Chloé DESDOUITS
Guillerme DUVILLIE
Swan ROCHER

M1 Informatique

Projet Transversal - Coloration de Graphes

2012

UM2

Table des matières

Chapitre 1

Introduction

1.1 Remarques

1.1.1 Connexité

Dans la suite, tous les graphes seront supposés connexes, sauf si l'inverse est précisé. En effet, dans le cas où ce ne serait pas le cas, les preuves / algorithmes peuvent s'appliquer à chaque sous graphe correspondant aux composantes.

Chapitre 2

Propriétés Générales

2.1 Exercice 1 - Coloration des Sommets

Considérons H un sous-graphe de G, il paraît évident qu'une coloration de G est aussi une coloration de H. De plus, par le théroème de Brooks, si le degré maximal du sous graphe H est inférieur à $\chi(G)$, alors $\chi(H) \leq \Delta \leq \chi(G)$.

D'où, la propriété suivante :

Propriété 2.1.1 *Si H est un sous-graphe de G alors $\chi(H) \leq \chi(G)$.*

2.2 Exercice 2 - Coloration des Sommets

Chaque composante connexe de G peut être vue comme un sous-graphe de ce dernier. Donc d'après la propriété 2.1.1, on a : $\chi(G) \geq \max\{\chi(C), C$ composante connexe de $G\}$. Appelons $C_1, C_2, C_3, \ldots, C_k$ les composantes connexes de G. Pour $1 \leq i \leq k$, appelons c_i la coloration de la composante C_i avec les couleurs $1, 2, \ldots, \chi(C_i)$ et c définie pour tout v sommet de G par $c(v) = c_i(v)$ si $v \in C_i$, dans la mesure où il n'existe aucune arête entre deux composantes connexes de G, c est une coloration de G, ce qui implique $\chi(G) \leq \max\{\chi(C_i)\}$. On en déduit donc la propriété suivante :

Propriété 2.2.1 $\chi(G) = \max\{\chi(C), C$ *composante connexe de* $G\}$

2.3 Exercice 3 - Coloration des Sommets

Deux sommets de même couleur ne sont reliés entre eux par aucune arête [1]. Ainsi tous les nœuds de la même couleur n'ont aucune arête entre eux et forment donc un stable. Ainsi, un graphe $k-$coloriable peut être divisé en k sous-ensembles des sommets formant des stables et donc on en déduit la propriété suivante :

Propriété 2.3.1 *Rechercher un $k-$stable est équivalent à montrer l'existence d'un graphe $k-$coloriable.*

1. Par définition de la coloration

2.4 Exercice 4 - Coloration des Sommets

Montrons que : $G = (V, E)$ est biparti $\leftrightarrow G$ est 2-colorable.
Un graphe biparti est défini de la manière suivante :

- $V = V_1 \cup V_2$ avec $V_1 \cap V_2 = \emptyset$,
- $\forall x_1, y_1 \in V1(x_1, y_1) \notin E$,
- $\forall x_2, y_2 \in V2(x_2, y_2) \notin E$.

$\rightarrow$ Soit $G = (V = V_1 \cup V_2, E)$.

Soit la coloration C suivante :

- $C(x) = 1 \forall x \in V_1$,
- $C(x) = 2 \forall x \in V_2$.

Par définition d'un graphe biparti, cette 2-coloration est bien valide (puisque il n'existe aucune arête à l'intérieur de chaque partition).

$\leftarrow$ Soit $G = (V, E)$ un graphe 2-colorable.
Soit C une 2-coloration valide de ses sommets.
On pose $V_1 = \{x \in V : C(x) = 1\}$ et $V_2 = \{x \in V : C(x) = 2\}$. Puisque cette coloration est valide $\forall x, y \in V$ si $C(x) = C(y)$ alors $(x, y) \notin E$.
Ainsi on peut noter $V = V_1 \cup V_2$, leur intersection est bien vide (puisque un même sommet ne peut pas avoir deux couleurs différentes) et il n'existe aucune arête ayant ses deux extrémités dans le même sous-ensemble de sommets.
G est donc un graphe biparti.

2.5 Exercice 5 - Coloration des Sommets

Soit A le plus grand stable d'un graphe quelconque G à n sommets. On note $\alpha(G) = |A|$.
Montrons que $\lceil \frac{n}{\alpha(G)} \rceil \leq \chi(G) \leq n - \alpha(G) + 1$. On pose (i) la première inégalité et (ii) la seconde.

Montrons (i) par récurrence sur la taille $\alpha(G)$ de A.
Si $\alpha(G) = 1$, on a : $\lceil \frac{n}{1} \rceil = n \leq \chi(G) \leq n$.
On suppose que (i) est vraie pour $\alpha(G) = c$ et on cherche à prouver que c'est toujours le cas pour $\alpha(G) = c+1$. Il est évident que $\lceil \frac{n}{c+1} \rceil \leq \lceil \frac{n}{c} \rceil$, or par hypothèse on a : $\lceil \frac{n}{c} \rceil \leq \chi(G)$.
(i) est donc vérifiée.

Montrons maintenant (ii) :
On sait que $\chi(G) \leq n$ (il suffit d'attribuer une couleur différente à chaque sommet).
Soit $G' = G \setminus A$, on voit que le nombre de sommets de G' est égal à $n - \alpha(G)$, et on en déduit que $\chi(G') \leq n - \alpha(G)$. A étant stable, on a $\chi(A) = 1$.
De plus, $\chi(G) \leq \chi(G') + \chi(A)$, et donc on peut conclure :
$\chi(G) \leq n - \alpha(G) + 1$.

2.6 Exercice 6 - Coloration des Sommets

On cherche à montrer que le graphe complet à n sommets K_n est n-colorable.
Soit la coloration C suivante : $C(x_i) = i \ \forall i \in [1, n]$.
Puisque chaque sommet a une couleur différente, C est valide, de plus puisque K_n possède n sommets, n couleurs ont été utilisées.
K_n est bien n-colorable.

Chapitre 3

Complexité

3.1 Exercice 11 - Coloration des sommets d'un graphe et coloration des arêtes d'un graphe

Un graphe est $2-$coloriable si et seulement si il est possible de partitionner ce dernier en deux ensembles stables. Un graphe 2-coloriable est donc un graphe biparti, or le problème de décider si un graphe est biparti est un problème polynomial, le problème de la $2-$coloration est donc un problème polynomial.

Un graphe est $2-$arête$-$coloriable s'il l'ensemble V de ses arêtes est divisible en deux ensembles F et E distincts vérifiant :

1. $E \cup F = V$
2. $E \cap F = \emptyset$
3. $\forall((i,j),(k,l)) \in E^2, \quad i \neq k, \neq l, j \neq k, j \neq l$
4. $\forall((i,j),(k,l)) \in J^2, \quad i \neq k, \neq l, j \neq k, j \neq l$

E et F sont alors des couplages de G. Un algorithme résolvant ce problème consiste en la recherche d'un couplage maximum C puis en vérifiant que V C est aussi un couplage. Si tel est le cas, le graphe est $2-$arête$-$coloriable.

La recherche d'un couplage se faisant en temps polynomial, l'algorithme énoncé cidessus est polynomial.

3.2 Exercice 12 - Coloration des sommets d'un graphe

3.2.1 Appartenance à NP

1. COLOR :
   Considérons une solution, un moyen de tester s'il s'agit d'une solution réalisable est de tester pour chacun des sommets si ses voisins sont d'une couleur différente en gardant le s couleurs rencontrées en mémoire afin de vérifier que leur nombre n'excède pas k. La vérification se fait alors en $O(n^2)$ et est donc polynomiale.

