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 ($\perp$) 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. On regarde le graphe orienté dont les sommets sont les littéraux possibles et on met un arc entre v et $\neg w$ s'il existe une clause $v \vee w$. La formule est satisfaisable si et seulement si aucune variable et sa négation appartiennent à une même composante fortement connexe de ce graphe.

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. CLIQUE $= \{\langle G, k\rangle \mid G$ est un graphe non orienté contenant une clique de taille au moins $k\}$.
2. 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. VERTEX - COVER $= \left\{\langle G, k\rangle \;\middle|\; \begin{array}{l} G \text{ est un graphe non orienté contenant une converture de} \\ \text{sommets de taille au plus } k \end{array}\right\}$.
4. 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 $3\text{-}SAT$ 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 $x_1, \ldots, x_n$ 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

$$E = \big\{\{(i, j), (i', j')\} \subseteq [\![1, m]\!] \times [\![1, 3]\!] \bigm| i \neq i' \wedge l_{i,j} \neq \overline{l_{i',j'}}\big\} .$$

2. $\langle G, m\rangle$ avec $G = (V, E)$ où $V = [\![1, m]\!] \times [\![1, 3]\!]$ et

$$\begin{aligned} E = &\big\{\{(i,j),(i',j')\} \subseteq [\![1,m]\!] \times [\![1,3]\!] \;\big|\; i \neq i' \wedge l_{i,j} = \overline{l_{i',j'}}\big\} \cup \\ &\big\{\{(i,j),(i,j')\} \subseteq [\![1,m]\!] \times [\![1,3]\!] \;\big|\; j \neq j'\big\} \ . \end{aligned}$$

3. $\langle G, 2m\rangle$ avec $G = (V, E)$ où $V = [\![1, m]\!] \times [\![1, 3]\!]$ et

$$\begin{aligned} E = &\big\{\{(i,j),(i',j')\} \subseteq [\![1,m]\!] \times [\![1,3]\!] \;\big|\; i \neq i' \wedge l_{i,j} = \overline{l_{i',j'}}\big\} \cup \\ &\big\{\{(i,j),(i,j')\} \subseteq [\![1,m]\!] \times [\![1,3]\!] \;\big|\; j \neq j'\big\} \ . \end{aligned}$$

4. $\langle G, n\rangle$ avec $G = (V, E)$ où $V = \{x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n, b_1, \ldots, b_n, C_1, \ldots, C_m\}$ et

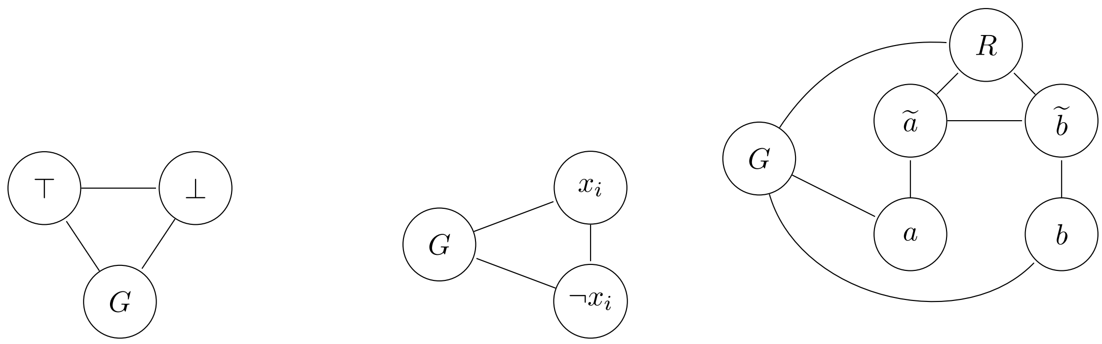
$$E = \bigcup_{i=1}^{n} \big\{\{x_i, \neg x_i\}, \{x_i, b_i\}, \{\neg x_i, b_i\}\big\} \cup \bigcup_{j=1}^{m} \big\{\{l_{j,1}, C_j\}, \{l_{j,2}, C_j\}, \{l_{j,3}, C_j\}\big\} \ .$$

3 Coloriage

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

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

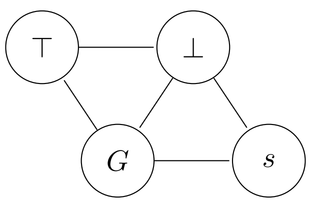


(a) Gadget de couleur (b) Gadget de littéraux (c) Gadget OR

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).

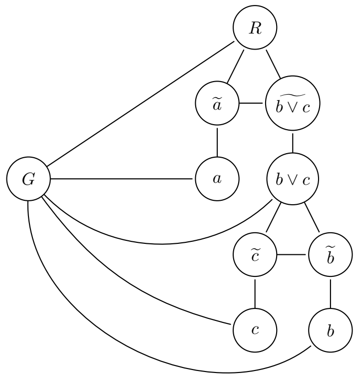
Exercice 8. Montrer que dans tout coloriage du gadget, la couleur de R est la même que celle de a ou la même que celle de b. Et montrer que si on limite le graphe à ce gadget, alors R peut être colorié par la couleur de a ou par celle de b.

