pour *Conjunctive Normal Form*) est une formule booléenne φ sur un ensemble de variables $x_1, \ldots, x_n$ (pour $n \in \mathbb{N}_{>0}$) de la forme $\bigwedge_{j=1}^{m} C_j$ avec $m \in \mathbb{N}_{>0}$, où chaque C_j est une *clause*, c'est-à-dire une disjonction de *littéraux* (un littéral étant soit une variable x_i, soit la négation d'une variable $\neg x_i$) de la forme $C_j = \bigvee_i v_i^j$ où $v_i^j = x_l$ ou $v_i^j = \neg x_l$ pour un certain l.

On pose

$$\mathrm{CNF\text{-}SAT} = \{\langle\varphi\rangle \mid \varphi \text{ est une formule CNF satisfaisable}\}\ .$$

Exercice 1. Montrer que CNF - SAT est NP-complet.

Solution 1. On traduit récursivement les formules avec t en CNF. Tout d'abord on pousse les négations en bas en utilisant les formules de De Morgan puis $t(\varphi \wedge \psi) = t(\varphi) \wedge t(\psi)$; et pour $t(\varphi \vee \psi)$ on ajoute x à chaque clause de $t(\varphi)$ et $\neg x$ à chaque clause de $t(\psi)$ et la conjonction des deux nous donne ce que l'on veut.

Cette réduction est polynomiale car si on prend l'arbre de la formule, le nombre de clauses est le nombre de feuilles tandis que le nombre de littéraux dans une clause est le nombre de $\vee$ rencontrés plus un.

Le problème k - SAT est le problème CNF - SAT où l'on impose que chaque clause contienne au plus k littéraux.

Pour tout $k \in \mathbb{N}_{>0}$, on pose

$$\mathrm{k\text{-}SAT} = \{\langle\varphi\rangle \mid \varphi \text{ est une formule CNF satisfaisable où chaque clause a au plus } k \text{ littéraux}\}\ .$$

Exercice 2. Montrer que 3 - SAT est NP-complet.

Solution 2. Étant donné une clause $C = l_1 \vee l_2 \vee l_3 \vee \cdots \vee l_k$ où $k \in \mathbb{N}, k \geq 4$, on observe que C est satisfaisable si et seulement si $(l_1 \vee l_2 \vee v) \wedge (\neg v \vee l_3 \vee \cdots \vee l_k)$ est satisfaisable, où v est une nouvelle variable n'apparaissant pas dans C. En appliquant ce procédé itérativement (en faisant attention à prendre de nouvelles variables à chaque étape), on peut transformer C en une formule CNF où chaque clause contient au plus 3 littéraux, qui est satisfaisable si et seulement si C l'est.

Pour réduire CNF - SAT à 3 - SAT, il suffit d'appliquer cette transformation à chaque clause de la formule CNF considérée. Il est direct de voir que cette réduction se fait en temps polynomial.

Exercice 3. Montrer que 2 - SAT $\in$ P.

Solution 3. Soit φ une formule CNF où chaque clause a au plus 2 littéraux. Si $x_1, \ldots, x_n$ pour $n \in \mathbb{N}_{>0}$ est l'ensemble de variables de φ, construisons le graphe orienté G dont les sommets sont les littéraux $x_1, \neg x_1, \ldots, x_n, \neg x_n$ et tel qu'il y a un arc entre v et $\neg w$ lorsqu'il existe une clause $v \vee w$ dans φ. Il s'avère que φ est satisfaisable si et seulement si aucune variable et sa négation appartiennent à une même composante fortement connexe de G.

Le sens direct est facile à montrer par contraposée. Pour ce qui est du sens indirect, on donne un algorithme itératif pour construire une assignation telle que φ s'évalue à vrai. On observe que dans la décomposition de G en composantes fortement connexes, on a un ordre partiel $\preceq$ sur ces composantes donné par $K \preceq K'$ si et seulement s'il existe un sommet x dans K et un sommet x' dans K' tel qu'il y ait un chemin de x à x' dans G. On construit itérativement une assignation des variables $x_1, \ldots, x_n$ en maintenant l'invariant suivant : à chaque étape, l'assignation partielle obtenue sur les variables d'indice $I \subseteq [\![1, n]\!]$ est telle que φ est satisfaisable si et seulement si $\varphi|_{[\![1,n]\!]\setminus I}$, c'est-à-dire φ restreinte aux clauses contenant uniquement des variables d'indice non dans I, est satisfaisable. Étant donné une assignation partielle obtenue pour les variables d'indice $I \subseteq [\![1, n]\!]$ vérifiant la propriété précédente, on considère $G_{[\![1,n]\!]\setminus I}$ le sous-graphe induit de G limité aux sommets dont l'indice n'est pas dans I. Il existe au moins une composante fortement connexe K_0 minimale pour $\preceq$; par construction de G et par hypothèse, il existe donc une composante fortement connexe K'_0 qui contient exactement tous les littéraux correspondant aux négations des littéraux de K_0 qui est maximale pour $\preceq$. Si J est le sous-ensemble de $[\![1, n]\!]$ des indices des littéraux de K_0, on peut donc assigner les variables d'indice J de manière à ce que tous les littéraux de K_0 soient faux (et donc tous ceux de K'_0 soient vrais), ce qui implique que $\varphi|_{[\![1,n]\!]\setminus I}$ est satisfaisable si et seulement si $\varphi|_{[\![1,n]\!]\setminus J}$ est satisfaisable.

