# Exercices corrigés sur problèmes NP-complets

Johanne Cohen

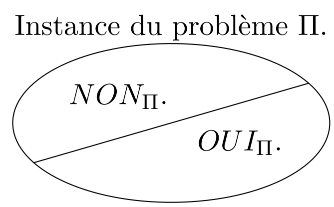
12 septembre 2018

# Table des matières

# Partie 1: Rappel succinct de cours

### Problème de décision.

Un *problème de décision* $\Pi = (D_\Pi, OUI_\Pi)$ correspond
à un ensemble d'instances $D_\Pi$
et à un sous-ensemble $OUI_\Pi \subset D_\Pi$ d'instances positives.




### Les classes de complexité.

— La *classe* $P$ est la classe des problèmes de décision qui admettent un algorithme de complexité polynomiale.
— La *classe* $NP$ est formée des problèmes de décision $\Pi$ qui possèdent un *vérificateur polynomial*.
— Par définition $P \subset NP$.
— Un *vérificateur* $V$ est un algorithme qui prend une information en plus (**certificat**) pour vérifier qu'une instance est positive.

**Exemple :** Graphe Hamiltonien

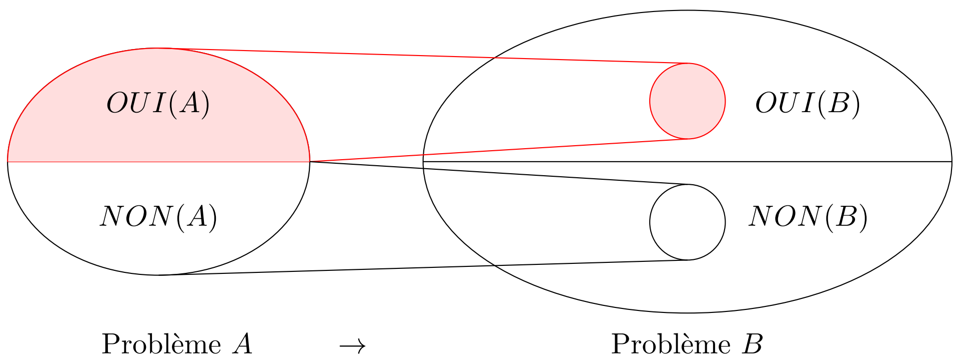
— Problème de décision : **Données** : un graphe non-orienté $G = (V, E)$.
**Question** : $G$ a-il un cycle hamiltonien ?
— Le certificat correspond à une suite $S$ de sommets. Le vérificateur $V$ vérifie que $S$ est un cycle et qu'il transverse chaque sommet une unique fois. $V$ fonctionne bien en temps polynomial.
— Graphe Hamiltonien est dans NP.

### Comment comparer les problèmes

Soient $A$ et $B$ deux problèmes de décision.

Une *réduction de $A$ vers $B$* est une fonction $f : I_A \to I_B$ calculable **en temps polynomial** telle que $w \in Oui(A)$ si et seulement si $f(w) \in Oui(B)$.

On note $A \leq B$ lorsque $A$ se réduit à $B$.



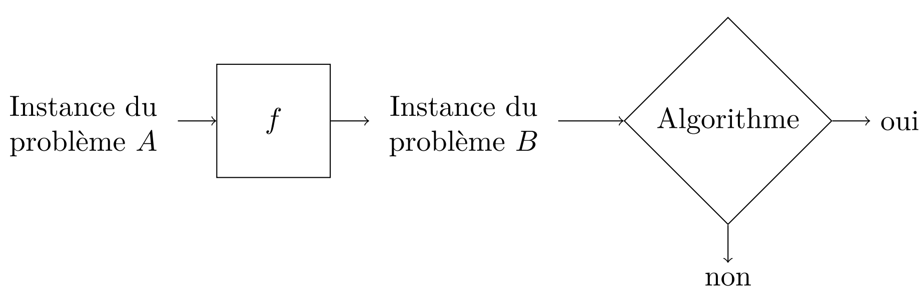

Intuitivement : $A \leq B$ signifie que $A$ est plus facile que $B$.

### Remarque : Si $A \leq B$, et si $B$ est dans $P$, alors $A$ est dans $P$.

Pour cela, il suffit de concevoir un algorithme polynomial qui permet de décider si l'instance de $A$ est positive ou non en temps polynomial :

**Algorithme 1 :** Décider si l'instance $I$ de problème $A$ est positive ou non

**Output :** un booléen $b$
**début**
  Transformer l'instance $I$ en une instance $I'$ de $B$ en utilisant $f$ :
$$I' \leftarrow f(I) \ ;$$
  **si** *$I'$ est une instance positive de $I$ alors* **alors**
    retournez vrai
  **fin**
  retournez faux
**fin**




### Remarque : Si $A \leq B$, et si $A$ n'est pas dans $P$, alors $B$ n'est pas dans $P$.

## Problème NP-difficile

Un problème $A$ est dit $NP$*-difficile* si tout autre problème $B$ de $NP$ est tel que $B \leq A$. Intuitivement : il est plus difficile que tous les problèmes dans la classe.

Un problème $A$ est dit $NP$*-complet* si en plus on a $A \in NP$. Autrement dit : $A$ est $NP$-complet signifie que $A$ est un élément maximum dans $NP$ pour $\leq$.

## Théorème de Cook-Levin

Les problèmes SAT et 3-SAT sont $NP$-complet.

### Fonctions booléennes

Nous allons considérer des fonctions *booléennes*, c'est-à-dire de fonctions de $\phi$ de $\{1, 0\}^n \to \{1, 0\}$.

— Les variables ne peuvent prendre que deux valeurs, vrai (codé par 1) ou faux (codé par 0).

— $\phi$ est composée de variables et d'opérateurs comme négation ($\neg$) la conjonction ( $\wedge$ ) la disjonction ($\vee$), l'implication $\to$

| $u$ | $v$ | $\neg u$ | $(u \wedge v)$ | $u \vee v$ | $u \to v$ |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |

On dira que la fonction $t : U \to \{0, 1\}$ *satisfait* la fonction $\phi$ si la fonction $\phi$ retourne 1 avec les valeurs de $t$ en entrée.

### Problème 3- SAT

— Une *clause* est une fonction de $\phi$ de $\{1, 0\}^n \to \{1, 0\}$ composée de variables et d'opérateurs comme négation ($\neg$) et la disjonction ($\vee$).

Par exemple $C(u_1, u_2, u_3) = (u_1 \vee u_2 \vee \neg u_3)$ est une clause.

— Le *problème* 3-SAT est défini de la façon suivante

**Données** :

un ensemble $U$ de variables $\{u_1, u_2, \ldots, u_n\}$

et une formule logique $\phi = C_1 \wedge \cdots \wedge C_\ell$ des clauses de 3 variables

**Question** : Existe-t-il une fonction $t : U \to \{0, 1\}$ telle que $t$ satisfait $\phi$ ?

**Exemple d'instance pour le 3-SAT** Soit $I$ une instance du problème 3-SAT

— $U = \{u_1, u_2, u_3, u_4\}$ de variables

— $\phi(U) = (u_1 \vee \neg u_2 \vee u_3) \wedge (\neg u_1 \vee \neg u_3 \vee u_4) \wedge (u_2 \vee \neg u_3 \vee \neg u_4)$,

**Remarque :** $3 - SAT$ est dans NP car
- Certificat $t$ :
  - $t = x_1 x_2 \cdots x_n \in \{0,1\}^n$ donne la liste de valeurs de chaque variable.
- Vérification :
  - Vérifier que $t$ rend la formule $F$ vraie se fait bien en un temps polynomial en la taille de $F$.

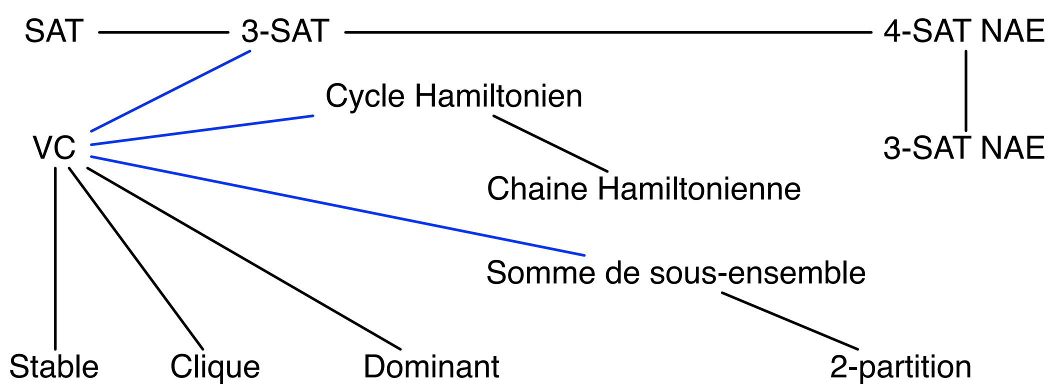
### Comment prouver qu'un problème est NP-complet

Pour prouver la $NP$-complétude d'un problème $A$, il suffit de prouver :

1. qu'il admet un vérificateur polynomial ;
2. que $B \leq A$ avec un problème connu $NP$-complet $B$

Pourquoi ?
- Si $A$ admet un vérificateur polynomial, cela permet de garantir que $A \in NP$,
- Si $B \leq A$ avec un problème connu $NP$-complet $B$ cela permet que on a $C \leq A$ pour tout problème $C \in NP$
  - on a $C \leq B$ comme $B$ est $NP$-complet,
  - et $C \leq A$ puisque $B \leq A$.

### Une liste de problèmes NP-complets connus



# Partie 2: Enoncés des exercices

# I) Problème de décisions

## a) Graphe eulérien

Le graphe $G$ est *eulérien* si il existe un cycle en empruntant exactement une fois chaque arête du graphe $G$. On rappelle qu'un graphe connexe est eulérien si et seulement si chacun de ses sommets a un degré pair.

**Question 1.1**. Ecrire le problème de décision qui lui est associé et donner la taille de l'instance

**Question 1.2**. Trouver un algorithme polynomial qui détermine si le graphe est eulérien.

## b) Formulation des problèmes de décisions

Mettre sous forme de problème de décision et évaluer la taille de leurs instances.

**Question 2.1**. Problème de savoir s'il existe un chemin entre deux sommets disjoints dans un graphe ;

**Question 2.2**. Problème de connaitre la distance entre deux sommets disjoints dans un graphe ;

**Question 2.3**. Problème de connaitre la longueur de la chaîne maximum dans un graphe pondéré.

## c) Problèmes dans NP ou dans P

Les problèmes suivants sont-ils dans NP, dans P ? Justifier votre réponse.

**Problème** *P1*
**Données** : Un graphe $G = (V, E)$
**Question** : Existe-t-il cycle de longueur égale à $\lfloor \frac{|V|}{2} \rfloor$ ?

**Problème** *P2*
**Données** : Un graphe $G = (V, E)$
**Question** : Existe-t-il cycle de longueur égale à 4 ?

**Problème** *P3*
**Données** : Un graphe $G = (V, E)$, deux sommets $u$ et $v$ distincts de $G$ et un entier $k$.
**Question** : Existe-t-il un simple chemin entre $u$ et $v$ de longueur inférieure ou égale à $k$ ?

**Problème** *P4*
**Données** : Un graphe $G = (V, E)$, et un entier $k$.
**Question** : Existe-t-il un arbre couvrant tous les sommets de $G$ ayant moins de $k$ feuilles ?

# II) Réduction polynomiale.

## a) Propriétés sur l'ordre $\leq$

Soient $A$ et $B$ deux problèmes de décision.

Une *réduction de* $A$ *vers* $B$ est une fonction $f : I_A \to I_B$ calculable **en temps polynomial** telle que $w \in Oui(A)$ si et seulement si $f(w) \in Oui(B)$.

Nous notons $A \leq B$ lorsque $A$ se réduit à $B$.

**Question 4.1**. Montrer que $\leq$ est réflexive ($L \leq L$).

**Question 4.2**. Montrer que $\leq$ est transitive (i.e ; $L_1 \leq L_2$, $L_2 \leq L_3$ impliquent $L_1 \leq L_3$.)

**Question 4.3**. Montrer que $P = NP$ si et seulement si 3-SAT $\in P$.

## b) Réduction

Soient A, B et Q des problèmes de décision. Supposons que A est dans $P$, et que B est $NP$-dur. Dire si les affirmations suivantes sont vraies ou fausses :

1. Si A se réduit polynômialement à Q, alors $Q$ est dans $P$.
2. Si Q se réduit polynômialement à A, alors $Q$ est dans $P$.
3. Si Q se réduit polynômialement à B, alors $Q$ est $NP - dur$.
4. Si B se réduit polynômialement à Q, alors $Q$ est $NP - dur$.

## III) Problèmes de logique

### a) Le problème 2-SAT est dans $P$

Nous allons considérer de cet exercice des fonctions *booléennes*, c'est-à-dire de fonctions de $f$ de $\{1,0\}^n \to \{1,0\}$.

— Les variables ne peuvent prendre que deux valeurs, vrai (codé par 1) ou faux (codé par 0).
— La fonction booléenne $f$ est composée de variables et d'opérateurs comme négation ($\neg$) la conjonction ( $\wedge$ ) la disjonction ($\vee$), l'implication $\to$.

Voici la table de vérité pour les opérateurs booléens cités ci-dessus :

| $u$ | $v$ | $\neg u$ | $(u \wedge v)$ | $u \vee v$ | $u \to v$ |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |

Considérons une fonction $t : U \to \{0,1\}$. On dira que la fonction $t$ satisfait la fonction $f$ si la fonction $f$ retourne 1 avec les valeurs de $t$ en entrée.

**Question 6.1**. Nous allons considérer la fonction $\phi_1$ : telle que $\{1,0\}^3 \to \{1,0\}$.

$$\phi_1(x,y,z) = (y \vee \neg z) \wedge (\neg y \vee z) \wedge (y \vee \neg x).$$

Donner une fonction $t : U \to \{0,1\}$ telle que $t$ satisfait $\phi_1$

**Question 6.2**. Que se passe-t-il pour $\phi_2$

$$\phi_2(x,y,z) = (y \vee z) \wedge (\neg y \vee z) \wedge (\neg z \vee x) \wedge (\neg z \vee \neg x)?$$

Une clause est une fonction de $f$ de $\{1,0\}^n \to \{1,0\}$ composée de variables et d'opérateurs comme négation ($\neg$) et la disjonction ($\vee$). Par exemple $C(u_2,u_3) = (u_2 \vee \neg u_3)$ est une clause.

Donnons la définition du problème 2-SAT

*Données* : un ensemble $U$ de variables $\{u_1,u_2,\ldots,u_n\}$ et une formule logique $\phi = C_1 \wedge \cdots \wedge C_\ell$ des clauses de 2 littéraux

*Question* : Existe-t-il une fonction $t : U \to \{0,1\}$ telle que $t$ satisfait $\phi$ ?