2. 3-COLOR :
   On procède de la même manière que pour COLOR en s'assurant que le nombre de couleur ne dépasse pas 3, la vérification est alors aussi polynomiale
3. 3-COLOR-PLAN :
   La vérification de la validité de la $3-$coloration se fait comme précédemment en $O(n^2)$. Il fait cependant tester si le graphe est planaire qui est un problème polynomial, la vérification est don cpolynomiale.

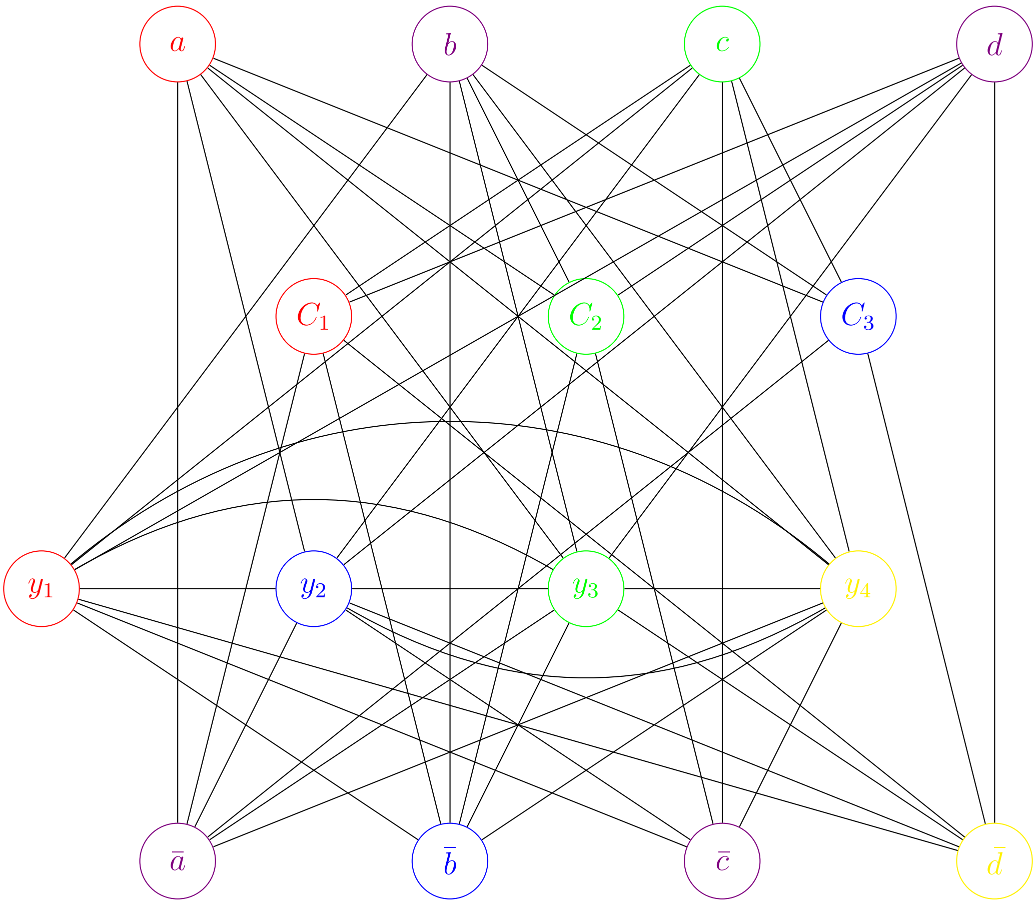
Tous ces problèmes appartiennent donc à NP.

3.2.2 Relation enre 3-COLOR et COLOR

Supposons qu'il existe un algorithme résolvant COLOR en temps polynomial quelque soit k, il existe alors un algorithme polynomial résolvant 3-COLOR. Par contraposition, on vient de démontrer que si 3-COLOR est NP-difficile alors COLOR est NP-difficile.

3.2.3 Étude du problème COLOR

Graphe G_ϕ



ϕ satisfiable $\Rightarrow n+1-$coloration de G_ϕ

Nous allons chercher à montrer que la coloration donnée est une coloration valide. Prenons les différents points un à un :

- les y_i sont reliés les uns avec les autres mais sont tous de couleur différente, et ils sont reliés aux $x_j et \bar{x}_j$ tels que $i \neq j$, or ces derniers sont soit de la couleur j, donc différente de la couleur i, soit de la couleur $n+1$ donc différente aussi
- les C_k sont de la couleur d'un des x_i la satisfaisant et sont reliés aux x_j et $\bar{x}_j$ n'apparaissant pas dans cette dernière, or ces derniers sont soit de couleur $n+1$ soit de couleur j différente de la couleur de C_k [1].
- les x_i sont reliés aux $\bar{x}_i$, or si un x_i est de couleur i, $\bar{x}_i$ est de couleur $n+1$ et vice-versa, ils sont donc d'une couleur différente

Nous avons passé en revue l'ensemble des arêtes de G_ϕ sans trouver de failles, une solution de ϕ nous donne une coloration de G_ϕ.

Couleurs des y_i

Tous les y_i sont adjacents les uns aux autres, ils forment alors un graphe complet. Or le nombre chromatique d'un graphe complet à n sommets et n et tous les sommets ont des couleurs différentes. Donc si le graphe est $n+1-$colorié, tous les y_i ont une couleur différente.

Couleurs des C_k

Le nombre de variables est supérieur à 3, donc il y a forcément au moins une variable n'apparaissant pas dans la clause k^e clause, nous appellerons x_i l'une de ces variables. Par définition x_i et $\bar{x}_i$ sont reliés à C_k, or l'une des deux variables est de couleur $n+1$ donc C_k ne peut être de couleur $n+1$.

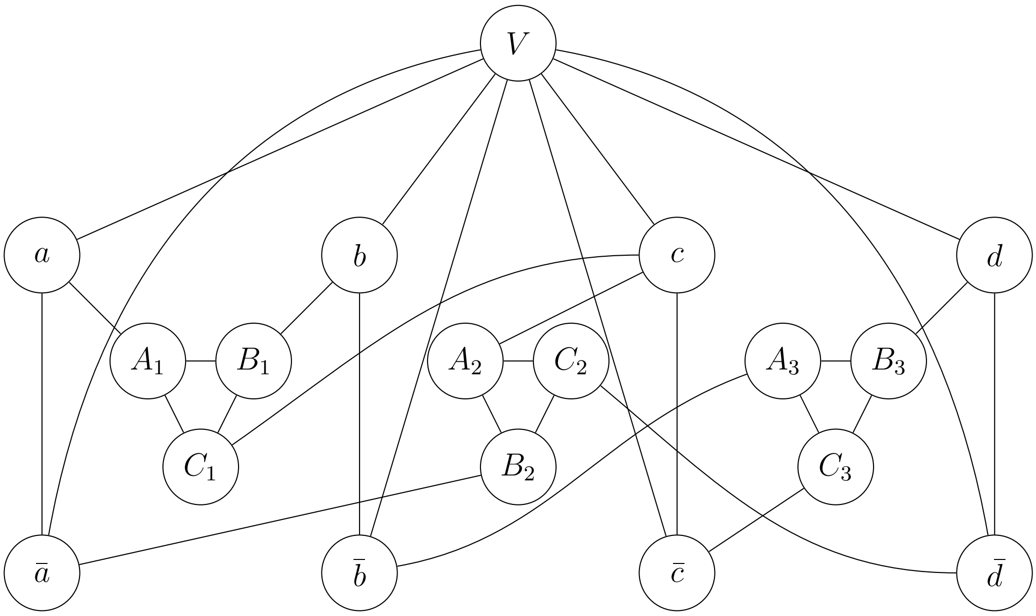
1. Imaginons qu'un des x_j soit de la couleur de C_k, ceci n'est possible que si $j = i$ et donc que si le nœud $_i = \bar{x}_j$, or si x_i a donné sa couleur à C_k c'est qu'il satisfait cette dernière et donc que x_j est de couleur $n+1$, il ne peut donc pas être de la même couleur que C_k