Une formule *DNF* (pour *Disjunctive Normal Form*) est une formule booléenne φ sur un ensemble de variables $x_1, \ldots, x_n$ (pour $n \in \mathbb{N}_{>0}$) de la forme $\bigvee_{j=1}^{m} M_j$ avec $m \in \mathbb{N}_{>0}$, où chaque M_j est un *monôme*, c'est-à-dire une conjonction de littéraux de la forme $M_j = \bigwedge_i v_i^j$ où $v_i^j = x_l$ ou $v_i^j = \neg x_l$ pour un certain l.

On pose

$$\text{DNF-SAT} = \{\langle\varphi\rangle \mid \varphi \text{ est une formule DNF satisfaisable}\} .$$

Exercice 4. Dans lesquelles des classes NP, coNP et P le langage DNF-SAT se trouve-t-il ?

Solution 4. Le langage DNF-SAT est dans ces trois classes. En effet, pour satisfaire une formule DNF, il suffit de satisfaire l'un de ses monômes, donc on traite monôme par monôme, sachant qu'un monôme est satisfaisable si et seulement s'il ne comporte pas une variable et sa négation.

2 Quelques problèmes sur les graphes

Soit $G = (V, E)$ un graphe non orienté.

Une *clique de taille* $k \in \mathbb{N}_{>0}$ dans G est un sous-graphe induit (S, E') de G (où $E' = \big\{\{x, y\} \in E \bigm| x, y \in S\big\}$) tel que toute paire de sommets qu'il contient est adjacente, c'est-à-dire que pour tous $x, y \in S$, on a $\{x, y\} \in E$.

Un *ensemble indépendant de taille* $k \in \mathbb{N}_{>0}$ dans G est un sous-ensemble de sommets $S \subseteq V$ de cardinal k tel qu'aucune paire de sommets qu'il contient est adjacente, c'est-à-dire que pour tous $x, y \in S$, on a $\{x, y\} \notin E$.

Une *couverture par sommets de taille* $k \in \mathbb{N}_{>0}$ dans G est un sous-ensemble de sommets $S \subseteq V$ de cardinal k tel que toute arête de G est incidente à un sommet de S, c'est-à-dire que pour tout $\{x, y\} \in E$, on a $x \in S$ ou $y \in S$.

Un *ensemble dominant de taille* $k \in \mathbb{N}_{>0}$ dans G est un sous-ensemble de sommets $S \subseteq V$ de cardinal k tel que tout sommet de G appartient à S ou est adjacent à un sommet de S, c'est-à-dire que pour tout $x \in V$, on a $x \in S$ ou il existe $y \in S$ tel que $\{x, y\} \in E$.

Exercice 5. Montrer que les langages suivants sont NP-complets.

1. $\text{CLIQUE} = \{\langle G, k\rangle \mid G$ est un graphe non orienté contenant une clique de taille au moins $k\}$.
2. $\text{INDEPENDENT-SET} = \left\{\langle G, k\rangle \;\middle|\; \begin{array}{l} G \text{ est un graphe non orienté contenant un ensemble} \\ \text{indépendant de taille au moins } k \end{array}\right\}.$

3. $\text{VERTEX-COVER} = \left\{ \langle G, k\rangle \;\middle|\; \begin{array}{l} G \text{ est un graphe non orienté contenant une converture par} \\ \text{sommets de taille au plus } k \end{array} \right\}$.
4. $\text{DOMINATING-SET} = \left\{ \langle G, k\rangle \;\middle|\; \begin{array}{l} G \text{ est un graphe non orienté contenant un ensemble} \\ \text{dominant de taille au plus } k \end{array} \right\}$.