**Question 6.3**. Montrer que $u \vee v = (\neg u \Rightarrow v) \wedge (\neg v \Rightarrow u)$.

A partir d'une instance $(U,\phi)$ de 2-SAT, nous construisons un graphe orienté $G_\phi = (V,E)$ tel que

— un sommet par un littéral de $U$
— un arc par implication (en transformant chaque clause par deux implications)

**Question 6.4**. Dessiner les graphes $G_{\phi_1}$ et $G_{\phi_2}$

**Question 6.5**. Montrer que si $G_\phi$ possède un chemin entre $u$ et $v$, alors il possède aussi un chemin entre $\neg v$ et $\neg u$

**Question 6.6**. Montrer qu'il existe une variable $u$ de $U$ tel que $G_\phi$ contient un cycle entre $u$ vers $\neg u$ si et seulement si $\phi$ n'est pas satisfiable.

**Question 6.7**. Montrer que un algorithme en temps polynomial peut résoudre le problème 2-SAT.

Rappellons la définition du problème : 3-SAT

*Données* : Un ensemble $U$ de variables, et une formule $F$ sous la forme d'une conjonction de disjonction de 3 littéraux (variables ou leur négation). Formellement : une formule

$$F = C_1 \wedge C_2 \cdots \wedge C_\ell$$

avec

$$C_i = y_{i,1} \vee y_{i,2} \vee y_{i,3},$$

où pour tout $i$, $j$, $y_{i,j}$ est soit $x_k$, soit $\neg x_k$ pour l'un des $x_k$, pour des variables $\{x_1, \cdots, x_n\}$
*Question* :Décider si $F$ est satisfiable : c'est-à-dire décider s'il existe $x_1, \cdots, x_n \in \{0, 1\}$ tel que $F$ s'évalue à vrai pour cette valeur de ses variables $x_1, \cdots, x_n$.

## b) Variante du problème 3-SAT :

**Donnée**: Une formule $F'$ sous la forme d'une conjonction de clauses même si chaque variable apparaisse **au plus soit trois fois** dans $F$ sous sa forme négative ou positive.

**Réponse**: Décider si $F'$ est satisfiable.

**Question 7.1**. Soit $x_1$, $x_2 \dots x_k$, Considérons une formule $\phi$ par l'ensemble de $k$ clauses $(x_i \vee \neg x_{i+1})$ pour $i = 1, \dots k-1$, et de la clause $(x_k \vee \neg x_1)$. Montrer que une assignation $\phi$ est satisfaite si et seulement si pour $i = 1, \dots k-1$ $x_k = x_1$

**Question 7.2**. Prouver que ce problème est dans NP.

**Question 7.3**. Prouver que ce problème est NP-complet à partir d'une réduction du problème $3 - SAT$.

**Question 7.4**. Prouver $3 - SAT$ est NP-complet même si chaque variable apparaisse **exactement trois fois** dans $F$ sous sa forme négative ou positive.

**Question 7.5**. Prouver $3 - SAT$ est NP-complet même si chaque variable apparaisse **deux fois** sous sa forme positive et **une fois** sous sa forme négative dans $F$.

## c) Le problème MAX2SAT est NP-complet.

L'objectif de cet exercice est de prouver que le problème MAX2SAT est NP-complet.

**Donnée**: Une formule $F'$ sous la forme d'une conjonction de clauses d'au plus 2 littéraux, un entier $K$

**Réponse**: Décider s'il existe une assignation $x_1, \cdots, x_n \in \{0, 1\}$ telle que au moins $K$ clauses de $F'$ sont satisfaites. s'évalue à vrai pour cette valeur de ses variables $x_1, \cdots, x_n$.

**Question 8.1**. Soit une clause $C = x \vee y \vee z$. Nous construisons une formule $\phi_C$ sous la forme d'une conjonction de 10 clauses en introduisant un nouveau littéral $p_C$ :

$$\phi_C = \begin{array}{l} (x) \wedge (y) \wedge (z) \wedge (p_C) \\ (\neg x \vee \neg y) \wedge (\neg y \vee \neg z) \wedge (\neg z \vee \neg x) \\ (x \vee \neg p_C) \wedge (y \vee \neg p_C) \wedge (z \vee \neg p_C) \end{array}$$

Déterminer le nombre maximun de clauses satisfaites dans le cas où

1. la clause $C$ est satisfaite

2. la clause $C$ n'est pas satisfaite

**Question 8.2**. Prouver que le problème MAX2SAT est dans NP.

**Question 8.3**. Montrer que le problème Max2SAT est NP-complet.

## d) Le problème $k$-SAT NAE est NP-complet

Considérez le problème suivant :

$k$-SAT NAE
*Données* : un ensemble $U'$ de variables $\{u_1, u_2, \ldots, u_n\}$ et une formule logique $L = C_1 \wedge \cdots \wedge C_\ell$ avec $C_i = (y_{i,1} \vee y_{i,2} \vee \cdots \vee y_{i,k})$ où $y_{i,j}$ est égal soit à l'un des $u_k$ ou soit à l'un des $\neg u_k$
*Question* : Existe-t-il une fonction $t : U' \rightarrow \{0, 1\}$ telle que $t$ satisfait $L$ et telle que les littéraux de chaque clause ne sont pas toutes de la même valeur ?

**Question 9.1**. Montrer que 4-SAT NAE est NP-complet sachant que 3-SAT est NP-complet. *Indication : Introduire une nouvelle variable z et l'insérer dans toutes les clauses*

**Question 9.2**. Montrer que 3-SAT NAE est NP-complet sachant que 4-SAT NAE est NP-complet. 3-SAT est NP-complet. *Indication : Utiliser la technique pour la transformation de SAT en 3-SAT.*

# IV) Différentes Variantes du problème CYCLE HAMILTONIEN

Nous allons supposer que le problème CYCLE HAMILTONIEN est NP-Complet.
CYCLE HAMILTONIEN
*Données* : un graphe non-orienté $G$.
*Question* : $G$ contient-il un cycle hamiltonien ?

## a) Le problème CHAINE HAMILTONIEN est NP-complet

Considérons le problème CHAINE HAMILTONIEN suivant :
CHAINE HAMILTONIENNE
*Données* : un graphe non-orienté $G$, deux sommets $u$ et $v$ distincts de $G$.
*Question* : $G$ contient-il une chaine hamiltonienne entre $u$ et $v$ ?

**Question 10.1**. Montrer que le problème CHAINE HAMILTONIEN est dans NP.

**Question 10.2**. Montrer que le problème CHAINE HAMILTONIEN est NP-complet.

Pour cela, nous allons faire la réduction à partir du problème CYCLE HAMILTONIEN. Soit $\mathcal{I} =< G = (V, E) >$ une instance du problème CYCLE HAMILTONIEN. Maintenant, nous allons transformer cette instance en une instance $\mathcal{I}'$ du problème CHAINE HAMILTONIENNE de la façon suivante : Nous allons construire un graphe $G' = (V', E')$ tel que

— Soit $u$ un sommet arbitraire de $V$
— $V' := V \cup \{v\}$ tel que $v$ est un sommet n'appartenant pas dans $V$
— $E' := E \cup \{(v, \ell) : \ell$ est un voisin de $u$ dans $G\}$

Continuez la preuve.

## b) Le problème CHAINE est NP-complet

Le problème CHAINE est le problème de décision suivant
CHAINE
*Données* : un graphe non-orienté $G$ de $n$ sommets, deux sommets $u$ et $v$ distincts de $G$.
*Question* : $G$ contient-il une chaine de longueur $n/2$ entre $u$ et $v$ ?

**Question 11.1**. Montrer que le problème CHAINE est dans NP.

**Question 11.2**. Montrer que le problème CHAINE est NP-complet.

## c) Chevaliers de la table ronde

Etant donnés $n$ chevaliers, et connaissant toutes les paires de féroces ennemis parmi eux, est-il possible de les placer autour d'une table circulaire de telle sorte qu'aucune paire de féroces ennemis ne soit côte à côte ?

## d) Le problème VOYAGEUR DE COMMERCE est NP-complet

Considérez le problème VOYAGEUR DE COMMERCE :
*Données* : Un graphe complet $G$ muni d'une fonction $d$ de coût positif sur les arêtes, et un entier $k$.

*Question* : Existe-t-il un circuit passant au moins une fois par chaque ville et dont la somme des coûts des arêtes est au plus $k$ ?

**Question 13.1**. Montrer que Voyageur de Commerce est NP-complet même si la fonction poids respecte l'inégalité triangulaire. (C'est-à-dire que si les arêtes vérifient l'inégalité triangulaire, c'est-à-dire si pour tout triplet de sommets $(u, v, w)$, on a $d(u, v) \leq d(u, w) + d(w, v)$.)

**Question 13.2**. Montrer que le problème du voyageur de commerce ayant une métrique respectant l'inégalité triangulaire est dans NP-complet.

# V) Problèmes de graphe

## a) Les problèmes COLORATION DE GRAPHE

Une *coloration* d'un graphe $G$ est une fonction associant à tout sommet $V$ du graphe $G$ un entier représentant une couleur dans $[1, \ldots, |V|]$, telle que deux sommets adjacents n'aient pas la même couleur. Le problème d'optimisation classique est de trouver une coloration avec un nombre de couleurs minimum.

**Question 14.1**. Donner un algorithme polynomial construisant une coloration d'un arbre $T = (V_T, E_T)$ qui utilise seulement 2 couleurs.

**Question 14.2**. Le problème *k-Arbre-coloration* est-il NP-complet ou dans P ?
**Problème** *k-Arbre-coloration*
**Données** : Un arbre $T = (V, E)$ et un entier $k$.
**Question** : Existe-t'il une façon de colorier les sommets de $T$ avec au plus $k$ couleurs de telle sorte qu'aucune arête n'ait les extrémités de la même couleur ?

Définissons le problème *k-coloration*.

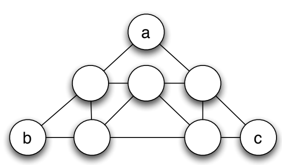
**Problème** *k-coloration*
**Entrée** : Un graphe $G = (V, E)$, un entier $k$
**Question** : Existe-t-il une coloration propre des sommets utilisant moins de $k$ couleurs ?

**Question 14.3**. Le problème 2-coloration est-il NP-complet ou dans P ? Justifier votre réponse.

A partir de maintenant, nous supposerons que le problème *3-coloration* est NP-complet.

**Question 14.4**. Montrer que le problème *4-coloration* est NP-complet.

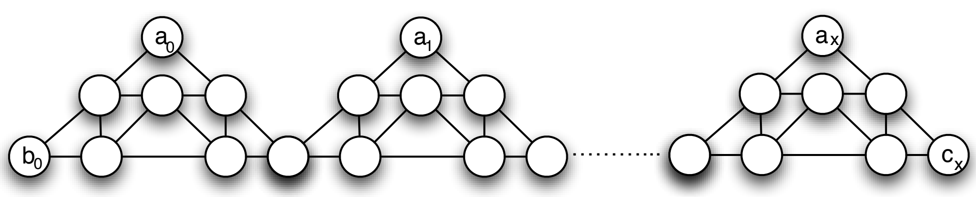
Considérons le graphe $H$ :




**Question 14.5**. Montrer que ce graphe ne possède pas de coloration propre du graphe utilisant au plus 2 couleurs.

**Question 14.6**. Montrer que dans toute coloration propre du graphe utilisant au plus 3 couleurs, $a$, $b$ et $c$ ont la même couleur.

Soit $x$ un entier strictement positif. Considérons le graphe $H_x$ suivant :



**Question 14.7**. Montrer que dans toute coloration propre du graphe $H$ utilisant au plus 3 couleurs, $(a_i)_{0\leq i<x}$, $(b_i)_{0\leq i\leq x}$, sont de même couleur.

**Question 14.8**. Montrer que le problème COLOR suivant est NP-complet :

**Entrée** : Un graphe $G' = (V', E')$ dont tous les sommets sont de degré au plus 4
**Question** : Existe-t-il une coloration propre des sommets utilisant moins de 3 couleurs ?

*Indications : pour chaque sommet de $v$, associer un graphe $H_x$.*

**Problème** *k-COULEUR*
**Données** : Un graphe $G = (V, E)$, et un entier $k$.
**Question** : Décider s'il existe un coloriage du graphe utilisant exactement $k$ couleurs ?

**Question 14.9**. Prouver que le problème 3-*COULEUR* est $NP$-complet.

**Question 14.10**. Prouver que le problème *k-COULEUR* est $NP$-complet.

Nous supposerons que le problème Couverture de sommets est NP-complet
Couverture de sommets
**Données** : un graphe non-orienté $G$ et un entier $k$.
**Question** : $G$ contient-il une *couverture de sommets* de cardinalité au plus $k$ : c'est-à-dire un ensemble $\mathcal{S}$ de sommets tel que toutes les arêtes de $G$ sont incidentes à au moins un sommet de $\mathcal{S}$ ?

### b) Le problème de la clique maximum.

Considérons le problème de décision CLIQUE :
**Données** : un graphe non-orienté $G = (V, E)$ et un entier $k$.
**Question** : Existe-t-il une clique de taille $k$ (un sous graphe complet de $k$ sommets) ?

**Question 15.1**. Nous noterons par $G^c$ le complémentaire du graphe $G$.

Montrer que $G$ a une clique de taille $k$ si et seulement si $G^c$ a une couverture de sommets de taille $n - k$.

**Question 15.2**. Montrer que le problème CLIQUE est NP-complet.

Nous allons travailler sur une restriction du problème CLIQUE en considérant uniquemement les graphes dans lesquels tous leurs sommets sont de degré au plus 3. Nous le noterons 3-CLIQUE.

**Question 15.3**. Montrer que 3-CLIQUE est dans NP.

**Question 15.4**. Trouver l'erreur dans le raisonnement suivant : *Nous savons que le problème* CLIQUE *est NP-complet, il suffit donc de présenter une réduction de* 3-CLIQUE *à* CLIQUE. *Étant donné un graphe $G$ dont les sommets sont de degré inférieur à 3, et un entier $k$, la réduction laisse inchangée le graphe et le paramètre $k$ : clairement le résultat de la réduction est une entrée possible pour le problème CLIQUE. Par ailleurs, la réponse aux deux problèmes est identique. Cela prouve la justesse de la réduction et, par conséquent, la NP-complétude de la* 3-CLIQUE.

**Question 15.5**. Donner un algorithme polynomial en $O(|V|^4)$ pour le problème 3-CLIQUE.

### c) Problème du SET COVER

Rappellons la définition suivante : une *couverture de sommets* $S$ d'un graphe $G$ est un sous-ensemble de sommets tel que toutes les arêtes de $G$ sont incidentes à au moins un sommet de $S$.

Nous allons considérer le problème (voir figure 3.1 pour illustration) : SET COVER
**Données :** un ensemble $U$ de $u$ éléments, un ensemble $\mathcal{S}$ composé de sous-ensembles $S_1, \ldots, S_t$ de $U$, et un entier $k$.
**Question :** Existe-il au plus $k$ éléments de $\mathcal{S}$ tels que leur union est égale à $U$ ?

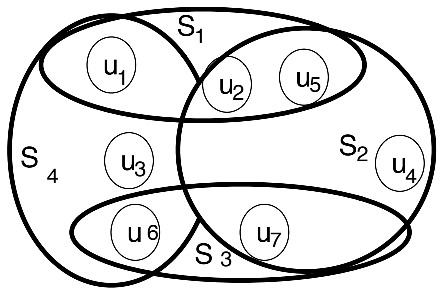



FIGURE 2.1 – $U = \{u_1, \ldots, u_7\}$, $S_1, S_2, S_4$ est un *set cover* de $U$ et de cardinalité 3.