G_ϕ $n+1-$**coloriable** $\Rightarrow$ ϕ**est satisfaisable**

NP**-complétude de COLOR**

3.2.4 Étude de 3-COLOR

Construction du graphe



$\phi \in$ **NAE-3-SAT** $\Rightarrow$ **3-coloration de** G_ϕ

Nous allons cherhcer à démontrer que la coloration obtenue est une coloration valide. pour ce faire, prenons les différents points un à un :

- les x_i sont reliés au $\bar{x}_i$, or si x_i est vraie le sommet est coloré en rouge, $\bar{x}_i$ est alors coloré en noir et inversement. Les deux n'ont donc pas la même couleur.
- V, coloré en vert, est rélié aux variables, colorées en rouge ou en noir. Les sommets n'ont donc pas la même couleur
- utilisons, pour les triangles de clauses, le fait que $\phi \in$ NAE-3-SAT. Ceci indique qu'à lintérieur d'une même clause, toutes les variables ne peuvent avoir la même valeur. On aura donc, quelque soit le cas, un sommet du triangle relié à une variable colorée en noir (respectivement rouge) et les autres sommets du triangle reliés à des variables colorées en rouge (respectivement noir). Chaque sommet du triangle est alors relié à trois sommets dont les couleurs sont différentes de la sienne.

Il s'agit alors d'une coloration valide.

Les triangles de clauses

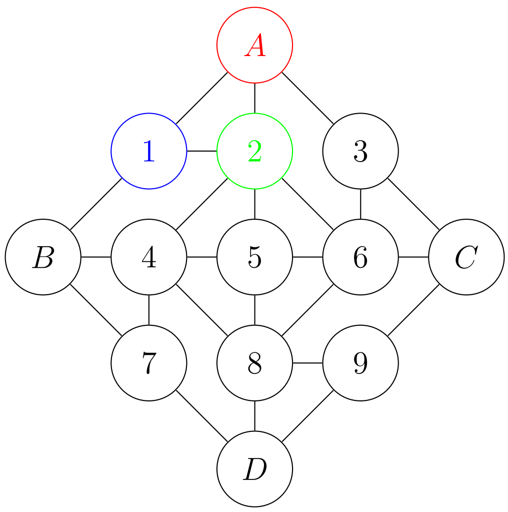
Supposons G_ϕ coloré avec 3 couleurs et supposons que l'un des triangles de clauses soit relié à trois sommets de couleur identique. Le triangle étant un graphe complet de taille 3, le nombre de couleurs pour le colorer est de 3, si la couleur des trois sommets reliés au triangle est rouge, le sommet du triangle de couleur rouge sera relié à un sommet rouge, il ne s'agit alors plus d'une coloration valide. On peut faire le même raisonnement sur le vert et le noir. Les trois sommets reliés au triangle ne peuvent être de couleur identique.

De plus, le fait que V soit coloré en vert et relié à chacune des variables imposent à ces dernières d'être soit noires, soit rouges. Et donc le triangle est relié à au moins un littéral de couleur noire et au moins un de couleur rouge.

3.2.5 3-COLOR-PLAN

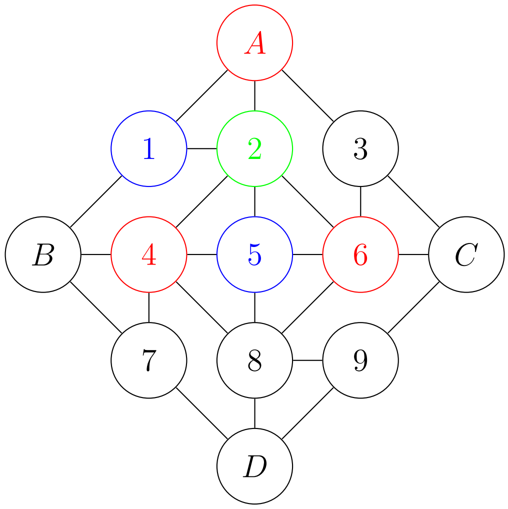
Couleur des sommets deux à deux

Nous allons montrer qu'il n'existe qu'une seule manière de colorer ce graphe avec 3 couleurs [2]. Pour ce faire commençons par rermarquer que chaque point appartient à au moins une clique de taille trois, ce qui impose la couleur de ses voisins. Donc quelques soit le point de départ, il existera une unique façon de colorer les voisins de ce point appartenant à la clique de taille 3, eux même imposant la couleur de leur voisin et ainsi de suite. Afin de démontrer l'imossibilité de colorer d'une autre manière nous ne réaliserons aucune coloration arbitraire de nœud, si ce n'est la coloration initiale. Commençons donc par le point A coloré en rouge. Ce dernier appartien à une clique de taille 3 avec les sommets 1 et 2, ils doivent donc avoir une couleur différente, 1 sera bleu et 2 sera vert.

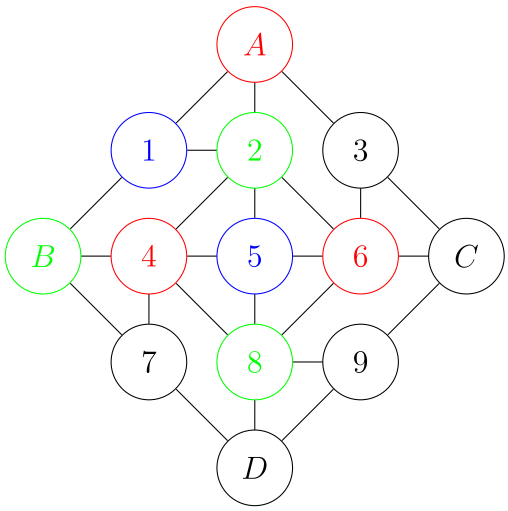


On voi alors que 2 appartient à deux cliques de taille 3, on colore donc ces dernières.

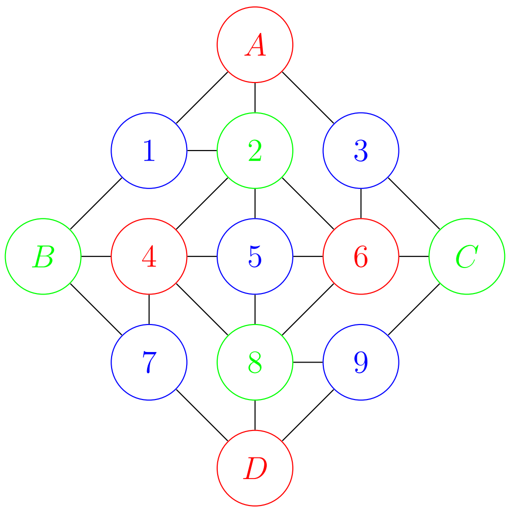
2. à permutation des couleurs près



Le sommet 5 appartient à une clique, et le sommet B n'a qu'une seule couleur possible, on réalise donc les colorations nécessaires.