Solution 5. Dans chacun des cas, on donne une réduction polynomiale de 3 - SAT au langage considéré.

À tout mot qui n'est pas un encodage valide d'une formule CNF avec au plus 3 littéraux dans chaque clause, on associe simplement un mot qui n'est pas un encodage valide d'un couple composé d'un graphe non orienté et d'un entier.

Autrement, étant donné une formule CNF φ sur l'ensemble de variables $x_1, \dots, x_n$ avec $n \in \mathbb{N}_{>0}$ de la forme $\varphi = \bigwedge_{j=1}^{m} C_j$ avec $m \in \mathbb{N}_{>0}$ et chaque clause C_j pour $j \in [\![1, m]\!]$ de la forme $l_{j,1} \vee l_{j,2} \vee l_{j,3}$ où chaque littéral correspond à une variable différente (ce à quoi on peut toujours se ramener sans perte de généralité), on associe les instances suivantes.

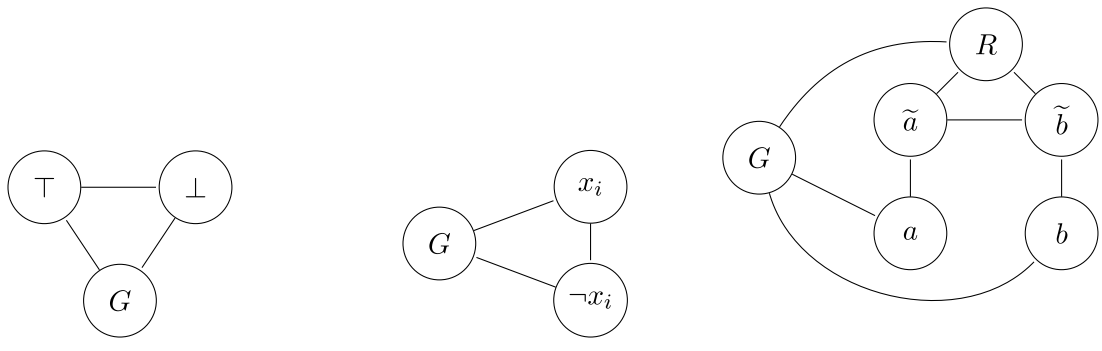
1. $\langle G, m\rangle$ avec $G = (V, E)$ où $V = [\![1, m]\!] \times [\![1, 3]\!]$ et
$$E = \left\{ \{(i, j), (i', j')\} \subseteq [\![1, m]\!] \times [\![1, 3]\!] \;\middle|\; i \neq i' \wedge l_{i,j} \neq \overline{l_{i',j'}} \right\} .$$
2. $\langle G, m\rangle$ avec $G = (V, E)$ où $V = [\![1, m]\!] \times [\![1, 3]\!]$ et
$$\begin{aligned} E =& \left\{ \{(i, j), (i', j')\} \subseteq [\![1, m]\!] \times [\![1, 3]\!] \;\middle|\; i \neq i' \wedge l_{i,j} = \overline{l_{i',j'}} \right\} \cup \\ & \left\{ \{(i, j), (i, j')\} \subseteq [\![1, m]\!] \times [\![1, 3]\!] \;\middle|\; j \neq j' \right\} . \end{aligned}$$
3. $\langle G, 2m\rangle$ avec $G = (V, E)$ où $V = [\![1, m]\!] \times [\![1, 3]\!]$ et
$$\begin{aligned} E =& \left\{ \{(i, j), (i', j')\} \subseteq [\![1, m]\!] \times [\![1, 3]\!] \;\middle|\; i \neq i' \wedge l_{i,j} = \overline{l_{i',j'}} \right\} \cup \\ & \left\{ \{(i, j), (i, j')\} \subseteq [\![1, m]\!] \times [\![1, 3]\!] \;\middle|\; j \neq j' \right\} . \end{aligned}$$
4. $\langle G, n\rangle$ avec $G = (V, E)$ où $V = \{x_1, \dots, x_n, \neg x_1, \dots, \neg x_n, b_1, \dots, b_n, C_1, \dots, C_m\}$ et
$$E = \bigcup_{i=1}^{n} \left\{ \{x_i, \neg x_i\}, \{x_i, b_i\}, \{\neg x_i, b_i\} \right\} \cup \bigcup_{j=1}^{m} \left\{ \{l_{j,1}, C_j\}, \{l_{j,2}, C_j\}, \{l_{j,3}, C_j\} \right\} .$$

3 Coloriage

Exercice 6. Montrer que pour tout $k \in \mathbb{N}_{>0}$, si (k + 1) - COL $\in \mathsf{P}$, alors k - COL $\in \mathsf{P}$.