**Question 16.1**. Montrer que ce problème est dans NP.

**Question 16.2**. Montrer que ce problème est dans NP-complet.

### d) Le problème *Stable* est NP-complet

Considérons le problème de décision $S$table
**Données** : un graphe non-orienté $G = (V, E)$ et un entier $k$.
**Question** : Existe-t-il un sous-ensemble $V' \subseteq V$, avec $|V'| = k$, tel que $u, v \in V' \Longrightarrow (u, v) \notin E$ ?

**Question 17.1**. Montrer que

1. si le graphe a une couverture de sommets de cardinalité $k$, alors il a un stable de cardinalité $n - k$ où $n$ est le nombre de sommets.
2. et réciproquement

**Question 17.2**. Montrer que le problème $S$table est NP-complet en supposant que le problème $C$ouverture de sommets (VC) est NP-complet.
**Données** : un graphe non-orienté $G = (V, E)$ et un entier $k$.
**Question** : $G$ contient-il une *couverture de sommets* $\mathcal{S}$ de cardinalité au plus $k$ ?

# VI) Le problème du $k$-centre

Dans cette partie, nous allons nous concentrer sur le problème du $k$-centre : étant donné un ensemble de villes dont les distances sont spécifiées, choisir $k$ villes afin d'installer des entrepôts de façon à minimiser la distance maximale d'une ville à l'entrepôt le plus proche. Un tel ensemble de $k$ villes est appelé $k$-centre.

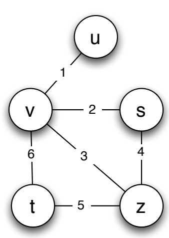
Le problème associé de décision est le suivant $k$-CENTRE
**Données :** un graphe complet $K = (V, E)$ muni d'une fonction de poids $w$ sur les arêtes, et des entiers strictement positifs $k$ et $b$.
**Question :** Existe-il un ensemble $S$ de sommets tel que $|S| = k$ et tel que tout sommet $v$ de $V$ satisfait la condition suivante

$$min\{w(v, u) : u \in S\} \leq b$$

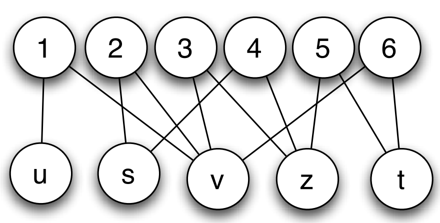
## a) Le problème Ensemble dominant est NP-complet

Rappellons les définitions suivantes :
- une couverture de sommets $S$ du graphe $G$ est un sous-ensemble de sommets tel que toutes les arêtes de $G$ sont incidentes à au moins un sommet de $S$.
- un ensemble dominant $C$ du graphe $G$ est un sous-ensemble de sommets tel que tout sommet est soit dans $C$ soit voisin d'un sommet de $C$.

Soit $G$ un graphe $G = (V, E)$. Nous allons construire un graphe $G' = (V', E')$ à partir de $G$ tel que
- $V' = V \cup E$ ;
- $E' = E \cup \{(v, a) | v \in V, a \in E, v$ est extrémité de l'arête a dans $G\}$




Graphe $G$




Graphe $G'$ (les arêtes dans $E$ ne sont pas dessinées)

**Question 18.1**. Montrer que si $S$ est une couverture de sommets du graphe $G$, alors $S$ est un ensemble dominant de $G'$.

**Question 18.2**. Montrer que si $S'$ est un ensemble dominant de $G'$, alors il existe un ensemble $S$ de même cardinalité de $S'$ qui est une couverture de graphe $G$,

**Question 18.3**. Exprimer le problème de minimisation de l'ensemble dominant sous forme de problème de décision.

**Question 18.4**. Montrer que ce problème est dans NP.

**Question 18.5**. Montrer que ce problème est dans NP-complet.

### b) Le problème $k$-CENTRE est NP-Complet

**Question 19.1**. Montrer que le problème $k$-CENTRE est dans NP.

**Question 19.2**. Montrer que $k$-CENTRE est NP-complet sachant que DOMINANT est NP-complet.

### c) Le problème $k$-CENTRE est in-approximable

**Question 20.1**. Montrer que si $P \neq NP$, quel que soit $\epsilon > 0$, il n'existe pas de $(2-\epsilon)$-approximation pour le problème du $k$-CENTRE MÉTRIQUE.

**Question 20.2**. Montrer que quel que soit $\alpha(n)$ calculable en temps polynomial, il n'existe pas de $\alpha(n)$-approximation pour $k$-CENTRE, à moins que $P = NP$.

# VII) Problèmes encodant les entiers

## a) Le problème *SOMME DE SOUS-ENSEMBLE* est NP-complet

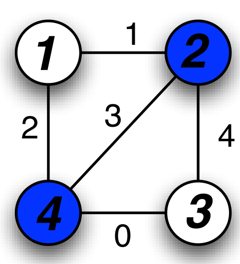
Nous considérons le problème *SOMME DE SOUS-ENSEMBLE*
**Données** : un ensemble fini d'entiers $A = \{a_1, a_2, \ldots, a_\ell\}$ et un entier $t \in \mathbb{N}$
**Question** : Existe-t-il un sous-ensemble $A' \subseteq A$ tel que $\sum_{a \in A'} a = t$ ?
Nous supposerons que le problème $C$ouverture de sommets (VC) est NP-complet.
**Données** : un graphe non-orienté $G = (V, E)$ et un entier $k$.
**Question** : $G$ contient-il une *couverture de sommets* $\mathcal{S}$ de cardinalité au plus $k$ ?

Considérons la transformation suivante
- On se donne un graphe $G = (V, E)$ et un entier $k$.
- Pour cela,
  - on numérote les sommets entre 0 et $n-1$ et des arêtes entre 0 et $m-1$.
  - on associe un entier $b_{ij}$ pour chaque couple (arête, sommet)

$$b_{ij} = \begin{cases} 1 & \text{si l'arête } i \text{ est incidente au sommet } j, \\ 0 & \text{sinon} \end{cases}$$

- On construit l'ensemble $A$ de la façon suivante : $(b = 4)$
  - Pour chaque sommet $j$ : $a_j = b^m + \sum_{i=0}^{m-1} b_{ij} b^i$
  - Pour chaque arête $j$ : est associé l'entier $b^j$
- On construit l'entier $t$ : $t = \underbrace{kb^m}_{cardinalite\ de\ la\ couverture} \sum_{i=0}^{m-1} \underbrace{2b^i}_{cout\ de\ l'\ arete\ i}$

**Question 21.1**. Calculer l'instance du problème *SOMME DE SOUS-ENSEMBLE* construite à partir de l'instance suivante de *VC* $\mathcal{I} = (G, k)$ :




**Question 21.2**. Montrer que si $G$ admet une couverture sommet $S$ de cardinalité $k$ alors il existe un sous-ensemble $A' \subseteq A$ tel que $\sum_{a \in A'} a = t$

**Question 21.3**. Montrer que s' il existe un sous-ensemble $A' \subseteq A$ tel que $\sum_{a \in A'} a = t$ alors $G$ admet une couverture sommet $S$ de cardinalité $k$

## b) Le problème *SOMME-DES-CARRES* est NP-complet

Nous considérons le problème *SOMME-DES-CARRES*
**Données** : Un ensemble fini d'entiers $A$, deux entiers positifs $k$ et $b$

**Question** : Décider si l'ensemble $A$ peut se partitionner en $k$ sous-ensembles disjoints $A_1, \dots, A_k$ tels que

$$\sum_{i=1}^{k} \left( \sum_{x \in A_i} x \right)^2 \leq b$$

Nous supposerons que le problème *PARTITION* est NP-complet. **Problème** *PARTITION*
**Données** : Un ensemble fini d'entiers $A$.
**Question** : Décider s'il existe un sous-ensemble $A' \subset A$ tel que $\sum_{x \in A'} x = \sum_{x \in A \setminus A'} x$ ?

**Question 22.1**. Soit $A$ un ensemble d'entiers et $A'$ un sous ensemble de $A$. Montrer que

$$(\sum_{x \in A'} x)^2 + (\sum_{x \in A \setminus A'} x)^2 \geq \frac{(\sum_{x \in A} x)^2}{2}$$

**Question 22.2**. Soit $A$ un ensemble d'entiers et $A'$ un sous ensemble de $A$. Montrer que si $(\sum_{x \in A'} x)^2 + (\sum_{x \in A \setminus A'} x)^2 = \frac{(\sum_{x \in A} x)^2}{2}$ alors $A'$ est tel que $\sum_{x \in A'} x = \sum_{x \in A \setminus A'} x$

**Question 22.3**. Donner la taille d'une instance du problème *SOMME-DES-CARRES* en supposant que les entiers sont codés en binaire.

**Question 22.4**. Montrer que le problème *SOMME-DES-CARRES* est dans NP.

**Question 22.5**. Montrer que le problème *SOMME-DES-CARRES* est NP-complet.

# Partie 3: Corrections des exercices

# I) Problème de décisions

## a) Graphe eulérien

Le graphe $G$ est *eulérien* si il existe un cycle en empruntant exactement une fois chaque arête du graphe $G$. On rappelle qu'un graphe connexe est eulérien si et seulement si chacun de ses sommets a un degré pair.

**Question 1.1**. Ecrire le problème de décision qui lui est associé et donner la taille de l'instance

Correction

**Données** : un graphe $G$

**Question** : Existe-t-il un cycle eulérien dans $G$ ?

La taille de l'instance est $O(|V|^2)$ car il faut coder le graphe $(O(|V|^2))$. En effet, si le graphe est représenté par la matrice d'adjacence, alors il nécessite $O(|V|^2)$ bits pour coder un graphe. Sinon, si le graphe est représenté par liste d'adjacence alors pour coder le graphe complet, il nécessite $O(|V|^2 \log |V|)$ bits pour le coder ce graphe (car le graphe complet possède $O(|V|^2)$ arêtes, et pour coder un nombre entier (étiquette du sommet) entre 0 et $n-1$, il faut $O(\log |V|)$ bits). □

**Question 1.2**. Trouver un algorithme polynomial qui détermine si le graphe est eulérien.

Correction

**Entrée** : un graphe $G$

**Sortie** : un booléen $b$

1. $b \leftarrow True$ ; *// O(1) opérations*
2. Pour tout sommet $v$ faire *// la boucle est exécutée $O(|V|)$ fois*
   (a) Calculer le degré de $v$ ; *// $O(|V|)$ opérations au pire*
   (b) Si le degré de $v$ est impair alors $b \leftarrow False$ ; *// O(1) opérations*
3. Retourner $b$ ; *// O(1) opérations*

Complexité : $O(|V|^2)$ opérations

□

## b) Formulation des problèmes de décisions

Mettre sous forme de problème de décision et évaluer la taille de leurs instances.

**Question 2.1**. Problème de savoir s'il existe un chemin entre deux sommets disjoints dans un graphe ;

Correction

Voici le problème sous forme de problème de décision :

**Données** : un graphe $G$, deux sommets distincts $u$ et $v$

**Question** : Existe-t-il un chemin entre $u$ et $v$ dans $G$ ?

La taille de l'instance est $O(|V|^2)$ car il faut coder le graphe (en utilisant $O(|V|^2)$ bits) et les étiquettes de deux sommets (en utilisant $O(log|V|)$ bits).

□

**Question 2.2**. Problème de connaitre la distance entre deux sommets disjoints dans un graphe ;

Correction

Voici le problème sous forme de problème de décision :
**Données** : un graphe $G$, deux sommets distincts $u$ et $v$, un entier $k$
**Question** : Existe-t-il un chemin entre $u$ et $v$ dans $G$ tel que sa longueur soit inférieure à $k$ ?

La taille de l'instance est $O(|V|^2)$ car il faut coder le graphe, les étiquettes de deux sommets, l'entier $k$ qui est $k \leq |V|$ (en utilisant $O(log|V|)$ bits). □

**Question 2.3**. Problème de connaitre la longueur de la chaîne maximum dans un graphe pondéré.

Correction

Voici le problème sous forme de problème de décision :
**Données** : un graphe $G = (V, E)$, une fonction $w : E \to \mathbb{N}$ deux sommets distincts $u$ et $v$, un entier $k$
**Question** : Existe-t-il un chemin entre $u$ et $v$ dans $G$ tel que sa longueur soit supérieur à $k$ ?

Nous allons calculer la taille de l'instance. Il faut coder le graphe, les étiquettes de deux sommets, l'entier $k$ qui est $k \leq |V|$. Cela nécessite $O(V^2)$ bits. Il faut coder en plus la fonction de poids.

Soit $w_{max} = max\{w(e) : e \in E\}$. Coder la fonction de poids nécessitent de coder $O(V^2)$ entiers qui sont inférieure à $w_{max}$

La taille de l'instance est : $\begin{array}{ll} O(V^2 \log w_{max}) & \text{si les entiers sont codés en binaire,} \\ O(V^2 * w_{max}) & \text{si les entiers sont codés en unaire.} \end{array}$

□

### c) Problèmes dans NP ou dans P

Les problèmes suivants sont-ils dans NP, dans P ? Justifier votre réponse.

**Problème** *P1*
**Données** : Un graphe $G = (V, E)$
**Question** : Existe-t-il cycle de longueur égale à $\lfloor \frac{|V|}{2} \rfloor$ ?

Correction

Le problème P1 est dans NP car étant donnée une suite de sommets $s$, on peut vérifier en temps polynomial si $s$ est un cycle de de longueur égale à $\lfloor \frac{|V|}{2} \rfloor$ dans $G$ (en $O(n^2)$ opérations).

□

**Problème** *P2*
**Données** : Un graphe $G = (V, E)$
**Question** : Existe-t-il cycle de longueur égale à 4 ?

Correction

Le problème P2 est dans NP car étant donnée une suite de sommets $s$, on peut vérifier en temps polynomial si $s$ est un cycle de de longueur égale à 4 dans $G$ (en $O(n)$ opérations).

Le problème P2 est dans P. Considérons l'algorithme suivant. Pour tout 4-uplets de sommets $(x, y, z, t)$ vérifier s'il est un cycle de de longueur égale à 4. Si c'est un cas, pour un seul $4-uplets$, alors répondre **oui** sinon répondre **non**. La complexité de cet algorithme est $O(n^5)$ (il y a $O(n^4)$ $4-uplets$ et tester s'il est un cycle de de longueur égale à 4 nécessite $O(n)$ opérations.

□

**Problème** *P3*
**Données** : Un graphe $G = (V, E)$, deux sommets $u$ et $v$ distincts de $G$ et un entier $k$.
**Question** : Existe-t-il un simple chemin entre $u$ et $v$ de longueur inférieure ou égale à $k$ ?

Correction

Le problème P2 est dans NP car étant donnée une suite de sommets $P$, on peut vérifier en temps polynomial si $P$ est un simple chemin entre $u$ et $v$ de longueur inférieure ou égale à $k$ (en $O(n)$ opérations).