La couleur du sommet 7 est obligatoirement bleue, ce qui obliqe D à être rouge, puis 9 à être bleu, C vert et enfin 3 bleu.

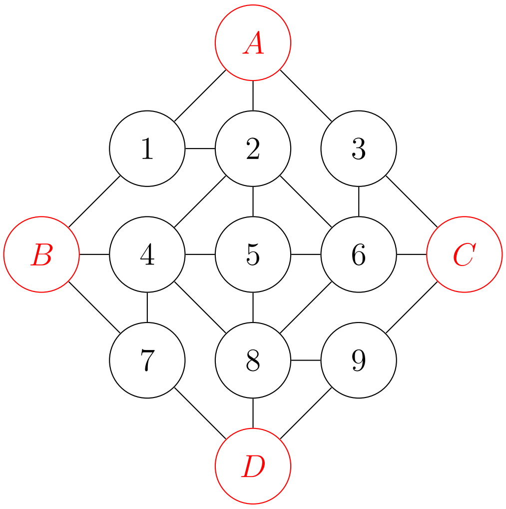


L'unicité de cette coloration (à permutation près) est sue à l'unicité de la coloration du sous graphe formé par les sommets 2, 4, 5, 6 et 8. Et on a bien les couleur de A et D (respectivement B et C) appareillée.

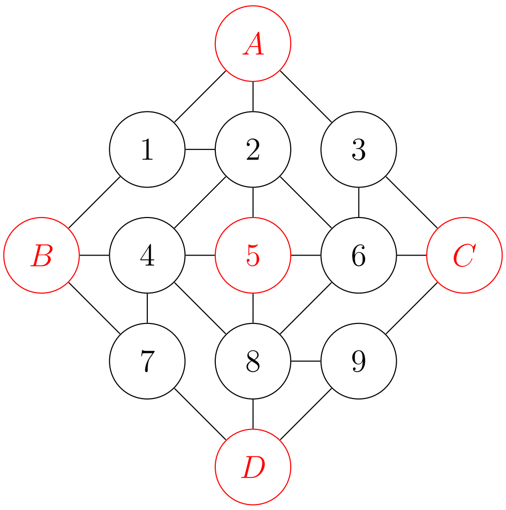
Coloration avec les couleurs des extrêmités données

Comme annoncé précédemment, lorsque les couleurs sont fixées, il n'y a plus de choix possibles, toutes les possibilités de coloration sont fixées.

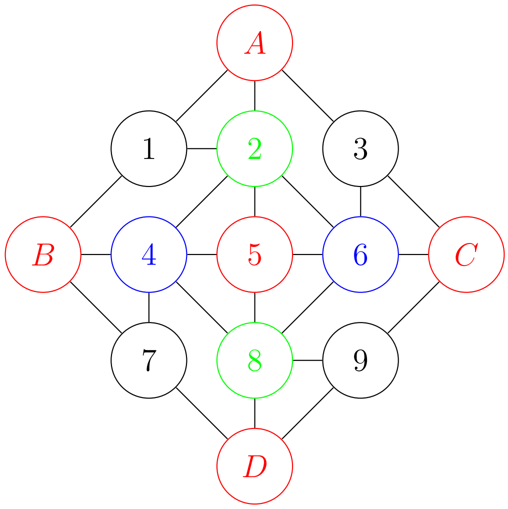
Prenons un nouvel exemple, le cas ou A et B sont de couleur différente ayant été présenté précédemment, nous et fixons A, B, C et D à rouge.



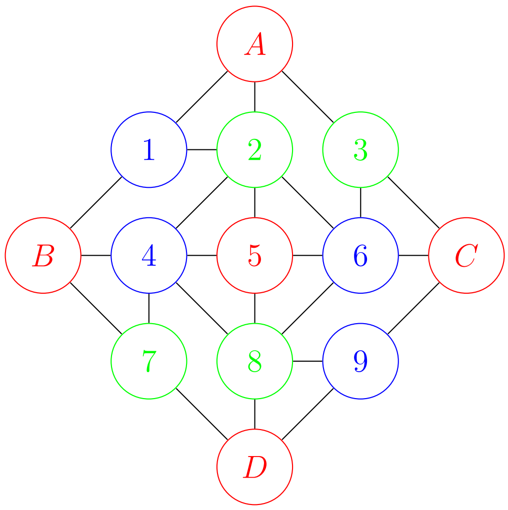
Le sommet 5 est obligatoirement rouge lui aussi.



On a alors 2 et 8 de même couleur, ainsi que 4 et 6 et 2 et 4 de couleur différente.



Toutes les coueurs des autres nœuds sont alors fixées.



NP-complétude de 3-COLOR-PLAN

Considéros un grapge G et H la transformation de G réalisée de la sorte : pour tout couple d'arêtes (i, j) et (k, l) s'intersectant de G, on insère au point d'intersection le graphe étudié précédemment de sorte que j soit confondu avec C, l'arête (i, j) devient (i, B), l soit confondu avec D et (k, l) devient (k, A).

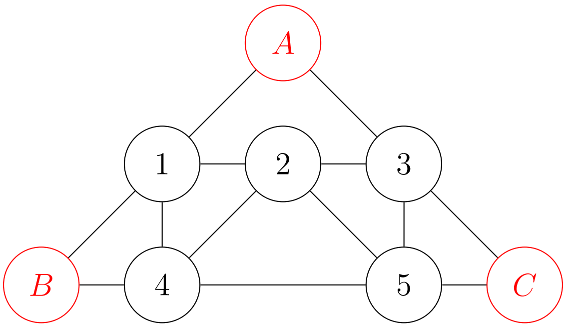
Comme A a la même couleur que D (et donc que j), et que B à la même couleur de D (et donc que l), on s'assure qu'une coloration valide pour G est valide pour H. Ceci est vrai aussi car il existe toujours une coloration du gadget pour une couleur fixée pour A et B.

Supposons à présent qu'il existe une algorithme polynomial permettant de résoudre 3-COLOR-PLAN, la transformation présentée ci dessus se faisant en $O(m^2)$, il existerait alors un algortihme polynomial pour 3-COLOR. Or ce dernier est NP-difficile donc 3-COLOR-PLAN est aussi NP-difficile.

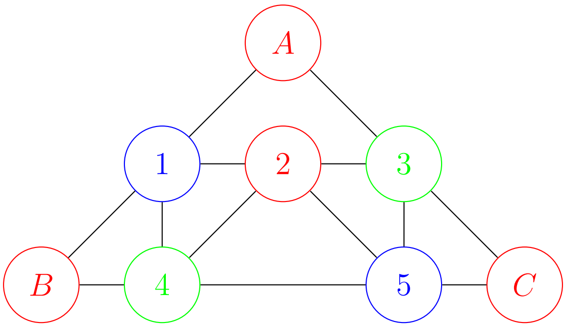
3.2.6 Restriction du problème 3-COLOR-PLAN

Exemple de coloration

On fixe la couleur des sommets A, B et C à rouge.