Solution 6. Si l'on sait résoudre la $k + 1$-coloriabilité en temps polynomial, alors il suffit d'ajouter un sommet relié à tous les autres et $k + 1$-colorier le graphe obtenu. En retirant le sommet ajouté, on obtient une k-coloration du graphe.

Nous allons maintenant montrer que 3 - SAT se réduit à 3 - COL. Pour cela nous allons introduire plusieurs gadgets.

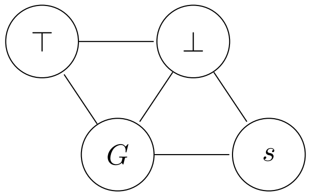


(a) Gadget de couleur (b) Gadget de littéraux (c) Gadget OR

Gadget de couleurs Dans notre graphe, les trois couleurs vont représenter soit vrai ($\top$), faux ($\perp$) ou autre (G pour Ground). Pour encoder ces couleurs, notre premier gadget est d'inclure le graphe de la figure 1a. Nous brancherons ensuite souvent des sommets à G ou $\perp$ pour interdire ces couleurs, il s'agira toujours de ces mêmes sommets G et $\perp$ (le gadget n'est présent qu'une seule fois même si ses sommets sont utilisés dans les autres gadgets).

Exercice 7. Trouver un gadget qui impose qu'un sommet soit colorié de la même couleur que $\top$.

Solution 7. Pour s un sommet, on utilise le gadget suivant :



.

Gadget de littéraux Pour chaque variable x_i nous allons créer deux sommets x_i et $\neg x_i$ que nous allons brancher à G et entre eux selon le schéma de la figure 1b.

Gadget OR Supposons que a et b représentent des valeurs booléennes (ce que l'on force en les branchant à G), le gadget OR est celui de la figure 1c (le G est celui du premier gadget).

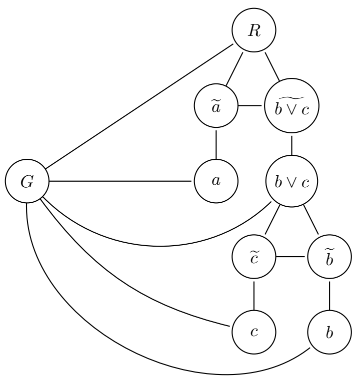
Exercice 8. Montrer que dans tout coloriage du gadget, la couleur de R est la même que celle de a ou la même que celle de b et que, si ces couleurs sont différentes, alors R peut avoir l'une ou l'autre des couleurs, selon le coloriage.

Solution 8. Soit a et b sont de la même couleur et alors $\widetilde{a}$ et $\widetilde{b}$ doivent être coloriés en les deux couleurs complémentaires, donc R est forcément colorié de cette couleur commune.

Soit a et b sont coloriés en $\top$ et $\perp$ tandis que R est forcément colorié en l'une de ces couleurs, au choix, en fonction du coloriage de $\widetilde{a}$ et $\widetilde{b}$.

Exercice 9. Trouver un gadget qui réalise le OR entre trois valeurs booléennes a, b, c (représentées par des sommets coloriés de la même couleur que $\top$ ou $\perp$).

Solution 9.



.

Preuve finale Nous allons maintenant tout rassembler pour construire la réduction de 3-SAT à 3-COL.

Exercice 10. Trouver un gadget pour vérifier chacune des clauses.

Solution 10. Pour chaque clause $v_1 \vee v_2 \vee v_3$ on utilise le 3-OR avec $a = v_1$, $b = v_2$, $c = v_3$ (pris directement sur les gagdets littéraux) puis on relie R à G et $\bot$. Ce gadget ne peut être colorié que si un des v_i est colorié avec la couleur de $\top$.

Exercice 11. Donner une réduction polynomiale de 3 - SAT à 3 - COL.

Solution 11. Nos gadgets littéraux introduisent 2 sommets et 3 arêtes par variable tandis que nos gadgets clauses introduisent 6 sommets et 12 arêtes. Dans tous les cas la formule est transformée en graphe de taille linéaire et a fortiori polynomial.

4 Chemin hamiltonien

Dans un graphe G, qu'il soit orienté ou non orienté, un chemin de s à t dans G est dit *hamiltonien* lorsque le chemin passe exactement une fois par chaque sommet du graphe.