Le problème P2 est dans P. Considérons l'algorithme suivant. Il suffit simplement de calculer le plus court chemin entre $u$ et $V$. Si la longueur du chemin est plus grande que $k$, ou si il n'existe pas de chemin entre $u$ et $v$ alors répondre **oui** sinon répondre **non**.

□

**Problème** *P4*
**Données** : Un graphe $G = (V, E)$, et un entier $k$.
**Question** : Existe-t-il un arbre couvrant tous les sommets de $G$ ayant moins de $k$ feuilles ?

Correction

NP-Complet (reduction avec chaine hamiltonien en prenant $k = 2$)

□

# II) Réduction polynomiale.

## a) Propriétés sur l'ordre $\leq$

Soient $A$ et $B$ deux problèmes de décision.

Une *réduction de $A$ vers $B$* est une fonction $f : I_A \to I_B$ calculable **en temps polynomial** telle que $w \in Oui(A)$ si et seulement si $f(w) \in Oui(B)$.

Nous notons $A \leq B$ lorsque $A$ se réduit à $B$.

**Question 4.1**. Montrer que $\leq$ est réflexive ($L \leq L$).

Correction

Intuitivement : un problème est aussi facile (et difficile) que lui-même. Il suffit de considérer la fonction identité pour $f$.

□

**Question 4.2**. Montrer que $\leq$ est transitive (i.e ; $L_1 \leq L_2$, $L_2 \leq L_3$ impliquent $L_1 \leq L_3$.)

Correction

Intuitivement : la relation "être plus facile que" est transitive.

$L_1 \leq L_2$ via la réduction $f$

$L_2 \leq L_3$ via la réduction $g$

$\Longrightarrow x \in OUI(I_{L_1})$ ssi $g(f(x)) \in OUI(I_{L_2})$. La composée de deux fonctions calculable **en temps polynomial** est calculable. □

**Question 4.3**. Montrer que $P = NP$ si et seulement si 3-SAT $\in P$.

Correction

Si $P = NP$, alors puisque $3 - SAT$ est dans $NP$, alors 3-SAT $\in P$. Réciproquement, Supposons qie $3 - SAT \in P$. Puisque $3 - SAT$ est complet, pour tout problème $B \in NP$, $B \leq 3 - SAT$. Donc $B \in P$.

□

## b) Réduction

Soient A, B et Q des problèmes de décision. Supposons que A est dans $P$, et que B est $NP$-dur. Dire si les affirmations suivantes sont vraies ou fausses :

1. Si A se réduit polynômialement à Q, alors $Q$ est dans $P$.

   Correction

   □

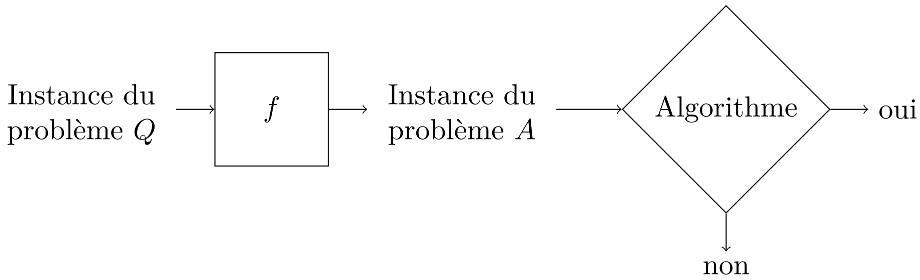
2. Si Q se réduit polynômialement à A, alors $Q$ est dans $P$.

   Correction

   Considérons une instance $I_Q$ du problème $Q$. Comme Q se réduit polynômialement à A, il existe une transformation polynomiale $f$ qui transforme une instance $I_Q$ de $Q$ en une instance $I_A$ de $A$ telle que $I_A \in OUI(A)$ si et seulement si $I_Q \in OUI(A)$.

   Comme $A$ est dans $P$, il existe un algorithme qui permet de vérifier $I_A \in OUI(A)$ en temps polynomial. Comme $I_A \in OUI(A)$ si et seulement si $I_Q \in OUI(A)$, il existe un algorithme qui permet de vérifier $I_Q \in OUI(Q)$ en temps polynomial.

Donc le problème $Q$ est dans $P$.




$\square$

3. Si Q se réduit polynômialement à B, alors $Q$ est $NP - dur$.

Correction

$\square$

4. Si B se réduit polynômialement à Q, alors $Q$ est $NP - dur$.

Correction

$\square$

# III) Problèmes de logique

## a) Le problème 2-SAT est dans $P$

Nous allons considérer de cet exercice des fonctions *booléennes*, c'est-à-dire de fonctions de $f$ de $\{1,0\}^n \to \{1,0\}$.

— Les variables ne peuvent prendre que deux valeurs, vrai (codé par 1) ou faux (codé par 0).

— La fonction booléenne $f$ est composée de variables et d'opérateurs comme négation $(\neg)$ la conjonction $(\wedge)$ la disjonction $(\vee)$, l'implication $\to$.

Voici la table de vérité pour les opérateurs booléens cités ci-dessus :

| $u$ | $v$ | $\neg u$ | $(u \wedge v)$ | $u \vee v$ | $u \to v$ |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |

Considérons une fonction $t : U \to \{0,1\}$. On dira que la fonction $t$ satisfait la fonction $f$ si la fonction $f$ retourne 1 avec les valeurs de $t$ en entrée.

**Question 6.1**. Nous allons considérer la fonction $\phi_1$ : telle que $\{1,0\}^3 \to \{1,0\}$.

$$\phi_1(x,y,z) = (y \vee \neg z) \wedge (\neg y \vee z) \wedge (y \vee \neg x).$$

Donner une fonction $t : U \to \{0,1\}$ telle que $t$ satisfait $\phi_1$

Correction

Il existe une affectation $t$ qui satisfait $t(x) = 0$, $t(y) = 1$ et $t(z) = 1$.

$$t(\phi) = \phi_1(0,1,1) = 1 \wedge 1 \wedge 1 = 1$$

$\square$

**Question 6.2**. Que se passe-t-il pour $\phi_2$

$$\phi_2(x,y,z) = (y \vee z) \wedge (\neg y \vee z) \wedge (\neg z \vee x) \wedge (\neg z \vee \neg x)?$$

Correction

Il n'existe pas d'affectation qui satisfait $\phi_2$.

| $x$ | $y$ | $z$ | $(y \vee z)$ | $(\neg y \vee z)$ | $(\neg z \vee x)$ | $(\neg z \vee \neg x)$ | $\phi_2(x,y,z)$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

$\square$

Une clause est une fonction de $f$ de $\{1,0\}^n \to \{1,0\}$ composée de variables et d'opérateurs comme négation ($\neg$) et la disjonction ($\vee$). Par exemple $C(u_2,u_3) = (u_2 \vee \neg u_3)$ est une clause.

Donnons la définition du problème 2-SAT

*Données* : un ensemble $U$ de variables $\{u_1, u_2, \ldots, u_n\}$ et une formule logique $\phi = C_1 \wedge \cdots \wedge C_\ell$ des clauses de 2 littéraux

*Question* : Existe-t-il une fonction $t : U \to \{0,1\}$ telle que $t$ satisfait $\phi$ ?

**Question 6.3**. Montrer que $u \vee v = (\neg u \Rightarrow v) \wedge (\neg v \Rightarrow u)$.

Correction

| $u$ | $v$ | $(u \Rightarrow v)$ | $(\neg u \Rightarrow v)$ | $(\neg v \Rightarrow u)$ | $(\neg u \Rightarrow v) \wedge (\neg v \Rightarrow u)$ | $u \vee v$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |

□

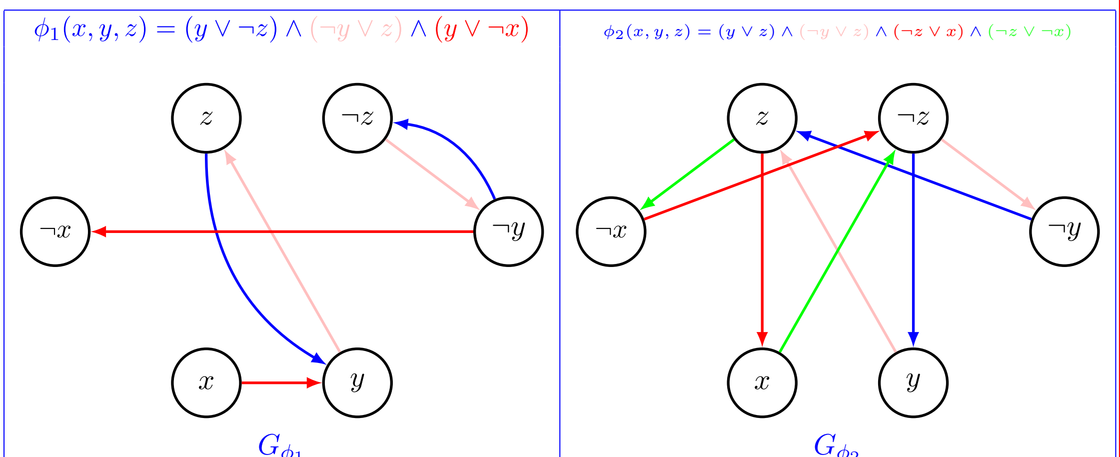
A partir d'une instance $(U,\phi)$ de 2-SAT, nous construisons un graphe orienté $G_\phi = (V,E)$ tel que

— un sommet par un littéral de $U$

— un arc par implication (en transformant chaque clause par deux implications)

**Question 6.4**. Dessiner les graphes $G_{\phi_1}$ et $G_{\phi_2}$

Correction




□

**Question 6.5**. Montrer que si $G_\phi$ possède un chemin entre $u$ et $v$, alors il possède aussi un chemin entre $\neg v$ et $\neg u$

Correction

Supposons qu'il existe un chemin $\mathcal{A}$ dans $G_\phi$ entre $u$ et $v$. Notons $\mathcal{A} = \{u_1 = u, u_2, \ldots, u_t = v\}$.

Remarquons que pour tout $1 \leq i \leq t$, $G_\phi$ a arc $(u_i \Rightarrow u_{i+1})$ et $G_\phi$ possède aussi l'arc $(\neg u_{i+1} \Rightarrow \neg u_i)$ ( Rappel $u \vee v = (\neg u \Rightarrow v) \wedge (\neg v \Rightarrow u)$.) Notons $\mathcal{A}^c = \{\neg u_t = \neg v, \neg u_{t-1}, \ldots \neg u_1 = \neg u\}$ existe dans $G_\phi$

Donc il possède aussi un chemin entre $\neg v$ et $\neg u$ .

□

**Question 6.6**. Montrer qu'il existe une variable $u$ de $U$ tel que $G_\phi$ contient un cycle entre $u$ vers $\neg u$ si et seulement si $\phi$ n'est pas satisfiable.

Correction

Supposons qu'il existe un cycle $\mathcal{A}$ dans $G_\phi$ tel qu'il contient un cycle entre $u$ vers $\neg u$. Notons $\mathcal{A} = \{u_0 = u, u_1, \ldots, u_k = \neg u, \ldots u_t = u\}$

Montrons par contradiction. Supposons qu'il existe une affectation $t$ telle qu'elle satisfasse $\phi$. Sans perte de généralité, supposons que $t(u) = 0$. Nous considérons le chemin $u_0 = u, \ldots u_k = \neg u$. Comme il y a un arc entre $u_0$ et $u_1$, cela implique qu'il existe la clause $(\neg u_0 \vee u_1)$ dans $\phi$. Comme $t(\phi) = 1$, la clause $(\neg u_0 \vee u_1)$ est satisfaite par $t$ et ainsi $t(u_1) = 1$. Donc, on peut étendre ce raisonnement et prouver que pour $0 \leq \ell \leq k$ on a $t(u_\ell) = 1$. Cela signifie que $t(u_k) = 1$, et que$t(\neg u) = 1$. Ceci contredit le fait que $t(u) = 0$.

Nous pouvons appliquer le même raisonnement si $t(u) = 1$ en considérant le chemin $\{u_k = \neg u, u_{k+1} \ldots u_t = u\}$.

Donc s'il existe une variable $u$ de $U$ tel que $G_\phi$ contient un cycle entre $u$ vers $\neg u$, alors $\phi$ n'est pas satisfiable.

□

**Question 6.7**. Montrer que un algorithme en temps polynomial peut résoudre le problème 2-SAT.

Correction

**Algorithme 2 :** Décider s'il existe une affectation $t : U \rightarrow \{0, 1\}$ qui satisfait la fonction $\phi$

**Entrées :** Un ensemble $U$ de variables, une formule logique $\phi$ de clauses de 2 littéraux

**Output :** un booléen $b$

```
début
    b ← True ;
    Construire le graphe G_φ ;
    pour chaque variable u dans U faire
        text loop
    fin
    read current section;
    si il existe un chemin de u vers ¬u alors
        si il existe un chemin de ¬u vers u alors b ← False ;
        ;
    fin
    Retourner b;
fin
```

Pour calculer un chemin de $s$ vers $d$, il suffit de réaliser soit un parcours en profondeur ou en largeur.

La complexité de cet algorithme est $O(|U|^2)$

□

Rappellons la définition du problème : 3-SAT

*Données* : Un ensemble $U$ de variables, et une formule $F$ sous la forme d'une conjonction de disjonction de 3 littéraux (variables ou leur négation). Formellement : une formule

$$F = C_1 \wedge C_2 \cdots \wedge C_\ell$$

avec

$$C_i = y_{i,1} \vee y_{i,2} \vee y_{i,3},$$

où pour tout $i$, $j$, $y_{i,j}$ est soit $x_k$, soit $\neg x_k$ pour l'un des $x_k$, pour des variables $\{x_1, \cdots, x_n\}$ *Question* :Décider si $F$ est satisfiable : c'est-à-dire décider s'il existe $x_1, \cdots, x_n \in \{0,1\}$ tel que $F$ s'évalue à vrai pour cette valeur de ses variables $x_1, \cdots, x_n$.

### b) Variante du problème 3-SAT :

**Donnée**: Une formule $F'$ sous la forme d'une conjonction de clauses même si chaque variable apparaisse **au plus soit trois fois** dans $F$ sous sa forme négative ou positive.

**Réponse**: Décider si $F'$ est satisfiable.

**Question 7.1**. Soit $x_1$, $x_2 \ldots x_k$, Considérons une formule $\phi$ par l'ensemble de $k$ clauses $(x_i \vee \neg x_{i+1})$ pour $i = 1, \ldots k-1$, et de la clause $(x_k \vee \neg x_1)$. Montrer que une assignation $\phi$ est satisfaite si et seulement si pour $i = 1, \ldots k-1$ $x_k = x_1$

Correction

Supposons que la clause $(x_i \vee \neg x_{i+1})$ est satisfait et que $x_i$ est faux. Ceci implique que $x_{i+1}$ est égale à faux. La structure cyclique de la formule implique que si $\phi$ est satisfiable si tous les $x_i$ ont les mêmes valeurs.