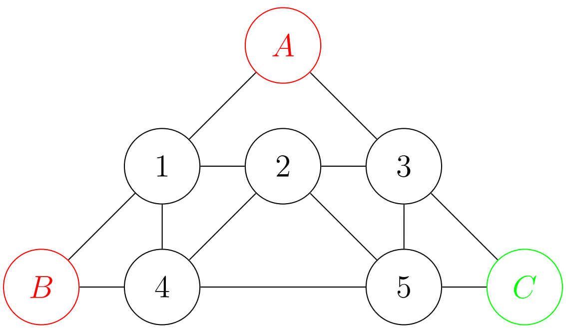


Le graphe est alors 3-coloriable :



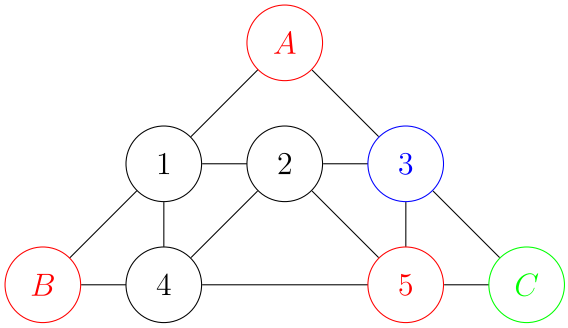
Couleur de A, B et C

Supposons, sans perte de généralité [3], que C est de couleur différente :

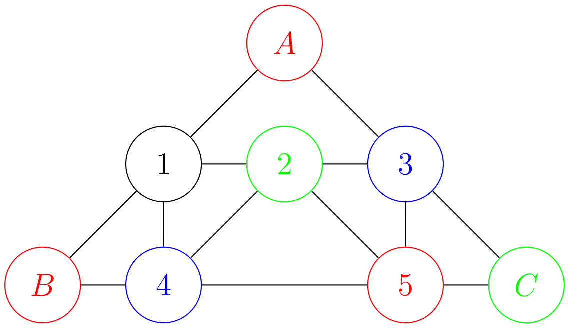


La couleur des sommets 3 et 5 est alors imposée.

3. Les autres configuration peuvent être obetnue par permutation des sommets et des couleurs.



Ce qui impose les couleurs de 2 et 4 et on ne peut pas colorer 1.



Degré des sommets dans le gadget étendu

Dans le graphe ci dessus, tous les sommets sont de degré 4, sauf A, B et C. Le gadget étendu est construit en répétant le graphe ci-dessus n fois et de telle sorte que $\forall k \in 1, \ldots, n-1$, $C_{k-1} = B_k$. Le degré des sommets B_k, $k \in 1, \ldots, n$ est donc de 4. Donc tous les sommets sont de degré 4 sauf B_0, C_{n-1}, A_k, $k \in 0 \ldots n-1$ qui sont de degré 2.

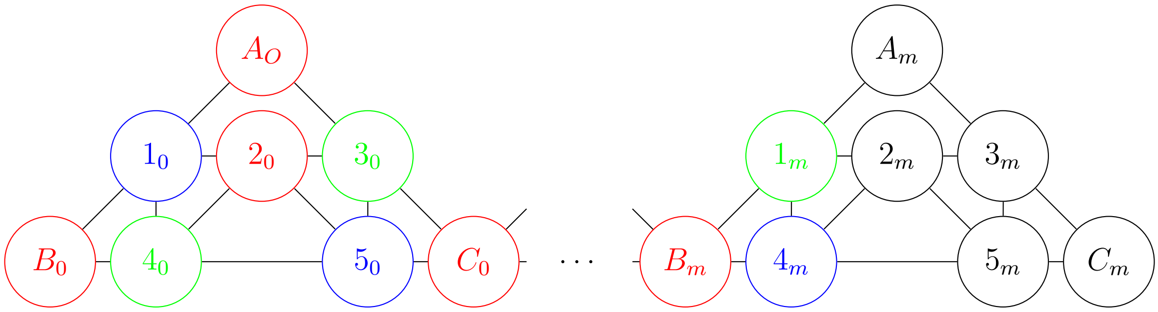
Graphe planaire

La brique élémentaire est planaire, et la répétition du motif n'entraînant pas d'intersection, le graphe final est planaire.

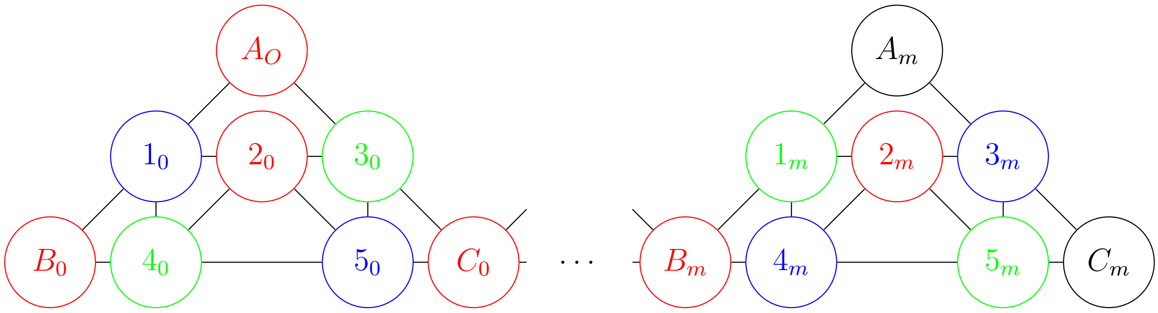
Couleur de A_n, B_0 et C_{n-1}

L'hypothèse est vérifié pour $n = 1$, on la suppose vraie pour $n = m$, qu'en est il de la $m + 1^e$ opération ?

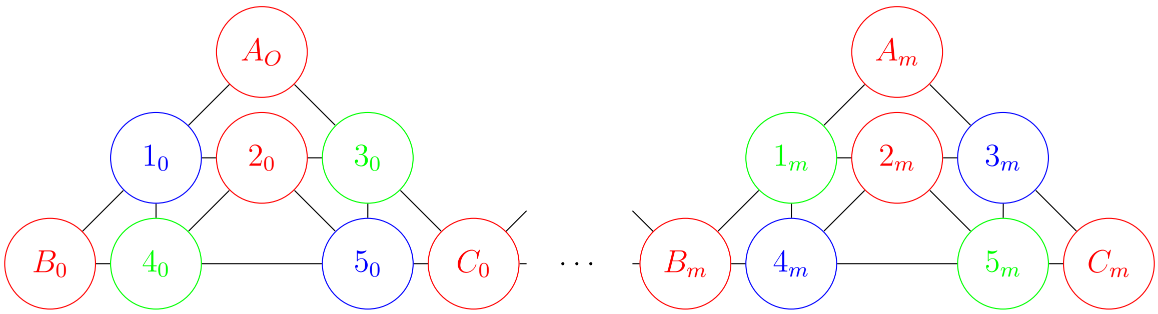
Puisque C_{m-1} est rouge, l'hypothèse est vraie au rang m, alors B_m est rouge aussi puisque les deux sommets sont confondu. Ce dernier appartenant à une clique de taille 3, il n'existe qu'une possibilité pour colorer ses sommets voisins (à permutation des couleurs près).



Le sommet 2_m est obligatoirement de couleur rouge et donc 5_m et de couleur, ce qui impose la coloration bleue de 3_m.



On a alors A_m et C_m qui sont de couleur rouge, la propriété est donc démontrée.



De manière moins visuelle et plus concise, on a démontré que pour colorer une brique élémentaire du schéma, il faut que les sommets A, B et C aient la même couleur. Dans la dernière brique du schéma, la couleur de B est fixée (expliqué précédemment), ce qui fixe la couleur de A et de C.

Coloration possible du gadget étendu

Procédons par récurrence : nous avons démontré que l'hypothèse est vraie au rang 0. On la suppose vraie au rang m. On a alors une coloration du gadget étendu telle que A_k, B_0, C_{m-1}, $k \in 0, \ldots, m-1$ ont la même couleur.