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 lorsqu'on limite le graphe à ce gadget, puisque R est lié à G.

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 $a = v_1$, $b = v_2$, $c = v_3$ puis on relit R à G et $\perp$. 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

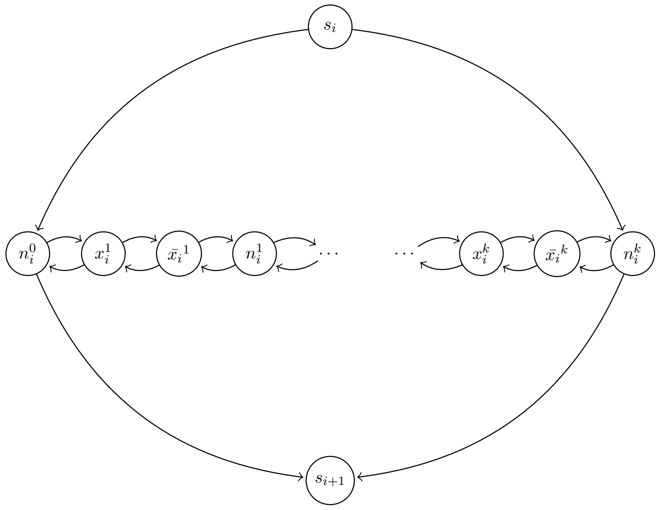
Dans un graphe G, qu'il soit orienté ou non orienté, un chemin de s à t dans G est dit *hamiltonien* lorsque le chemin passe exactement une fois par chaque sommet du graphe.

On pose

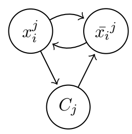
$$\text{HAMPATH} = \{\langle G, s, t\rangle \mid G \text{ est un graphe orienté contenant un chemin hamiltonien de } s \text{ à } t\}\ .$$

Exercice 12. Montrer que HAMPATH est NP-complet.

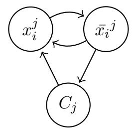
Suggestion : partir d'une formule CNF à k clauses $C_1, \ldots, C_k$ et l variables $x_1, \ldots, x_l$ et construire un graphe qui contient, pour x_i une variable et C_j une clause, trois sommets x_i^j, $\bar{x}_i^j$ et n_j^i ; deux sommets s_i et n_i^0 ; un sommet C_j plus un sommet s_{l+1} et utilise 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 $3l+2$ sommets par variable i ($v_0^i \ldots v_{3k}^i$) et s_i. On relit bilatéralement les v_j^i à v_{j-1}^i et v_{j+1}^i quand ils existent. On ajoute $s_i \to v_0^i$, $v_0^i \to s_{i+1}$, $s_i \to v_{3k}$ et $v_{3k} \to s_{i+1}$. Pour chaque clause j on ajoute un sommet C_j et on branche pour chaque x_i de la clause les arcs $v_{3j+1}^i \to C_j$ et $C_j \to v_{3j+2}^i$ tandis que pour chaque $\bar{x}_i$ dans C_j on ajoute $v_{3j+2}^i \to C_j$ et $v_{3j+1}^i \to C_j$. La réduction polynomiale consiste maintenant à chercher un chemin hamiltonien de s_1 à s_{k+1}.

Soit un chemin hamiltonien de s_1 à s_{k+1}. Nous allons montrer par induction sur i puis j que les sommets v_j^i apparaissent dans le chemin par i croissant et que pour un i donné les v_j^i apparaissent par j monotone (soit croissant soit décroissant) et que premier sommet visité après la couche i est s_{i+1}.

Supposons la propriété vraie pour tout $j < i$ avec $i \in \mathbb{N}^*$ (trivialement vraie pour $i = 0$ puisque s_1 est le départ du chemin). s_i n'est suivie que de v_0^i et v_{3k}^i. Les deux cas sont symétriques (mais l'un implique de traiter la couche par j croissant l'autre décroissant). Supposons s_i est suivi de v_0^i. Le sommet suivant de v_0^i ne peut pas être s_{i+1} car v_1^i n'est relié qu'à deux sommets dont v_0^i. Donc le sommet suivant v_0^i est v_1^i. Montrons maintenant la propriété de croissance par induction sur j :

— Elle est vraie pour $j = 1$.

— Soit $1 \leq j < 3k$ et supposons la propriété vraie pour j. Si j est un multiple de 3 alors comme v_j^i n'est reliée qu'à v_{j+1}^i et v_{j-1}^i (qui est visité par induction) le successeur de v_j^i ne peut être que v_{j+1}^i et donc la propriété est vérifiée pour $j+1$.