□

**Question 7.2**. Prouver que ce problème est dans NP.

Correction

Le problème est dans NP. A ECRIRE

□

**Question 7.3**. Prouver que ce problème est NP-complet à partir d'une réduction du problème $3-SAT$.

Correction

Nous allons construire une instance du problème $SC$ à partir d'une instance du problème $3-SAT$ de la façon suivante. Pour chaque variable $x$ qui apparaît dans plus de trois clauses dans $F$, en supposant que $x$ apparaît dans $k$ clauses, nous allons considérer la procédure suivante :

1. Créer $k$ nouvelles variables $x_1, \ldots, x_k$
2. Remplacer l'occurrence $i$ème de $x$ par $x_i$
3. Ajouter les $k$ clauses $(x_i \vee \neg x_{i+1})$ pour $i = 1, \ldots k-1$, et de la clause $(x_k \vee \neg x_1)$.

Nous obtenons ainsi une nouvelle formule $F'$.

Supposons que $F$ est satisfiable. Soit $t$ une assignation telle que son affectation rend $F$ satisfiable. Nous allons construire une assignation $t'$ à partir de $t$ telle que

1. $t'(x) = t(x)$ pour toute variable $x$ apparaissant moins de 3 fois.
2. $t'(x_i) = t(x)$ pour $i = 1, \ldots k$,pour toute variable $x$ apparaissant plus de $k > 3$ fois.

Il est facile de remarquer que l'assignation $t'$ satisfait $F'$.

Maintenant, supposons que $F'$ est satisfiable. Soit $x$ une variable apparaissant $k$ fois dans $F$. D'après la question précédente, pour que l'ensemble de $k$ clauses $(x_i \vee \neg x_{i+1})$ pour $i = 1, \dots k-1$, et de la clause $(x_k \vee \neg x_1)$ soient satisfaite, il faut que pour $i = 1, \dots k-1$ $x_k = x_1$

Soit $t'$ une assignation telle que son affectation rend $F'$ satisfiable. Nous allons construire une assignation $t$ à partir de $t'$ telle que

1. $t(x) = t'(x)$ pour toute variable $x$ apparaissant moins de 3 fois.
2. $t(x) = t(x_1)$ pour toute variable $x$ apparaissant plus de $k > 3$ fois.

Il est facile de remarquer que l'assignation $t$ satisfait $F'$.

$F$ est satisfiable si et seulement si $F'$ est satisfiable.

□

**Question 7.4**. Prouver $3 - SAT$ est NP-complet même si chaque variable apparaisse **exactement trois fois** dans $F$ sous sa forme négative ou positive.

Correction

A ECRIRE

□

**Question 7.5**. Prouver $3 - SAT$ est NP-complet même si chaque variable apparaisse **deux fois** sous sa forme positive et **une fois** sous sa forme négative dans $F$.

Correction

A ECRIRE

□

### c) Le problème MAX2SAT est NP-complet.

L'objectif de cet exercice est de prouver que le problème MAX2SAT est NP-complet.

**Donnée**: Une formule $F'$ sous la forme d'une conjonction de clauses d'au plus 2 littéraux, un entier $K$

**Réponse**: Décider s'il existe une assignation $x_1, \cdots, x_n \in \{0, 1\}$ telle que au moins $K$ clauses de $F'$ sont satisfaites. s'évalue à vrai pour cette valeur de ses variables $x_1, \cdots, x_n$.

**Question 8.1**. Soit une clause $C = x \vee y \vee z$. Nous construisons une formule $\phi_C$ sous la forme d'une conjonction de 10 clauses en introduisant un nouveau littéral $p_C$ :

$$\phi_C = \begin{array}{l} (x) \wedge (y) \wedge (z) \wedge (p_C) \\ (\neg x \vee \neg y) \wedge (\neg y \vee \neg z) \wedge (\neg z \vee \neg x) \\ (x \vee \neg p_C) \wedge (y \vee \neg p_C) \wedge (z \vee \neg p_C) \end{array}$$

Déterminer le nombre maximun de clauses satisfaites dans le cas où

1. la clause $C$ est satisfaite
2. la clause $C$ n'est pas satisfaite

Correction

1. Considérons le cas où la clause $C$ est satisfaite. Nous allons énumérer les différents cas.

(a) Supposons que tous les littéraux de $C$ sont égaux à vrai. Les 3 premières clauses et les 3 dernières de $\phi_C$ sont satisfaites. De plus, les cinquième, sixième, septième clauses de $\phi_C$ ne sont pas satisfaites. Ensuite, la clause $(p_C)$ est satisfaite si $p_C = 1$. Le nombre maximum de clauses satisfaites est égale à 7.

(b) Supposons qu'un seul littéral de $C$ est égal à faux. Sans perte de généralité supposons que cela soit $z = 0$. Alors les clauses $(x)$, $(y)$, $(\neg y \vee \neg z)$, $(\neg z \vee \neg x)$, $(x \vee \neg p_C)$, $(y \vee \neg p_C)$ sont satisfaites. Quelque soit la valeur de $p_C$, seule une de ces deux clauses $(p_C)$ et $(z \vee \neg p_C)$ peuvent être satisfaite. Le nombre maximum de clauses satisfaites est égale à 7.

(c) Supposons que deux littéral de $C$ sont égaux à faux. Sans perte de généralité supposons que cela soit $z = 0$ et $y = 0$. Alors les clauses $(x)$, $(\neg x \vee \neg y)$, $(\neg y \vee \neg z)$, $(\neg z \vee \neg x)$, $(x \vee \neg p_C)$ sont satisfaites. En prenant $p_C = 0$, le nombre maximun de clauses satisfaites parmi les 3 clauses $(p_C)$ et $(y \vee \neg p_C)$ et $(z \vee \neg p_C)$ vaut 2. Le nombre maximum de clauses satisfaites est égale à 7.

Donc si la clause $C$ est satisfaite, alors le nombre maximum de clauses satisfaites de $\phi_C$ est égale à 7.

2. Considérons le cas où la clause $C$ n'est pas satisfaite. Les trois littéraux sont à faux. Cela signifie que les 3 premières clauses de $\phi_C$ ne sont pas satisfaites. Les cinquième, sixième, septième clauses de $\phi_C$ sont satisfaites. Ensuite, la valeur des autres clauses dépendent du littéral $p_C$. Afin de maximiser le nombre de clauses, il faut que $p_C = 1$.

Donc si la clause $C$ n'est pas satisfaite, alors le nombre maximum de clauses satisfaites de $\phi_C$ est égale à 6.

□

**Question 8.2**. Prouver que le problème MAX2SAT est dans NP.

Correction

Le problème MAX2SAT est dans NP car étant donnée une assignation $t$, on peut vérifier en temps polynomial si $t(F') = 1$ (il suffit de vérifier que pour chaque clause $c$ de $F'$, $t(c) = 1$. Cela s'effectue en $O(m + n)$ opérations avec $m$ correspondant au nombre de clauses et $n$ au nombre de variables.

□

**Question 8.3**. Montrer que le problème Max2SAT est NP-complet.

Correction

D'après la question précédente, le problème Max2SAT est dans NP.

Nous allons transformer une instance de Max2SAT à partir d'une instance de 3SAT de la façon suivante. Etant donnée une instance $I$ de 3-SAT, on construit une instance $I'$ de Max2SAT de la façon suivante : pour chaque clause $C_i = (x_1 \vee x_2 \vee x_3)$ de $F$, on rajoute les 10 clauses précédentes à $F'$ où $x_1$, $x_2$ et $x_3$ remplacent $x$, $y$ et $z$. $p_C$ est remplacée par une variable supplémentaire $p_{C_i}$ associée à la clause $C_i$. Ainsi, l'instance $I'$ a $10 \cdot m$ clauses si $F$ en possède m et $K = 7m$. La réduction proposée est bien polynomiale.

D'après la question précédente, on peut prouver que au moins $K$ clauses de $F'$ est satisfiable ssi $F$ est satisfiable. □

### d) Le problème $k$-SAT NAE est NP-complet

Considérez le problème suivant :

$k$-SAT NAE
*Données* : un ensemble $U'$ de variables $\{u_1, u_2, \ldots, u_n\}$ et une formule logique $L = C_1 \wedge \cdots \wedge C_\ell$ avec $C_i = (y_{i,1} \vee y_{i,2} \vee \cdots \vee y_{i,k})$ où $y_{i,j}$ est égal soit à l'un des $u_k$ ou soit à l'un des $\neg u_k$
*Question* : Existe-t-il une fonction $t : U' \to \{0, 1\}$ telle que $t$ satisfait $L$ et telle que les littéraux de chaque clause ne sont pas toutes de la même valeur ?

**Question 9.1**. Montrer que 4-SAT NAE est NP-complet sachant que 3-SAT est NP-complet. *Indication : Introduire une nouvelle variable $z$ et l'insérer dans toutes les clauses*

Correction

Le problème est dans NP car étant donné une affectation des variables, on peut vérifier en temps polynomial que cette affectation satisfait $L$. On va réduire 3-SAT à 4-SAT NAE. Soit $F$ une formule de 3-SAT sur les variables $U$. On ajoute une unique variable distincte $z$ et on forme les clauses pour 4-SAT NAE en remplaçant chaque clause $C_i = y_{i,1} \vee y_{i,2} \vee y_{i,3}$ de l'instance de 3-SAT par $C'_i = y_{i,1} \vee y_{i,2} \vee y_{i,3} \vee z$ .

Cette transformation se fait bien en temps polynomial en la taille de l'instance 3-SAT.

Si l'instance donnée de 3-SAT est satisfiable, la même affectation des variables tout en fixant pour $z$ la valeur 0 fournit une affectation valide pour 4-SAT NAE.

Réciproquement, supposons que l'instance construite de 4-SAT NAE soit satisfiable.

Si la valeur de vérité de $z$ dans l'affectation correspondante est 0, alors les valeurs des variables $u_i$ dans l'affectation donnent une affectation valide pour la formule $\phi$ pour l'instance de 3-SAT . Si au contraire $z$ vaut 1, on change toutes les valeurs de toutes les variables dans l'affectation. L'affectation reste valide pour 4-SAT NAE car au moins un littéral par clause dans l'affectation initiale valait 0, et vaut donc maintenant 1, tandis que z vaut 0. Et on se retrouve dans le cas précédent.

On a donc bien prouvé que $3 - SAT \leq 4 - SATNAE$.

□

**Question 9.2**. Montrer que 3-SAT NAE est NP-complet sachant que 4-SAT NAE est NP-complet. 3-SAT est NP-complet. *Indication : Utiliser la technique pour la transformation de SAT en 3-SAT.*

Correction

A ECRIRE

□

# IV) Différentes Variantes du problème CYCLE HAMILTONIEN

Nous allons supposer que le problème CYCLE HAMILTONIEN est NP-Complet.
CYCLE HAMILTONIEN
*Données* : un graphe non-orienté $G$.
*Question* : $G$ contient-il un cycle hamiltonien ?

## a) Le problème CHAINE HAMILTONIEN est NP-complet

Considérons le problème CHAINE HAMILTONIEN suivant :
CHAINE HAMILTONIENNE
*Données* : un graphe non-orienté $G$, deux sommets $u$ et $v$ distincts de $G$.
*Question* : $G$ contient-il une chaine hamiltonienne entre $u$ et $v$ ?

**Question 10.1**. Montrer que le problème CHAINE HAMILTONIEN est dans NP.

Correction

Le problème CHAINE HAMILTONIENNE est dans NP car étant donnée une chaîne, on peut vérifier en temps polynomial si elle passe une fois et une seule par chaque sommet du graphe et qu'elle a $u$ et $v$ comme extrémité.

□

**Question 10.2**. Montrer que le problème CHAINE HAMILTONIEN est NP-complet.

Pour cela, nous allons faire la réduction à partir du problème CYCLE HAMILTONIEN. Soit $\mathcal{I} =< G = (V, E) >$ une instance du problème CYCLE HAMILTONIEN. Maintenant, nous allons transformer cette instance en une instance $\mathcal{I}'$ du problème CHAINE HAMILTONIENNE de la façon suivante : Nous allons construire un graphe $G' = (V', E')$ tel que

— Soit $u$ un sommet arbitraire de $V$
— $V' := V \cup \{v\}$ tel que $v$ est un sommet n'appartenant pas dans $V$
— $E' := E \cup \{(v, \ell) : \ell$ est un voisin de $u$ dans $G\}$

Continuez la preuve.

Correction

Cette transformation peut se faire en temps polynomial (nous avons juste copier le graphe $G$ en rajoutant un sommet et des arêtes).

Il est facile de prouver que :

— Si il existe un cycle hamiltonien dans $G$, alors il existe une chaine hamiltonnienne dans $G'$.
Soit $C = (u, \ell_1, \ldots, \ell_{n-1}, u)$ un cycle hamiltonien dans $G$. Nous construisons la chaîne $\mathcal{P} = (u, \ell_1, \ldots, \ell_{n-1}, v)$ dans le graphe $G'$. Cette chaîne est hamiltonienne : elle passe une fois et une seule par $v$ et par chaque sommet de $G$ puisque $C$ est un cycle hamiltonien.
— Si il existe une chaine hamiltonnienne dans $G'$, alors il existe un cycle hamiltonien dans $G$.
Soit $\mathcal{P} = (u, \ell_1, \ldots, \ell_{n-1}, v)$ une chaîne dans un graphe $G'$. Le cycle $C = (u, \ell_1, \ldots, \ell_{n-1}, u)$ est hamiltonnien pour le graphe $G$.

Donc CYCLE HAMILTONIEN $\leq$ CHAINE HAMILTONIENNE

Donc le problème CHAINE HAMILTONIENNE est NP-complet.

□

## b) Le problème Chaine est NP-complet

Le problème Chaine est le problème de décision suivant
Chaine
*Données* : un graphe non-orienté $G$ de $n$ sommets, deux sommets $u$ et $v$ distincts de $G$.
*Question* : $G$ contient-il une chaine de longueur $n/2$ entre $u$ et $v$ ?

**Question 11.1**. Montrer que le problème Chaine est dans NP.

Correction

Le problème Chaine est dans NP car étant donnée une chaîne, on peut vérifier en temps polynomial si sa taille est de longueur $n/2$ et qu'elle a $u$ et $v$ comme extrémité.

□

**Question 11.2**. Montrer que le problème Chaine est NP-complet.

Correction

Nous allons faire la réduction à partir du problème Chaine Hamiltonienne. Soit $\mathcal{I} =< G = (V, E), u, v >$ une instance du problème Chaine Hamiltonienne.

Nous transformons cette instance en une instance $\mathcal{I}'$ du problème Chaine de la façon suivante. Nous construisons un graphe $G' = (V', E')$ tel que

> $G'$ est une copie du graphe $G$ plus une chaîne de $|V|$ sommets dont un seul sommet de cette chaîne est voisin de $u$.

Cette transformation se fait en temps polynomial (nous avons juste copier le graphe $G$ en rajoutant un sommet et des arêtes).

Il est facile de prouver que

> il existe une chaine hamiltonnienne dans $G$ si et seulement si il existe une chaine hamiltonnienne dans $G'$ de longueur $\frac{|V'|}{2}$.

Donc le problème Chaine Hamiltonienne se réduit au problème chaine ; et ce dernier est donc NP-complet.

□

## c) Chevaliers de la table ronde

Etant donnés $n$ chevaliers, et connaissant toutes les paires de féroces ennemis parmi eux, est-il possible de les placer autour d'une table circulaire de telle sorte qu'aucune paire de féroces ennemis ne soit côte à côte ?