Au rang $m+1$, la couleur de A_m est C_m est fixée et est la même que B_0 et tous les A_k, or B_m est confondu avec C_{m-1} donc B_m est de la même couleur que les sommets précédents. Or si tous les sommets ont la même couleur, il existe une coloration possible de la dernière brique élémentaire et donc une coloration du gadget étendu au rang m. L'hypothèse est donc vérifiée.

3.2.7 NP-complétude de la 3-coloration d'un graphe dont les sommets sont de degré inférieur ou égal à 4

Soit G un graphe planaire quelconque. Pour tous les sommets i de G de dégré strictement $d > 4$, on réalise la transformation suivante pour obtenir H : on sélectionne $z = d - 3$ arêtes sortant de i et on les supprime, i est alors confondu avec le sommet B_0 d'un gadget étendu au rang $z - 1$. les sommets adjacents aux arêtes sélectionnées sont reliés aux sommets C_{z-2} etA_k, $k = 1, \dots, z - 2$.

On transforme alors G en graphe planaire dont tous les sommets ont un degré inférieur ou égal à 4. De plus une 3-coloration pour G est une 3-coloration pour H puisqu'il est possible de colorer le gadget en fixant la couleur des sommets A_k et C_{z-2} à celle de B_0.

Inversement une 3-coloration de H est une 3-coloration de G puisque les sommets du gadget étendu auxquels sont reliés les sommets adjecents à i dont de la même couleur que i, donc lors de la suppression des gadgets étendus, les sommets adjacents à i seront d'une couleur différente de celle de i.

Supposons à présent qu'il existe un algorithme polynomial résolvant 3-COLOR-PLAN(G) tel que les sommets de G ont un degré inférieur ou égal à 4. La transformation d'un graphe planaire H dont les sommets sont de degrés quelconque est en temps polynomial, donc il existerait un algorithme résolvant 3-COLOR-PLAN. Or ce dernier est NP-complet, donc l'existence de cet algorithme n'est vraie que si $P = NP$.

Graphe planaire dont les sommets sont de degré inférieur ou égal à 2

Tous ces graphes sont 3-coloriable.

Chapitre 4

Heuristiques

4.1 Exercice 14 - Coloration de Sommets

Algorithm 1 Approximation de χ(G) séquentielle

Require: G : graphe de n sommets
Ensure: $\chi(G)$: nombre de couleurs nécessaires pour colorer G

Soit $x_1, ..., x_n$ une numérotation des sommets de G.
Soit $C = \{1, 2, ..., k\}$ un ensemble de couleurs.
for all x_i **do**
    $\chi(G) = min\{k \in C : \forall y \in voisinage(x_i), \chi(y) \neq k\}$
end for
return $\chi(G)$

4.1.1 Question 1

Soit la numérotation suivante (parcours en profondeur à partir du premier sommet) :

En appliquant l'algorithme ci-dessus on obtient la coloration suivante :

On remarque que sur cet exemple, la solution est optimale : $\chi(G) = 3$.

4.1.2 Question 2

Malheureusement la solution renvoyée n'est pas toujours tr'es bonne, en effet il est possible de construire un graphe 2-colorable pour lequel l'algorithme renverra $\frac{n}{2}$.

En effet, soit le graphe biparti $G = (V, E)$ suivant :

- $V = \{1, 2, ..., n\}$;
- $E = \{(i, j) \forall i, j \in V : i\ impair,\ j\ pair,\ et\ j \neq i + 1\}$

En exécutant l'algorithme sur un tel graphe, on voit rapidement que le nombre de couleurs utilisées sera bien égal au nombre de sommets divisé par deux. Sur l'exemple de la figure, on voit que pour $n = 6$, la solution $\chi_{alg}(G_{fig3}) = 3$ tandis que $\chi_{opt}(G_{fig3}) = 2$.

4.2 Exercice 15 - Coloration de Sommets

Algorithm 2 Approximation de χ(G) via les degrés des sommets

Require: G : graphe de n sommets
Ensure: $\chi(G)$: nombre de couleurs nécessaires pour colorer G

$C = \emptyset$
$k = 0$
$d_1 >= d_2 >= ... >= d_n$ les degrés des sommets
repeat
    $k = k + 1$
    colorier le sommet x ayant le plus haut degré avec k
    marquer les voisins de x
    $C = C \cup \{x\}$
    while $\exists$ un sommet y non marqué et non coloré **do**
        colorier y ayant le plus haut degré avec k
        marquer les voisins de y
    end while
    effacer toutes les marques
until tous les sommets sont colorés
return k

4.2.1 Question 1

Soit la numérotation suivante (en fonction des degrés) :

En appliquant l'algorithme ci-dessus on obtient la coloration suivante :

On remarque que sur cet exemple, la solution est également optimale : $\chi(G) = 3$.

4.2.2 Question 2

On doit trouver un graphe 2-colorable pour lequel l'algorithme renvoie n.
Prouvons qu'il n'en existe aucun. Soit $G = (V, E)$ un tel graphe.
La seule manière pour que cet algorithme renvoie n, est qu'à chaque itération, k soit incrémenté, i.e. il ne doit jamais rentrer dans la boucle while.
Soit x le sommet de degré maximum de G.
Si degré(x)$< n - 1$, alors soit y un sommet de G tel que $(x, y) \notin E$. L'algorithme lui attribut alors la même couleur (1) que x (en passant dans le while). Ne restant que n-2 sommets à colorer tout en ayant utilisé qu'une seule couleur, l'algoritme renverra donc au plus n-1 couleurs. Ce qui est absurde par hypothèse.
Donc degré(x) = n-1.
Soit $G_2 = (V_2, E_2)$ le sous graphe de G privé de x.
On note x_2 le sommet de degré maximum de G_2.
Si x_2 n'est pas adjacent à tous les sommets non colorés alors comme précédemment deux sommets possèderont la même couleur. Ce qui empêchera l'algorithme de renvyé n. x_2 est donc adjacent à tous les sommets de G_2, ie degré(x_2) $>= n - 2$.
Or x est adjacent à x_2 (cf plus haut), et donc degré(x_2) = n - 1.
En raisonnant de la même manière pour tous les sommets, nous arrivons donc à la conclusion que G est le graphe complet à n sommets.

C'est à dire que G ne peut être coloré au mieux qu'avec n couleurs. Et donc lorsque l'algorithme renvoie n, le graphe ne peut pas être 2-colorable.

Par contre, il existe des graphes 2-colorables pour lequel l'algorithme renvoie $\frac{n}{2}$, par exemple la couronne à n sommets (en supposant que la numérotation des sommets fait les pires choix possibles) :

Chapitre 5

Approximations

5.1 Exercice 16 - Coloration des Sommets d'un Graphe

5.1.1 Question 1

Nous devons montrer que l'algorithme 1 n'est pas r-approché quel que soit $r \in \mathbb{N}$ (celui-ci est détaillé en 4.1).

On cherche donc que $\forall r \in \mathbb{N}$, $\exists G$ tel que $\chi_{algo}(G) > r\chi_{opt}(G)$.
Soit le graphe biparti G défini dans la question 4.1.2 :
- $V = \{1, 2, ..., n\}$;
- $E = \{(i, j) \forall i, j \in V : i\ impair,\ j\ pair,\ et\ j \neq i + 1\}$

Etant biparti, $\chi_{opt}(G) = 2$, et d'après 4.1.2 $\chi_{algo}(G) = \frac{n}{2}$.
Donc $\chi_{algo}(G) = \frac{n}{4} \times \chi_{opt}(G)$.