— Si $j = 3n+1$ alors v_j^i ne peut être suivi que de C_n et de v_{j+1}^i (par induction v_{j-1}^i est déjà visité). Si le sommet suivant est v_{j+1}^i la propriété est obtenue. Sinon nous avons dans le chemin $v_{3n+1}^i \to C_n$. Si le sommet suivant C_n n'est pas v_{j+1}^i alors v_{3n+2}^i ne sera plus relié qu'à un unique sommet non visité v_{3n+3}^i et donc ne pourra pas être pris par un chemin hamiltonien. Donc le sommet suivant C_n est forcément v_{j+1}^i.

— Si $j = 3n+2$ alors les successeurs possibles de v_j^i sont v_{j+1}^i (auquel cas la propriété est claire) et C_n (quand $\bar{x}_i$ apparaît dans C_n). Dans le cas où v_j^i est suivi de C_n alors v_{j+1}^i n'est plus relié qu'à un unique sommet et donc le chemin hamiltonien ne pourra plus être construit, donc v_j^i ne peut être suivi que de v_{j+1}^i.

Dans le cas où v_{3k}^i est le successeur de s_i alors on prouve la même chose dans l'autre sens, les sommets v_j^i peuvent suivis que de v_{j-1}^i sauf dans le cas où $j = 3n+2$ auquel cas on peut avoir $v_j^i \to C_{\lfloor j/3 \rfloor} \to v_{j-1}^i$ (dans le cas $j = 3n+1$ ce n'est pas possible car v_{j-1}^i ne serait plus relié qu'à un unique sommet).

Dans tous les cas les v_j^i sont parcourus en j monotone et en arrivant au bout le seul successeur possible est s_{i+1}.

Nous voulons montrer que si l'on a un chemin hamiltonien alors nous avons une solution à la formule CNF. Pour cela on fixe $x_i = \top$ ssi les v_j^i sont parcourus en ordre croissant. Par la façon dont nous avons encodé les clauses, comme tous les sommets C_j sont visités nous avons $v_{3j+1}^i \to C_j \to v_{3j+2}^i$ quand $x_i = \top$ et x_i apparaît dans C_j et inversement $v_{3j+2}^i \to C_j \to v_{3j+1}^i$ quand $x_i = \bot$ et $\bar{x}_i$ apparaît dans C_j. Donc la formule est bien satisfaite par l'assignement donné.

Montrons maintenant que si l'on a un assignement alors il existe une solution au chemin hamiltonien. Pour cela, on choisit pour chaque clause un de ses représentants qui la satisfait et le chemin est de parcourir chaque couche dans l'ordre croissant quand $x_i = \top$ et décroissant quand $x_i = \bot$. Si x_i est le littéral qui satisfait C_j alors on prend le chemin $v_{3j+1}^i \to C_j \to v_{3j+2}^i$ sinon $v_{3j+1}^i \to v_{3j+2}^i$. Dans le cas où $\bar{x}_i$ satisfait C_j alors on prend le chemin $v_{3j+2}^i \to C_j \to v_{3j+1}^i$ sinon $v_{3j+2}^i \to v_{3j+1}^i$. On voit que tous les v_j^i, les s_i aussi et les C_j sont visités une unique fois.

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. Étant donné une instance du problème du chemin hamiltonien sur le graphe $G = (V, E)$ on triple chaque sommet n en n_{in}, n_{out} et n_{mid}. Pour chaque sommet on ajoute $n_{in} \to n_{mid} \to n_{out}$ et chaque arc (u, v) est transformé en $u_{out} \to v_{in}$. Enfin on enlève les sommets s_{in}, s_{mid}, t_{out} et t_{mid}.

Chaque chemin dans G de s à t est clairement transposable en un chemin dans G' de s_{out} à t_{in}.

Soit un chemin $n_1, \dots n_l$ de G' de s_{out} à t_{in}. Nous allons montrer que le chemin $n_1, \dots, n_k$ est une succession de sommets n_{in}, n_{mid}, n_{out} (le *in*, *mid*, *out* du même sommet) où n_{out} précède n'_{in} quand $n \to n'$ dans par induction sur k sauf le premier qui vaut s_{out} et le dernier qui vaut t_{in}.

Pour $k = 1$, on a $n_1 = s_{out}$ donc c'est clair .

Pour $k < l$ si on suppose que $v_1, \dots v_k$ est une alternance *in*, *mid*, *out*, alors si on a $v_k = n_mid$ c'est que l'on vient de n_{in} et que le seul sommet suivant est n_{out}. Si $v_k = n_in$ alors le noeud n_{mid} n'est pas encore visité (car la seule manière de rentrer dans n_{mid} c'est par n_{in} ou n_{out}) et doit l'être immédiatement sinon il ne sera plus relié qu'à un sommet non visité (n_{out}). Si $v_k = v_{out}$ alors n_{in} et n_{mid} ont été visités et n_{out} n'est relié qu'à des sommets n'_{in} tels que n' suit n dans G.

Au bout du compte, un chemin hamiltonien dans G' a la forme $s_{out}, n_{in}^1, n_{mid}^1, n_{out}^1, \dots, n_{in}^k, n_{mid}^k,$ n_{out}^k, t_{in} et donc l'on peut retrouver un chemin dans le graphe original $s, n^1, \dots, n^k, t$.