Correction

— Le problème est dans NP car étant donné un plan de table, on peut vérifier en temps polynomial si pour chaque chevalier, il n'est pas à coté d'un ennemi.
— Nous allons faire la réduction à partir du problème du cycle hamiltonien
Soit $\mathcal{I} =< G = (V, E) >$ une instance du problème du cycle hamiltonien.
Maintenant, nous allons transformer cette instance en une instance $\mathcal{I}_2$ du problème des Chevaliers de la table ronde de la façon suivante :
— chaque sommet du graphe est un chevalier.

— Deux chevaliers sont des ennemis si et seulement si il n'existe pas une arête dans $G$ impliquant les deux sommets représentés par ces deux chevaliers

Cette transformation peut se faire en temps polynomial (nous avons construit le complémentaire du graphe $G$).

Il est facile de prouver que :

— Si il existe un cycle hamiltonien dans $G$, alors il existe un plan de table. Il suffit de voir qu'une arête dans $G$ correspond au fait que les deux chevaliers ne sont pas ennemis. Donc le plan de table correspond au cycle hamiltonien.

— Si il existe un plan de table alors il existe un cycle hamiltonien dans $G$.

Donc le problème des CHEVALIERS DE LA TABLE RONDE est plus difficile que le problème du cycle hamiltonien.

Donc le problème des CHEVALIERS DE LA TABLE RONDE est NP-complet.

□

## d) Le problème VOYAGEUR DE COMMERCE est NP-complet

Considérez le problème VOYAGEUR DE COMMERCE :

*Données* : Un graphe complet $G$ muni d'une fonction $d$ de coût positif sur les arêtes, et un entier $k$.

*Question* : Existe-t-il un circuit passant au moins une fois par chaque ville et dont la somme des coûts des arêtes est au plus $k$ ?

**Question 13.1**. Montrer que VOYAGEUR DE COMMERCE est NP-complet même si la fonction poids respecte l'inégalité triangulaire. (C'est-à-dire que si les arêtes vérifient l'inégalité triangulaire, c'est-à-dire si pour tout triplet de sommets $(u, v, w)$, on a $d(u, v) \leq d(u, w) + d(w, v)$.)

Correction

Rapellons le problème suivant :

**Données :** un graphe complet $K = (V, V \times V)$ muni d'une fonction de coût positive $C : V \times V \to \mathbb{N}$ sur les arêtes, respectant l'inégalité triangulaire, un entier $k$.

**Question :** Existe-il un cycle passant par chaque sommet exactement une fois et ayant un coût inférieur à $k$ ?

Le problème *voyageur de commerce* ayant une métrique respectant l'inégalité triangulaire est dans NP car étant donné un cycle, on peut vérifier en temps polynomial si

— le cycle passe par chaque sommet exactement une fois

— son coût est inférieur à $k$.

Nous allons faire la réduction à partir du problème CYCLE HAMILTONIEN. Soit $\mathcal{I} =< G = (V, E), >$ une instance du problème CYCLE HAMILTONIEN.

Nous transformons cette instance en une instance $\mathcal{I}'$ du problème *voyageur de commerce* de la façon suivante. Nous affectons $|V|$ à $k$ et nous construisons un graphe $K = (V, V \times V)$ muni de la fonction de coût sur les arêtes suivantes :

$$C(u, v) = \begin{cases} 1 & si \quad (u, v) \in E \\ 2 & si \quad (u, v) \notin E \end{cases}$$

La fonction de coût respecte bien l'inégalité triangulaire. Cette transformation se fait en temps polynomial car nous avons

— copié le graphe $G$ et rajouté des arêtes en au pire $\mathcal{O}(|V|^2)$ opérations ;
— construit la fonction de coût $C$ en $\mathcal{O}(|V|^2)$ opérations.

Il est facile de prouver l'assertion suivante :

il existe un cycle hamiltonnienne dans $G$ si et seulement si il existe un cycle du voyage de commerce de coût $n$.

□

**Question 13.2**. Montrer que le problème du voyageur de commerce ayant une métrique respectant l'inégalité triangulaire est dans NP-complet.

Correction

Rapellons le problème suivant :

**Données :** un graphe complet $K = (V, V \times V)$ muni d'une fonction de coût positive $C : V \times V \to \mathbb{N}$ sur les arêtes, respectant l'inégalité triangulaire, un entier $k$.

**Question :** Existe-il un cycle passant par chaque sommet exactement une fois et ayant un coût inférieur à $k$ ?

Le problème *voyageur de commerce* ayant une métrique respectant l'inégalité triangulaire est dans NP car étant donné un cycle, on peut vérifier en temps polynomial si

— le cycle passe par chaque sommet exactement une fois
— son coût est inférieur à $k$.

Nous allons faire la réduction à partir du problème Cycle Hamiltonien. Soit $\mathcal{I} =< G = (V, E), >$ une instance du problème Cycle Hamiltonien.

Nous transformons cette instance en une instance $\mathcal{I}'$ du problème *voyageur de commerce* de la façon suivante. Nous affectons $|V|$ à $k$ et nous construisons un graphe $K = (V, V \times V)$ muni de la fonction de coût sur les arêtes suivantes :

$$C(u, v) = \begin{cases} 1 & si \quad (u, v) \in E \\ 2 & si \quad (u, v) \notin E \end{cases}$$

La fonction de coût respecte bien l'inégalité triangulaire. Cette transformation se fait en temps polynomial car nous avons

— copié le graphe $G$ et rajouté des arêtes en au pire $\mathcal{O}(|V|^2)$ opérations ;
— construit la fonction de coût $C$ en $\mathcal{O}(|V|^2)$ opérations.

Il est facile de prouver l'assertion suivante :

il existe un cycle hamiltonnienne dans $G$ si et seulement si il existe un cycle du voyage de commerce de coût $n$.

□

# V) Problèmes de graphe

## a) Les problèmes COLORATION DE GRAPHE

Une *coloration* d'un graphe $G$ est une fonction associant à tout sommet $V$ du graphe $G$ un entier représentant une couleur dans $[1, \ldots, |V|]$, telle que deux sommets adjacents n'aient pas la même couleur. Le problème d'optimisation classique est de trouver une coloration avec un nombre de couleurs minimum.

**Question 14.1**. Donner un algorithme polynomial construisant une coloration d'un arbre $T = (V_T, E_T)$ qui utilise seulement 2 couleurs.

Correction

A ECRIRE

□

**Question 14.2**. Le problème *k-Arbre-coloration* est-il NP-complet ou dans P ?

**Problème** *k-Arbre-coloration*

**Données** : Un arbre $T = (V, E)$ et un entier $k$.

**Question** : Existe-t'il une façon de colorier les sommets de $T$ avec au plus $k$ couleurs de telle sorte qu'aucune arête n'ait les extrémités de la même couleur ?

Correction

A ECRIRE

□

Définissons le problème *k-coloration*.

**Problème** *k-coloration*

**Entrée** : Un graphe $G = (V, E)$, un entier $k$

**Question** : Existe-t-il une coloration propre des sommets utilisant moins de $k$ couleurs ?

**Question 14.3**. Le problème 2-coloration est-il NP-complet ou dans P ? Justifier votre réponse.

Correction

A ECRIRE

□

A partir de maintenant, nous supposerons que le problème *3-coloration* est NP-complet.

**Question 14.4**. Montrer que le problème *4-coloration* est NP-complet.

Correction

Il est clair que le problème *4-coloration* est dans NP (car c'est un sous-problème du problème 3-coloration). Nous construisons l'instance du problème *4-coloration* à partir d'une instance du problème *3-coloration* composé du graphe $G = (V, E)$ . Le graphe $G' = (V', E')$ est définie de la façon suivante :

1. Créer un sommet $u$ qui n'est pas dans $V$
2. $V' \to V \cup \{u\}$
3. $E' = E \cup \{(u, v) : v \in V\}$

Il est clair le graphe $G'$ peut se construire en temps polynomial.

Supposons qu'il existe une fonction $c : V \to \{1, 2, 3\}$ qui est une propre coloration de trois couleurs dans $G$. Nous allons contruire une coloration $c'$ qui est une propre coloration de quatre couleurs dans $G'$ de la façon suivante :

1. pour chaque sommet $v$ de $G$
   (a) $c'(v) = \begin{cases} c(v) & si \quad v \in V \\ 4 & si \quad v = u \end{cases}$

La coloration $c'$ ainsi construire est une propre coloration utilisant 4 couleurs dans $G'$ car

1. Pour toute arête $e \in E$, ses extrémités sont de couleurs différentes dans $c'$ (puisque c'est le cas dans $c$).
2. Pour toute arête $e \in E' \backslash E$, l'arête $e$ a $u$ comme extrémité et donc ses extrémités sont de couleurs différentes dans $c'$ (puisque $u$ est le seul sommet de couleur 4 )

Supposons qu'il existe une fonction $c' : V' \to \{1, 2, 3, 4\}$ qui est une propre coloration de trois couleurs dans $G'$. Sans perte de généralité, $c'(u) = 4$ nous supposerons Nous allons contruire une coloration $c$ qui est une propre coloration de trois couleurs dans $G$ tel que pour chaque sommet $v$ de $G$, on a $c(v) = c'(v)$.

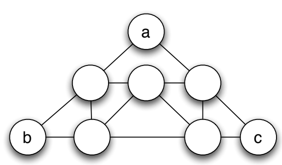
Comme $u$ est voisin de tous les autres sommets, les autres sommets ne sont pas de couleur 4. Donc la coloration $c$ ainsi construire est une propre coloration de trois couleurs dans $G$.

Ainsi $G'$ possède une 4-coloration propre si et seulement si $G$ possède une 3-coloration propre.

Donc le problème 4-COLORATION est dans NP-complet.

□

Considérons le graphe $H$ :




**Question 14.5**. Montrer que ce graphe ne possède pas de coloration propre du graphe utilisant au plus 2 couleurs.

Correction

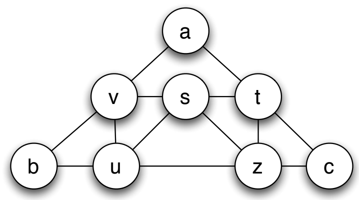
Il suffit de remarquer que le graphe $H$ possède un triangle. Et Il est impossible de colorier un triangle en 2 couleurs.

□

**Question 14.6**. Montrer que dans toute coloration propre du graphe utilisant au plus 3 couleurs, $a$, $b$ et $c$ ont la même couleur.

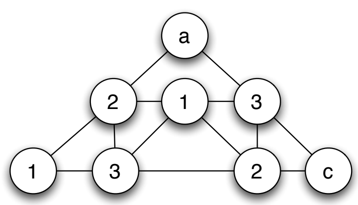
Correction

Commençons tout d'abord de nommer tous les sommets de graphe $H$ de la façon suivante :




Soit $C$ une coloration utilisant au plus 3 couleurs que l'on notera 1, 2, 3. Nous notons par $C(v)$ la couleur du sommet $v$. Sans perte de généralité, nous supposerons que $C(b) = 1$.

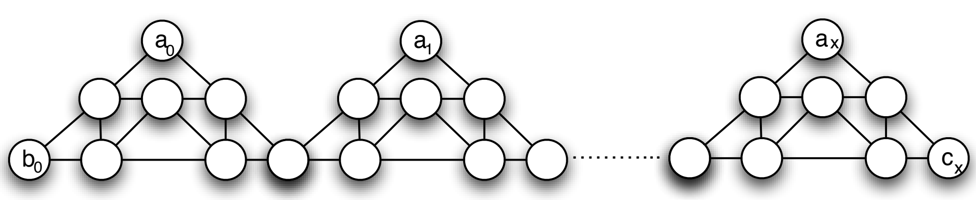
— Comme les sommets $b, u, v$ forment un triangle, ils sont tous de couleurs différentes. Sans perte de généralité, nous supposerons que $C(v) = 2$ et $C(u) = 3$.
— Les sommets $u, v, s$ forment un triangle et donc ils sont tous de couleurs différentes. Comme $C(v) = 2$ et $C(u) = 3$, on a $C(s) = 1$.
— Comme les sommets $s, u, v$ forment un triangle et comme $C(s) = 1$ et $C(u) = 3$, on a $C(z) = 2$
— Comme les sommets $s, t, z$ forment un triangle et comme $C(s) = 1$ et $C(z) = 2$, on a $C(t) = 3$
— Comme les sommets $c, t, z$ forment un triangle et comme $C(t) = 3$ et $C(z) = 2$, on a $C(c) = 1$
— Comme le sommet $a$ est incident aux sommets $v$ et au sommet $t$, on a $C(a) = 1$




Donc on a bien que dans toute coloration propre du graphe utilisant au plus 3 couleurs, $a$, $b$ et $c$ sont de même couleur.

□

Soit $x$ un entier strictement positif. Considérons le graphe $H_x$ suivant :




**Question 14.7**. Montrer que dans toute coloration propre du graphe $H$ utilisant au plus 3 couleurs, $(a_i)_{0\leq i<x}$, $(b_i)_{0\leq i\leq x}$, sont de même couleur.

Correction

Le graphe $H_x$ est $x$ copies de $H$. Par la question précédente a),on sait que pour tout $0 \leq i < x$, $a_i$, $b_{i-1}$, $b_i$ sont de même couleur.

□

**Question 14.8**. Montrer que le problème COLOR suivant est NP-complet :

**Entrée** : Un graphe $G' = (V', E')$ dont tous les sommets sont de degré au plus 4
**Question** : Existe-t-il une coloration propre des sommets utilisant moins de 3 couleurs ?

*Indications : pour chaque sommet de $v$, associer un graphe $H_x$.*

Correction

Il est clair que le problème COLOR est dans NP (car c'est un sous-problème du problème 3-coloration). Nous construisons l'instance du problème du COLOR à partir d'une instance du problème *3-coloration* composé du graphe $G = (V, E)$.

1. Numéroter les $m$ arêtes de $G$ tel que $E = \{e_i : 0 \leq i < m\}$
2. pour chaque sommet $v$ de $G$ faire
   (a) créer un graphe $G_v$ qui est un copie du graphe $H_m$
   (b) pour tout $0 \leq i < m$, les sommets $a_i$ sont nommé $(a_i, v)$
   (c) Insérer $G_v$ dans $G'$
3. pour chaque arête $e_\ell = (s, t)$ de $G$ faire
   (a) Créer une arête entre le sommet $(a_\ell, s)$ et sommet $(a_\ell, t)$ dans $G'$.

Il est clair le graphe $G'$ peut se construire en temps polynomial. Remarquons que tous ses sommets sont de degré inférieur où égal à 4.

Supposons qu'il existe une fonction $c : V \to \{1, 2, 3\}$ qui est une propre coloration de trois couleurs dans $G$. Nous allons contruire une coloration $c'$ qui est une propre coloration de trois couleurs dans $G'$ de la façon suivante :

1. pour chaque sommet $v$ de $G$
   (a) $c'(a_0, 0)) = c(v)$
   (b) Le sous-graphe $G_v$ de $G'$ est coloré en trois couleurs en utilisant les résultats des deux questions précédente.