On pose

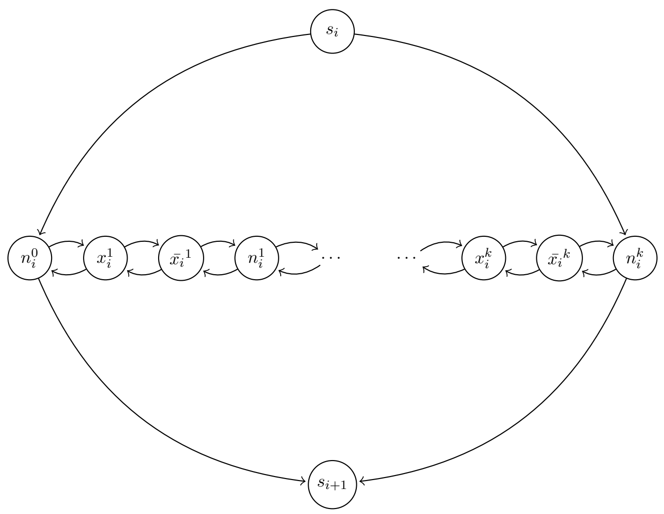
$$\text{HAMPATH} = \{\langle G, s, t\rangle \mid G \text{ est un graphe orienté contenant un chemin hamiltonien de } s \text{ à } t\}\ .$$

Exercice 12. Montrer que HAMPATH est NP-complet.

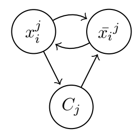
Suggestion : partir d'une formule CNF à k clauses $C_1, \ldots, C_k$ et l variables $x_1, \ldots, x_l$ et construire un graphe qui contient :

— *pour x_i une variable, trois sommets x_i^j, $\bar{x}_i^j$ et n_i^j pour chaque clause C_j et deux sommets s_i et n_i^0 ;*
— *pour C_j une clause, un sommet C_j ;*
— *un sommet s_{l+1} ;*

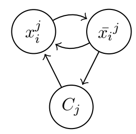
et utilise les gadgets suivants :



(a) Gadget de variable



(b) Gadget pour x_i utilisé dans C_j



(c) Gadget pour $\bar{x}_i$ utilisé dans C_j.

Solution 12. Considérons le graphe avec le sommet s_{l+1} et $3k+2$ sommets par variable i : $v_0^i, \ldots, v_{3k}^i$ et s_i. On relie bilatéralement les v_j^i à v_{j-1}^i et v_{j+1}^i quand ils existent. On ajoute $s_i \to v_0^i$, $v_0^i \to s_{i+1}$, $s_i \to v_{3k}^i$ et $v_{3k}^i \to s_{i+1}$. Pour chaque clause C_j, on ajoute un sommet C_j et on branche pour chaque x_i de la clause les arcs $v_{3(j-1)+1}^i \to C_j$ et $C_j \to v_{3(j-1)+2}^i$, tandis que pour chaque $\bar{x}_i$ dans C_j on ajoute $v_{3(j-1)+2}^i \to C_j$ et $C_j \to v_{3(j-1)+1}^i$. On prétend que la formule CNF est satisfaisable si et seulement s'il existe un chemin hamiltonien de s_1 à s_{l+1}.

Ceci donne une réduction polynomiale de CNF - SAT à HAMPATH.

On pose à présent

$$\text{UHAMPATH} = \{\langle G, s, t\rangle \mid G \text{ est un graphe non orienté contenant un chemin hamiltonien de } s \text{ à } t\}\ .$$

Exercice 13. Montrer que UHAMPATH est également NP-complet.

Solution 13. On donne une réduction polynomiale de HAMPATH à UHAMPATH.

Étant donné un graphe orienté $G = (V, A)$ et $s, t \in V$, on associe à $\langle G, s, t\rangle$ l'instance $\langle G', s_{\text{out}}, t_{\text{in}}\rangle$, où $G' = (V', E)$ est un graphe non orienté avec

$$V' = \{s_{\text{out}}, t_{\text{in}}\} \cup \bigcup_{x \in V \setminus \{s,t\}} \{x_{\text{in}}, x_{\text{mid}}, x_{\text{out}}\}$$

et

$$E = \big\{\{u_{\text{out}}, v_{\text{in}}\} \mid (u, v) \in A \setminus (V \times \{s\} \cup \{t\} \times V)\big\} \cup \bigcup_{x \in V \setminus \{s,t\}} \big\{\{x_{\text{in}}, x_{\text{mid}}\}, \{x_{\text{mid}}, x_{\text{out}}\}\big\} .$$