Pour chaque $r \in \mathbb{N}$ créons le graphe ci-dessus de taille $4 \times r + 4$.
$\chi_{algo}(G) = \frac{n}{4} \times \chi_{opt}(G) = (r + 1) \times \chi_{opt}(G) > r \times \chi_{opt}(G)$.
Ainsi pour chaque $r \in \mathbb{N}$ il existe un graphe qui pour lequel l'algorithme ne renvoie pas une r-approximation.

5.1.2 Question 2

Nous devons maintenant montrer qu'il existe toujours un ordre des sommets tel que l'algorithme 1 renvoie une solution optimale.

Soit G un graphe quelconque.
Soit $couleur(x_i) \forall x_i \in G$ une coloration optimale de G.
Soit la numérotation $\{y_i\}$ $\forall y_i \in G$ telle que $couleur(y_i) \leq couleur(y_{i+1})$.
Il est évident qu'une telle numérotation existe et qu'en suivant celle-ci l'algorithme renverra toujours la solution optimale.

5.1.3 Question 3

Le problème de savoir si un graphe est 2-colorable est polynomial puisque l'algorithme suivant permet toujours de donner la réponse :
On attribue au premier sommet la couleur 1, puis la couleur 2 à tous ses voisins, à nouveau la couleur 1 à leurs voisins respectifs, et on continue ainsi tant qu'il n'existe pas deux sommets adjacents partageant la même couleur.
Si aucune erreur n'est repérée a la fin de l'algorithme, le graphe est 2-colorable (preuve en **??**).

5.1.4 Question 4

Nous devons montrer que si G est 3-colorable, alors $\forall x \in G$ $voisinage(x)$ est un graphe biparti.

Soit G un graphe 3-colorable.
Soit $x \in G$ un sommet quelconque. On note 1 la couleur de x.
Soit G_x le graphe du voisinage de x.
Puisque tous les sommets $y \in G_x$ sont adjacents à x par construction, ils ne peuvent pas être de sa couleur. C'est à dire que $couleur(y) > 1$.
Or G est 3-colorable, donc $couleur(y) \leq 3$.
On a donc que $couleur(y) \in \{2, 3\}$. Ie, G_x est 2-colorable, et tout graphe 2-colorable admet une bipartition de ses sommets (cf 2.4).
G_x est donc un graphe biparti.

5.1.5 Question 5

Il est question de trouver un algorithme $\sqrt{n}$-approché permettant de colorer un graphe 3-colorable. Soit l'algorithme suivant :

Algorithm 3 Coloration de graphe 3-colorable

Require: G : graphe de n sommets 3-colorable
Ensure: $couleur(x)$: couleur attribué au sommet x $\forall x \in G$
trier les $x_i \in G$ par ordre décroissant de degré
$k \leftarrow 1$
$couleur \leftarrow \emptyset$
repeat
    $x \leftarrow sommet(G)$
    $G' \leftarrow voisinage(x)$
    $couleur \leftarrow couleur \cup 2coloration(G', k)$
    $k \leftarrow k + 2$
    $G \leftarrow G \setminus G'$
until $deg(sommet(G)) < \sqrt{n}$
$couleur \leftarrow couleur \cup DMaxPlus1coloration(G, k)$
return $couleur$

Il faut maintenant prouver que cet algorithme renvoie bien une $\sqrt{n}$-approximation.

Tout d'abord l'opération $2coloration(G, k)$ (qui colore un graphe 2-colorable avec les couleurs k et k+1) est expliquée en question 3 (5.1.3).
Quant à $DMaxPlus1coloration(G, k)$ qui colore un graphe avec les couleurs $[k, k + deg_{max}(G)]$, il suffit de voir qu'il est simple de colorer G si on s'autorise 1 couleur de plus que le degré maximum de G. En effet puisque chaque sommet a au plus D voisins, il reste toujours au moins une couleur non utilisée.
De plus, ces deux opérations sont évidemment polynomiales.

Il faut donc compter le nombre de couleurs utilisées.
A chaque fois qu'on entre dans la boucle on élimine $\sqrt{n}$ sommets (puisque le sommet x a un degré d'au moins $\sqrt{n}$).
On entrera donc dans cette boucle au plus $\frac{n}{\sqrt{n}} = \sqrt{n}$ fois. En effet, une fois ce nombre de passages effectués le degré maximum de G est forcément inférieur à $\sqrt{n}$.
Or à chaque itération on utilise seulement deux couleurs, en effet G' est 2-colorable (5.1.4).
Ce qui amène le nombre de couleurs à $2 \times \sqrt{n}$ à la fin de la boucle.
Une fois celle-ci terminée il ne nous reste plus que la dernière coloration qui utilise donc une couleur de plus que le degré maximum de G.
Or $deg_{max}(g) < \sqrt{n}$, ce qui au pire ajoutera encore $\sqrt{n}$ couleurs.
En faisant la somme nous arrivons donc à $3 \times \sqrt{n}$ couleurs.

Le graphe étant 3-colorable, le nombre de couleurs optimal est 3, donc en calculant le ratio de cet algorithme par rapport à la meilleure solution nous obtenons bien : $\frac{3\times\sqrt{n}}{3} = \sqrt{n}$.

Cet algorithme est bien $\sqrt{n}$-approché.

5.1.6 Question 6

Si un graphe dont le voisinage de tout sommet est un graphe biparti impliquait que ce graphe est 3-colorable, alors il existerait un algorithme polynomial permettant de savoir si un graphe est 3-colorable (problème NP) :

Algorithm 4 Graphe 3-colorable ?

Require: G : graphe de n sommets
Ensure: vrai ssi G est 3-colorable, faux sinon

```
for all sommet x ∈ G do
    G_x ← voisinage(x)
    if G_x n'est pas 2-colorable then
        return faux
    end if
end for
return vrai
```

Et donc en supposant que $P \neq NP$, ceci est impossible.

5.1.7 Question 7

Chapitre 6

Contraintes

6.1 Exercice 19 - Mélangeons les reines et les sudokus

6.1.1 Modèles

Pour le modèle orienté case de l'échiquier à remplir, on pose l'ensemble des $X_{ij} \in \{0, 1\}, \forall i, j \in \{1, \ldots, n\}$. Ces variables booléennes X_{ij} représentent la présence ou non d'une reine sur une case ij. Les contraintes sont les suivantes :

$$\textit{Lignes} : \forall i \in \{1, \ldots, n\}, \sum_{j=1}^{n} X_{ij} = 1$$

$$\textit{Colonnes} : \forall j \in \{1, \ldots, n\}, \sum_{i=1}^{n} X_{ij} = 1$$

$$\textit{Diagonales} : \forall i, j, k, l : X_{ij} = 1 \wedge X_{kl} = 1, |i - k| \neq |j - l|$$

Pour le modèle orienté position de la reine sur une colonne, on pose l'ensemble des $X_i \in \{1, \ldots, n\}, \forall i \in \{1, \ldots, n\}$. Ces variables X_i représentent la position d'une reine sur la colonne i. Les contraintes sont les suivantes :

$$\textit{Lignes} : \forall i, j : 1 \leq i < j \leq n, X_i \neq X_j$$

$$\textit{Diagonales} : \forall i, j : 1 \leq i < j \leq n, |X_i - X_j| \neq |j - i|$$

6.1.2 Dual du modèle position sur la colonne

Le dual de ce modèle est celui de la position des reines sur les lignes. On pose l'ensemble des $Y_i \in \{1, \ldots, n\}, \forall i \in \{1, \ldots, n\}$. Ces variables Y_i représentent la position d'une reine sur la ligne i. Les contraintes sont les suivantes :

$$\textit{Colonnes} : \forall i, j : 1 \leq i < j \leq n, Y_i \neq Y_j$$

$$\textit{Diagonales} : \forall i, j : 1 \leq i < j \leq n, |Y_i - Y_j| \neq |j - i|$$

On remarque que les deux modèles sont strictement équivalents grâce à la relation suivante : $X_i = j \iff Y_j = i$.