La coloration $c'$ ainsi construire est une propre coloration de trois couleurs dans $G'$. En effet, s'il existe une arête entre le sommet $(a_\ell, s)$ et sommet $(a_\ell, t)$ dans $G'$, cela signifie qu'il existe $e_\ell = (s, t)$ dans $G$. Cela implique que $c'(a_\ell, s)) = c(s)$ et $c'(a_\ell, t)) = c(t)$ et que $c'(a_\ell, s)) \neq c'(a_\ell, t))$ (car $c(s) \neq c(t)$).

Supposons qu'il existe une fonction $c' : V' \to \{1, 2, 3\}$ qui est une propre coloration de trois couleurs dans $G'$. Nous allons contruire une coloration $c$ qui est une propre coloration de trois couleurs dans $G$ de la façon suivante :

1. pour chaque sommet $v$ de $G$
   (a) $c(v) = c'(a_0, 0))$

La coloration $c$ ainsi construire est une propre coloration de trois couleurs dans $G$. En effet, s'il existe une arête $e_\ell$ entre le sommet $s$ et $t$ dans $G$. Par construction, il existe une arête entre le sommet $(a_\ell, s)$ et sommet $(a_\ell, t)$ dans $G'$. Comme $c'(a_\ell, s)) \neq c'(a_\ell, t))$, que $c'(a_\ell, s)) = c(s)$, et que $c'(a_\ell, t)) = c(t)$, on peut en déduire que $c(s) \neq c(t)$.

Donc $G'$ possède une 3-coloration propre si et seulement si $G$ possède une 3-coloration propre.

Donc le problème COLOR est dans NP-complet. □

**Problème** *k-COULEUR*
**Données** : Un graphe $G = (V, E)$, et un entier $k$.
**Question** : Décider s'il existe un coloriage du graphe utilisant exactement $k$ couleurs ?

**Question 14.9**. Prouver que le problème 3-*COULEUR* est $NP$-complet.

Correction

Il suffit de remarquer que la preuve vu en cours pour prouver que le problème 3-*COLORIABILITE* est NP-complet, peut aussi s'appliquer à ce problème. □

**Question 14.10**. Prouver que le problème $k$-*COULEUR* est $NP$-complet.

Correction

Le problème $k$-COULEURS est dans NP, car la donnée des sommets colorés par chacune des $k$ couleurs constitue un certificat vérifiable en temps polynomial. On va réduire 3-COLORABILITE à $k$-COULEURS.

On se donne donc un graphe $G = (V, E)$. Nous allons construire un graphe $G' = (V', E')$ de la façon suivante. $G'$ est une copie de $G$ en ajoutant un graphe complet de taille $k - 3$. De plus, chaque sommet de cette clique est sommet de $V$ est voisin de tous les sommets de $G$. Plus formellement

— $V' = V \cup \{v_1, \dots v_{k-3}\}$ (nous ajoutons $k$ sommets disjoints
— $E' = E \cup \{(v, v_i) : v \in V, 1 \leq i \leq k - 3\} \cup \{(v_{i'}, v_i) : 1 \leq i < i' \leq k - 3\}$

Le graphe $G'$ peut se construire en temps polynomial.

Supposons que $G$ peut se colorer en 3 couleurs. Soit $C$ un tel coloriage. Le graphe $G'$ peut se colorer en $k$ couleurs : le sous-graphe de $G'$ correspondant au graphe $G$ se colorie en utilisant $C$. Il reste les sommets de la clique. On les colorie utilisant les couleurs de 4 à $k$

Supposons que $G'$ peut se colorer en $k$ couleurs. Soit $C$ un tel coloriage. Chaque sommet de la clique utilise une couleur. Aucun autre sommet utilise cette couleur. Cela implique que le coloriage $C$ utilise que 3 couleurs. □

Nous supposerons que le problème COUVERTURE DE SOMMETS est NP-complet

COUVERTURE DE SOMMETS
**Données** : un graphe non-orienté $G$ et un entier $k$.
**Question** : $G$ contient-il une *couverture de sommets* de cardinalité au plus $k$ : c'est-à-dire un ensemble $\mathcal{S}$ de sommets tel que toutes les arêtes de $G$ sont incidentes à au moins un sommet de $\mathcal{S}$ ?

## b) Le problème de la clique maximum.

Considérons le problème de décision CLIQUE :
**Données** : un graphe non-orienté $G = (V, E)$ et un entier $k$.
**Question** : Existe-t-il une clique de taille $k$ (un sous graphe complet de $k$ sommets) ?

**Question 15.1**. Nous noterons par $G^c$ le complémentaire du graphe $G$.

Montrer que $G$ a une clique de taille $k$ si et seulement si $G^c$ a une couverture de sommets de taille $n-k$.

**Question 15.2**. Montrer que le problème CLIQUE est NP-complet.

Nous allons travailler sur une restriction du problème CLIQUE en considérant uniquement les graphes dans lesquels tous leurs sommets sont de degré au plus 3. Nous le noterons 3-CLIQUE.

**Question 15.3**. Montrer que 3-CLIQUE est dans NP.

**Question 15.4**. Trouver l'erreur dans le raisonnement suivant : *Nous savons que le problème* CLIQUE *est NP-complet, il suffit donc de présenter une réduction de* 3-CLIQUE *à* CLIQUE*. Étant donné un graphe $G$ dont les sommets sont de degré inférieur à* 3*, et un entier $k$, la réduction laisse inchangée le graphe et le paramètre $k$ : clairement le résultat de la réduction est une entrée possible pour le problème CLIQUE. Par ailleurs, la réponse aux deux problèmes est identique. Cela prouve la justesse de la réduction et, par conséquent, la NP-complétude de la* 3-CLIQUE.

**Question 15.5**. Donner un algorithme polynomial en $O(|V|^4)$ pour le problème 3-CLIQUE.

## c) Problème du Set Cover

Rappellons la définition suivante : une *couverture de sommets* $S$ d'un graphe $G$ est un sous-ensemble de sommets tel que toutes les arêtes de $G$ sont incidentes à au moins un sommet de $S$.

Nous allons considérer le problème (voir figure 3.1 pour illustration) : Set Cover

**Données :** un ensemble $U$ de $u$ éléments, un ensemble $\mathcal{S}$ composé de sous-ensembles $S_1, \ldots, S_t$ de $U$, et un entier $k$.

**Question :** Existe-il au plus $k$ éléments de $\mathcal{S}$ tels que leur union est égale à $U$ ?

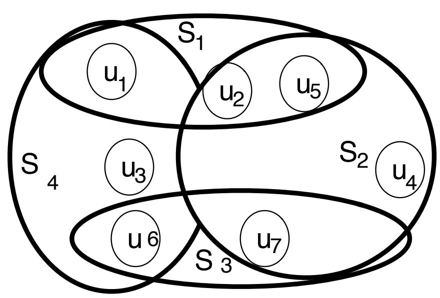



Figure 3.1 – $U = \{u_1, \ldots, u_7\}$, $S_1, S_2, S_4$ est un *set cover* de $U$ et de cardinalité 3.

**Question 16.1**. Montrer que ce problème est dans NP.

Correction

Le problème du *Set Cover* est dans NP car étant donné un ensemble $\mathcal{S}'$ composé d'éléments de $\mathcal{S}$ on peut vérifier en temps polynomial si

— $\mathcal{S}'$ possède $k$ éléments.

— chaque l'élément du $U$ est contenu dans un des éléments de $\mathcal{S}'$.

□

**Question 16.2**. Montrer que ce problème est dans NP-complet.

Correction

Nous allons faire la réduction à partir du problème du Vertex Cover. Rapellons la définition : Vertex Cover

**Données :** un graphe $G = (V, E)$ et un entier $j$.

**Question :** Existe-il une couverture sommet $C$ de cardinalité inférieure à égale à $j$ ?

Soit $\mathcal{I} =< G = (V, E), k >$ une instance du problème du Vertex Cover. Nous affectons $j$ à $k$ et nous construisons l'ensemble $U$ et un ensemble $\mathcal{S}$ de la façon suivante :

— $U = E$

— $\mathcal{S} = \bigcup_{v \in V} S_v$ où $S_v = \{e \in E | v \text{ est incident à l'arête } e\}$

Cette transformation se fait en temps polynomial car

— créer l'ensemble $U$ nécessite au pire $\mathcal{O}(|V|^2)$ opérations ;

— Pour chaque sommet $v$, créer l'ensemble $S_v$ nécessite au pire $\mathcal{O}(d_v)$ opérations où $d_v$ est le degré du sommet $v$ ;

— Au total, créer tous les ensembles $S_v$ nécessite au pire $\mathcal{O}(|V|^2)$ opérations.

Il reste à prouver les deux assertions suivantes :

1. Si $C$ est une couverture sommet de $G$ de cardinalité inférieur ou égale à $j$, alors

$$\mathcal{S} = \bigcup_{v \in C} S_v \text{ est un set cover de cardinalité inférieur ou égale à } j.$$

2. Si $\mathcal{S}$ est un set cover de cardinalité inférieur ou égale à $j$, alors

$C = \{v | S_v \in \mathcal{S}\}$ est une couverture sommet de $G$ de cardinalité inférieur ou égale à $j$.

□

### d) Le problème *Stable* est NP-complet

Considérons le problème de décision $S$table
**Données** : un graphe non-orienté $G = (V, E)$ et un entier $k$.
**Question** : Existe-t-il un sous-ensemble $V' \subseteq V$, avec $|V'| = k$, tel que $u, v \in V' \implies (u, v) \notin E$ ?

**Question 17.1**. Montrer que

1. si le graphe a une couverture de sommets de cardinalité $k$, alors il a un stable de cardinalité $n - k$ où $n$ est le nombre de sommets.
2. et réciproquement

Correction

Soit $G$ un graphe $G = (V, E)$

— Supposons que $S$ est un stable de $G$. Pour chaque arête $e = (u, v)$, au plus un sommet de ces sommets est dans $S$ et au plus un est dans $V - S$. Donc chaque arête a au plus une extrémité dans $V - S$. Donc $V - S$ est une couverture sommet de $G$,

— Supposons que $V - S$ est une couverture sommet de $G$. Nous allons prouver par contradiction que $S$ est un stable de $G$. Supposons que $S$ ne soit pas un stable. Cela signifie qu'il existe une arête $e = (u, v)$ dont les deux sommets extrémités sont dans $S$. Cela signifie que $e$ n'est pas couverte par $V - S$ et ceci contredit le fait que $V - S$ est une couverture sommet de $G$. Donc $S$ ne soit pas un stable.

$\square$

**Question 17.2**. Montrer que le problème $S$table est NP-complet en supposant que le problème $C$ouverture de sommets (VC) est NP-complet.
**Données** : un graphe non-orienté $G = (V, E)$ et un entier $k$.
**Question** : $G$ contient-il une *couverture de sommets* $\mathcal{S}$ de cardinalité au plus $k$ ?

Correction

1. $S$table est dans NP.

   Coder un graphe dépendant du nombre de sommets et d'arêtes. Donc la taille de l'instance est de l'ordre de $\mathcal{O}(n^2 \log n)$ où $n$ est le nombre de sommets.

   On peut vérifier en temps polynomial si un ensemble de sommets $S$ est un stable et si il est de cardinalité inférieur à $k$. En effet, il suffit de vérifier que pour chaque couple de sommets $S$ vérifier si ils ne sont pas voisins. Le nombre de vérifications nécessaire est $\mathcal{O}(|S|^2) = \mathcal{O}(n^2)$. Vérifier si il ne sont pas voisins nécessite $\mathcal{O}(1)$ opérations si le graphe est codé par une matrice d'adjacence ou $\mathcal{O}(n)$ opérations par une liste d'adjacence.

2. $S$table est NP-complet. Soit $(G, k)$ une instance du problème Couverture sommet. Posons que $n$ est le nombre de sommets de $G$. On peut construire un graphe $G'$ identique à $G$ Couverture sommet et affecter la valeur $n - k$ à $k$. Cela peut construire une instance du problème Stable à partir d'une instance du problème Couverture sommet en temps polynomial ($\mathcal{O}(n^2)$ opérations). La question précédente a permis de prouver que le fait suivant : Le graphe $G$ admet est une couverture de sommets $S$ de taille $k$ si et seulement si le graphe $G'$ admet a un stable de cardinalité $k' = (n - k)$ o Donc le problème est dans NP-complet.

$\square$

## VI) Le problème du $k$-centre

Dans cette partie, nous allons nous concentrer sur le problème du $k$-centre : étant donné un ensemble de villes dont les distances sont spécifiées, choisir $k$ villes afin d'installer des entrepôts de façon à minimiser la distance maximale d'une ville à l'entrepôt le plus proche. Un tel ensemble de $k$ villes est appelé $k$-centre.

Le problème associé de décision est le suivant $k$-CENTRE

**Données :** un graphe complet $K = (V, E)$ muni d'une fonction de poids $w$ sur les arêtes, et des entiers strictement positifs $k$ et $b$.

**Question :** Existe-il un ensemble $S$ de sommets tel que $|S| = k$ et tel que tout sommet $v$ de $V$ satisfait la condition suivante

$$min\{w(v, u) : u \in S\} \leq b$$

### a) Le problème ENSEMBLE DOMINANT est NP-complet

Rappellons les définitions suivantes :

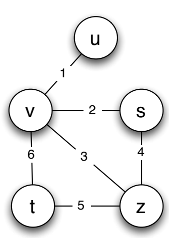
— une couverture de sommets $S$ du graphe $G$ est un sous-ensemble de sommets tel que toutes les arêtes de $G$ sont incidentes à au moins un sommet de $S$.

— un ensemble dominant $C$ du graphe $G$ est un sous-ensemble de sommets tel que tout sommet est soit dans $C$ soit voisin d'un sommet de $C$.

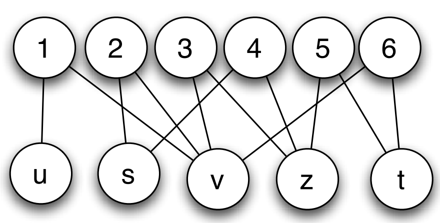
Soit $G$ un graphe $G = (V, E)$. Nous allons construire un graphe $G' = (V', E')$ à partir de $G$ tel que

— $V' = V \cup E$ ;

— $E' = E \cup \{(v, a) | v \in V, a \in E, v$ est extrémité de l'arête a dans $G\}$




Graphe $G$




Graphe $G'$ (les arêtes dans $E$ ne sont pas dessinées)

**Question 18.1**. Montrer que si $S$ est une couverture de sommets du graphe $G$, alors $S$ est un ensemble dominant de $G'$.

Correction

Soit $S$ une couverture de sommets du graphe $G$. Il faut prouver que tous les sommets du graphe soit soit voisin de $S$ ou soit dans $S$.

Toutes les arêtes de $G$ ont au moins un de ses extrémités dans $S$. Par conséquence, tous les sommets de $G$ sont soit dans $S$ ou soit voisins de $S$.

Comme toutes les arêtes de $G$ ont au moins un de ses extrémités dans $S$, tous les sommets correspondant à une arête de $G$ sont des voisins de $S$.

Donc $S$ est un ensemble dominant de $G'$. □

**Question 18.2**. Montrer que si $S'$ est un ensemble dominant de $G'$, alors il existe un ensemble $S$ de même cardinalité de $S'$ qui est une couverture de graphe $G$,

Correction

Si $S' \subseteq V$, alors tous les sommets du graphe $G$ sont soit voisin de $S$ ou soit dans $S$. Donc toutes les arêtes ont une de ces extrémités dans $S$.

Donc $S'$ est une couverture de graphe $G$,

Sinon, il existe un sommet $a$ dans $S'$ qui n'appartient pas à $V$ mais qui appartient à $V'$. Cela signifie que le sommet $a$ représente une arête $(u, v)$ dans $G$ et que couvre uniquement les arêtes $(a, u)$ et $(a, v)$ dans $G'$. Considérons un sous-ensemble $S''$ tel que

$$S'' = (S' \cap V) \cup \{u : a = (u, v) \wedge a \in S'\}$$

Remarquons que $|S''| = |S'|$. Soit $a \in S' \backslash V$. Comme le sommet $a$ arêtes $(a, u)$ et $(a, v)$ dans $G'$, alors le sommet $u$ couvre les même arêtes. Donc $S''$ est une couverture de graphe $G$.

□

**Question 18.3**. Exprimer le problème de minimisation de l'ensemble dominant sous forme de problème de décision.

Correction

*Données* : un graphe non-orienté $G$, et un entier $k$

*Question* : Existe-t-il un ensemble dominant $S$ de $G$ tel que $|S| \leq k$ ?

□

**Question 18.4**. Montrer que ce problème est dans NP.

Correction

On peut vérifier en temps polynomial si un ensemble de sommets est un ensemble dominant et si il est de cardinalité inférieur à $k$. □

**Question 18.5**. Montrer que ce problème est dans NP-complet.

Correction

Soit $(G, k)$ une instance du problème Couverture sommet. On peut contruire en temps polynomiale le graphe $G'$.

Les questions précédentes ont permis de prouver que le fait suivant : Le graphe $G$ admet est une couverture de sommets $S$ de taille $k$ si et seulement si le graphe $G'$ admet est un ensemble dominant de taille $k$. Donc le problème est dans NP-complet. □

## b) Le problème $k$-centre est NP-Complet

**Question 19.1**. Montrer que le problème $k$-centre est dans NP.

Correction

On peut vérifier en temps polynomiale si $S$ est de cardinalité □

**Question 19.2**. Montrer que $k$-centre est NP-complet sachant que DOMINANT est NP-complet.

### c) Le problème $k$-CENTRE est in-approximable

Correction

On peut vérifier en temps polynomiale si $S$ est de cardinalité □

**Question 20.1**. Montrer que si $P \neq NP$, quel que soit $\epsilon > 0$, il n'existe pas de $(2-\epsilon)$-approximation pour le problème du $k$-CENTRE MÉTRIQUE.

Correction

Il faut réduire le problème du dominant minimum à celui de l'approximation du problème du $k$-CENTRE MÉTRIQUE. Soit $I$ une instance du problème du dominant minimum : $G = (V, E)$ et un entier $k$.

Nous construisons l'instance du problème du $k$-CENTRE MÉTRIQUE : le graphe complet $K = (V, E')$ muni de la fonction de poids sur les arêtes suivantes :

$$w(u,v) = \begin{cases} 1 & si \quad (u,v) \in E \\ 2 & si \quad (u,v) \notin E \end{cases}$$

Remarquons que la fonction de poids vérifie l'inégalité triangulaire.

Cette réduction se réalise en temps polynomiale et elle vérifie les propriétés suivantes :

— $dom(G) \leq k$ alors $K$ admet un k-centre de coût 1 ;

— $dom(G) > k$ alors $K$ admet un k-centre de coût 2 ;

Si $dom(G) \leq k$, la solution optimale est de coût 1. Une $(2-\epsilon)$approximation pour le $k$-centre donne nécessairement une solution de coût $1 * (2-\epsilon)$. Puisqu'elle ne peut pas utiliser d'arête de coût 2, elle retourne une solution de coût 1.

Utiliser un tel algorithme permettrait de distinguer entre ces deux cas, et de résoudre le problème du dominant minimum en temps polynomial. Et cela supposerait que $P = NP$. □

**Question 20.2**. Montrer que quel que soit $\alpha(n)$ calculable en temps polynomial, il n'existe pas de $\alpha(n)$-approximation pour $k$-CENTRE, à moins que $P = NP$.

Correction

Par l'absurde, soit A une $\alpha(n)$-approximation (en temps polynomial) pour $k$-CENTRE. Nous allons démontrer que $A$ permet de résoudre le problème NP-difficile du dominant minimum en temps polynomial, et donc que $P = NP$. Soit $I$ une instance du problème du dominant minimum : $G = (V, E)$ et un entier $k$.

Nous construisons l'instance du problème du $k$-CENTRE : le graphe complet $K = (V, E')$ muni de la fonction de poids sur les arêtes suivantes :

$$w(u,v) = \begin{cases} 1 & si \quad (u,v) \in E \\ \alpha(n)+1 & si \quad (u,v) \notin E \end{cases}$$

Cette réduction se réalise en temps polynomiale et elle vérifie les propriétés suivantes :

— $dom(G) \leq k$ alors $K$ admet un k-centre de coût 1 ;

— $dom(G) > k$ alors $K$ admet un k-centre de coût $\alpha(n) + 1$ ;

Si $dom(G) \leq k$, la solution optimale est de coût 1. Une $\alpha(n)$-approximation pour le $k$-centre donne nécessairement une solution de coût $1 \cdot \alpha(n)$ qui ne peut pas utiliser d'arête de coût $\alpha(n) + 1$, elle retourne une solution de coût 1.

Utiliser un tel algorithme permettrait de distinguer entre ces deux cas, et de résoudre le problème du dominant minimum en temps polynomial. Et cela supposerait que $P = NP$.

$\square$

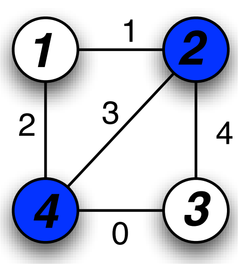
# VII) Problèmes encodant les entiers

## a) Le problème *SOMME DE SOUS-ENSEMBLE* est NP-complet

Nous considérons le problème *SOMME DE SOUS-ENSEMBLE*
**Données** : un ensemble fini d'entiers $A = \{a_1, a_2, \ldots, a_\ell\}$ et un entier $t \in \mathbb{N}$
**Question** : Existe-t-il un sous-ensemble $A' \subseteq A$ tel que $\sum_{a \in A'} a = t$ ?
Nous supposerons que le problème *C*ouverture de sommets (VC) est NP-complet.
**Données** : un graphe non-orienté $G = (V, E)$ et un entier $k$.
**Question** : $G$ contient-il une *couverture de sommets* $\mathcal{S}$ de cardinalité au plus $k$ ?

Considérons la transformation suivante
- On se donne un graphe $G = (V, E)$ et un entier $k$.
- Pour cela,
  - on numérote les sommets entre 0 et $n-1$ et des arêtes entre 0 et $m-1$.
  - on associe un entier $b_{ij}$ pour chaque couple (arête, sommet)
$$b_{ij} = \begin{cases} 1 & \text{si l'arête } i \text{ est incidente au sommet } j, \\ 0 & \text{sinon} \end{cases}$$
- On construit l'ensemble $A$ de la façon suivante : $(b = 4)$
  - Pour chaque sommet $j$ : $a_j = b^m + \sum_{i=0}^{m-1} b_{ij} b^i$
  - Pour chaque arête $j$ : est associé l'entier $b^j$
- On construit l'entier $t$ : $t = \underbrace{kb^m}_{cardinalite\ de\ la\ couverture} \sum_{i=0}^{m-1} \underbrace{2b^i}_{cout\ de\ l'\ arete\ i}$

**Question 21.1**. Calculer l'instance du problème *SOMME DE SOUS-ENSEMBLE* construite à partir de l'instance suivante de *VC* $\mathcal{I} = (G, k)$ :




Correction

L'ensemble $A$ est composé des entiers suivants :
- $a_1 = b^5 + b^1 + b^2$ (correspondant au sommet 1)
- $a_2 = b^5 + b^1 + b^3 + b^4$ ( sommet 2)
- $a_3 = b^5 + b^0 + b^4$ ( sommet 3)
- $a_4 = b^5 + b^0 + b^2 + b^3$ ( sommet 4)
- $b^0$, $b^1$, $b^2$, $b^3$, $b^4$ (les entiers correspondants aux arêtes)

□

**Question 21.2**. Montrer que si $G$ admet une couverture sommet $S$ de cardinalité $k$ alors il existe un sous-ensemble $A' \subseteq A$ tel que $\sum_{a\in A'} a = t$

**Question 21.3**. Montrer que s' il existe un sous-ensemble $A' \subseteq A$ tel que $\sum_{a\in A'} a = t$ alors $G$ admet une couverture sommet $S$ de cardinalité $k$

### b) Le problème *SOMME-DES-CARRES* est NP-complet

Nous considérons le problème *SOMME-DES-CARRES*
**Données** : Un ensemble fini d'entiers $A$, deux entiers positifs $k$ et $b$
**Question** : Décider si l'ensemble $A$ peut se partitionner en $k$ sous-ensembles disjoints $A_1, \ldots, A_k$ tels que

$$\sum_{i=1}^{k}\left(\sum_{x\in A_i} x\right)^2 \leq b$$

Nous supposerons que le problème *PARTITION* est NP-complet. **Problème** *PARTITION*
**Données** : Un ensemble fini d'entiers $A$.
**Question** : Décider s'il existe un sous-ensemble $A' \subset A$ tel que $\sum\limits_{x\in A'} x = \sum\limits_{x\in A\setminus A'} x$ ?

**Question 22.1**. Soit $A$ un ensemble d'entiers et $A'$ un sous ensemble de $A$. Montrer que

$$(\sum_{x\in A'} x)^2 + (\sum_{x\in A\setminus A'} x)^2 \geq \frac{\left(\sum_{x\in A} x\right)^2}{2}$$

Correction

Soit $a$ et $b$ deux entiers.
Tout d'abord nous allons prouver que $a^2 + b^2 \geq \frac{(a+b)^2}{2}$ ou que $a^2 + b^2 - \frac{(a+b)^2}{2} \geq 0$.

$$\begin{aligned} a^2 + b^2 - \frac{(a+b)^2}{2} &= \frac{a^2 + b^2 - 2ab}{2} \\ &= \frac{(a-b)^2}{2} \\ &\geq 0 \end{aligned}$$

En posant $a = \sum\limits_{x\in A'} x$ et $b = \sum\limits_{x\in A\setminus A'} x$, la précédent inégalité permet de déduire que $(\sum\limits_{x\in A'} x)^2 + (\sum\limits_{x\in A\setminus A'} x)^2 \geq \frac{\left(\sum_{x\in A} x\right)^2}{2}$. □

**Question 22.2**. Soit $A$ un ensemble d'entiers et $A'$ un sous ensemble de $A$. Montrer que si $(\sum\limits_{x\in A'} x)^2 + (\sum\limits_{x\in A\setminus A'} x)^2 = \frac{\left(\sum_{x\in A} x\right)^2}{2}$ alors $A'$ est tel que $\sum\limits_{x\in A'} x = \sum\limits_{x\in A\setminus A'} x$

Correction

Soit $a$ et $b$ deux entiers tels que $a \geq b + 1$.

Tout d'abord nous allons prouver que $a^2+b^2 > (a-1)^2+(b+1)^2$ si $a > b+1$.

$$\begin{aligned} a^2+b^2 =& (a-1)^2+1+2a+(b+1)^2+1-2b \\ =& \quad (a-1)^2+(b+1)^2+2+2a-2b \\ >& \qquad (a-1)^2+(b+1)^2 \end{aligned}$$

La première équation est obtenue en posant $a = a-1+1$ et $b = b+1-1$ et en utilisant les identités remarquables. La dernière est obtenue en constatant que $a > b+1$ et que $2+2a-2b > 0$.

On peut donc déduire

1. que si $a+b$ est pair alors $a^2+b^2 \geq (\frac{a+b}{2})^2+(\frac{a+b}{2})^2$ et que la borne est atteinte quand $a=b$.
2. que si $a+b$ est impair alors $a^2+b^2 > (\frac{a+b+1}{2})^2+(\frac{a+b}{2})^2$ et que la borne est atteinte quand $a=b+1$.

En posant $a = \sum\limits_{x\in A'} x$ et $b = \sum\limits_{x\in A\setminus A'} x$, on peut déduire que $(\sum\limits_{x\in A'} x)^2 + (\sum\limits_{x\in A\setminus A'} x)^2 = \frac{(\sum_{x\in A} x)^2}{2}$ alors $\sum\limits_{x\in A'} x = \sum\limits_{x\in A\setminus A'} x$ □

**Question 22.3**. Donner la taille d'une instance du problème *SOMME-DES-CARRES* en supposant que les entiers sont codés en binaire.

Correction

Posons $b_{max} = max\{a, b : a \in A\}$. La taille de l'instance est $O(|A|\log b_{max})$. En effet, coder un nombre entier (étiquette du sommet) entre 0 et $n-1$, il faut $O(\log n)$ bits).

□

**Question 22.4**. Montrer que le problème *SOMME-DES-CARRES* est dans NP.

Correction

Le problème *SOMME-DES-CARRES* est dans NP. En effet, on peut calculer en temps polynomial que les ensembles $A_1, \ldots, A_k$ partitionnent l'ensemble $A$ et que $\sum\limits_{i=1}^{k}\left(\sum\limits_{x\in A_i} x\right)^2 \leq b$.

1. Décider si $A_1, \ldots, A_k$ partitionnent l'ensemble $A$ peut se faire en $O(|A|)$ opérations.
2. Fixons un $k$. Calculer $\sum_{x\in A_k} x$ nécessite $O(|A_k|b_{max})$ opérations. Et calculer $\left(\sum_{x\in A_k} x)^2\right)$ nécessite $O(|A_k|b_{max}+b_{max}^2)$ opérations.
3. Comparer les deux valeurs $\sum\limits_{i=1}^{k}\left(\sum\limits_{x\in A_i} x\right)^2$ et $b$ nécessite $O(|b_{max}|)$ opérations.

□

**Question 22.5**. Montrer que le problème *SOMME-DES-CARRES* est NP-complet.

Correction

D'après la question précédente, le problème *SOMME-DES-CARRES* est dans NP.

Maintenant il faut montrer que le problème partition peut se réduire au problème *SOMME-DES-CARRES*

Pour construire une instance $(A)$ de *Partition* en une instance de *SOMME-DES-CARRES*, il suffit simplement de considérer le même ensemble d'entiers et de fixer les valeurs de $k$ et de $b$ : $k = 2$ et $b = \left(\frac{\sum_{x \in A} x}{2}\right)^2$. Cette construction se fait en temps polynomiale. De plus d'après les questions précédentes on peut déduire que

$$\sum_{x \in A'} x = \sum_{x \in A \setminus A'} x \iff (\sum_{x \in A'} x)^2 + (\sum_{x \in A \setminus A'} x)^2 \leq \frac{(\sum_{x \in A} x)^2}{2}$$

$\square$