6.1.3 Stratégie de recherche

Comme ce problème génère un grand espace de recherche, il est nécessaire de chercher à en explorer le moins possible (dans le cas de recherche d'une seule solution). Afin de réduire le facteur de branchement, on va chercher à affecter en priorité les variables dont le domaine est le plus réduit par les contraintes posées précédemment.

D'autre part, on va également chercher à affecter d'abord les variables qui génèrent le plus de contraintes car ce sont celles qui réduiront le plus les possibilités restantes.

6.1.4 Modèle pour la coloration des reines

Les modèles proposés précédemment permettent de résoudre ce problème avec une seule couleur, il faut donc ajouter une dimension à notre modèle : la dimension des couleurs. Reprenons donc le modèle orienté case de l'échiquier à remplir avec des variables entières cette fois et les contraintes suivantes :

$$Lignes : \forall i, j, k \in \{1, \ldots, n\} : j \neq k, X_{ij} \neq X_{ik}$$

$$Colonnes : \forall i, j, k \in \{1, \ldots, n\} : i \neq j, X_{ik} \neq X_{jk}$$

$$Diagonales : \forall i, j, k, l : |i - k| = |j - l|, X_{ij} \neq X_{kl}$$

Notons que chaque X_{ij} est affectée à la plus petite valeur possible de façon à minimiser le nombre de couleurs utilisées.

6.1.5 Solutions pour $n = 5$

Voici la solution donnée par l'énoncé :

0	1	2	3	4
3	4	0	1	2
1	2	3	4	0
4	0	1	2	3
2	3	4	0	1

Il existe d'autres solutions comme celle-ci par exemple :

0	2	4	1	3
1	3	0	2	4
2	4	1	3	0
3	0	2	4	1
4	1	3	0	2

6.1.6 Solution pour $n > 5$

Pour $n = 6$, il n'y a pas de solution. Pour $n = 7$ en voici une :

0	2	4	6	1	3	5
1	3	5	0	2	4	6
2	4	6	1	3	5	0
3	5	0	2	4	6	1
4	6	1	3	5	0	2
5	0	2	4	6	1	3
6	1	3	5	0	2	4

6.1.7 Conclusion

Nous remarquons que ces solutions répètent les mêmes motifs d'une ligne sur l'autre et que par conséquent, elles doivent être étendables à des grilles plus grandes.

Chapitre 7

Combinatoire des mots

7.1 Exercice 20 - Coloriage non répétitif

7.1.1 $\pi(P_n)$

Soit $\pi(P_n)$ le nombre de Thue du graphe P_n c-à-d le plus petit nombre de couleurs nécessaires pour colorier la chaîne à n sommets de manière non-répétitive.

Voici les graphes P_2, P_3 et P_4 ainsi que leur nombre de Thue.

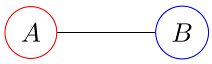


FIGURE 7.1 – $\pi(P_2) = 2$

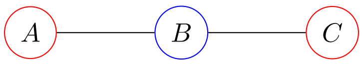


FIGURE 7.2 – $\pi(P_3) = 2$

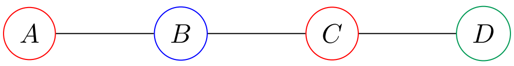


FIGURE 7.3 – $\pi(P_4) = 3$

Soit $H(n)$ l'hypothèse suivante : $\pi(P_n) \geq 3$. Montrons par récurrence que $H(n)$ est vraie pour tout $n \geq 4$.

D'après la figure du graphe P_4 ci-dessus, on a $H(4)$ vraie. On peut donc initialiser la récurrence à $n = 4$.

Supposons que $H(n)$ est vraie et penchons-nous sur le graphe P_{n+1}. Le graphe P_{n+1} étant la chaîne à $n+1$ sommets, on peut colorier une sous-chaîne de P_n à n sommets avec 3 couleurs ou plus de manière non-répétitive d'après l'hypothèse de récurrence. En coloriant le sommet restant avec une nouvelle couleur, on obtient donc un coloriage non-répétitif à 4 couleurs ou plus donc on a bien $H(n+1)$ vraie.

On vient de prouver que $H(n) \rightarrow H(n+1)$ donc on a bien $\pi(P_n) \geq 3, \forall n \geq 4$.

7.1.2 Mot ω sans chevauchement

Montrons que pour tout mot fini ω sur $\{a, b\}$, ω est sans chevauchement si et seulement si $\mu(\omega)$ est sans chevauchement.

Tout d'abord, supposons (sans perte de généralité) que $\omega = mno$ avec m, n, o des mots sur $\{a, b\}$ qui peuvent être le mot vide. Supposons que $n = \alpha u \alpha u \alpha$ avec α une lettre et u un mot. On a alors : $\mu(\omega) = \mu(m)\mu(n)\mu(o)$ par définition. Or $\mu(n) = \mu(\alpha)\mu(u)\mu(\alpha)\mu(u)\mu(\alpha)$. Si l'on décompose le mot avec $\mu(\alpha)_i$ la i^{me} lettre du mot $\mu(\alpha)$, on obtient : $\mu(n) = \mu(\alpha)_1 [\mu(\alpha)_2\mu(u)] \mu(\alpha)_1 [\mu(\alpha)_2\mu(u)] \mu(\alpha)_1\mu(\alpha)_2$. $\mu(n)$ privé de sa dernière lettre est donc bien un facteur de la forme $\alpha u \alpha u \alpha$ et on a :

$$\omega \textit{ est avec chevauchement} \rightarrow \mu(\omega) \textit{ est avec chevauchement}$$

et donc :

$$\mu(\omega) \textit{ est sans chevauchement} \rightarrow \omega \textit{ est sans chevauchement}$$

7.1.3 $\mu^n(a)$ sans chevauchement

Montrons par récurrence que les mots $\mu^n(a)$ sont sans chevauchement pour tout entier $n \geq 0$.

On a $\mu^0(a) = a$ qui est sans chevauchement. Donc au rang 0 la propriété est bien vérifiée.

Supposons $\mu^n(a)$ est sans chevauchement.

On a $\mu^{n+1}(a) = \mu(\mu^n(a))$. Or d'après l'hypothèse de récurrence, $\mu^n(a)$ est sans chevauchement et d'après la question précédente, $\mu(\omega)$ est sans chevauchement ssi ω est sans chevauchement. On a donc bien $\mu^{n+1}(a)$ est sans chevauchement si $\mu^n(a)$ est sans chevauchement. cqfd

7.1.4 Mot infini sans chevauchement

D'après la question précédente, les mots $\mu^n(a)$ sont sans chevauchement pour tout entier $n \geq 0$. Donc $\lim\limits_{n \to \infty} \mu^n(a)$ est sans chevauchement. Or ce mot est un mot infini. Il existe donc bien un mot infini sans chevauchement.

7.1.5 Chevauchements et carrés

7.1.6 Réciproque

7.1.7 Mot infini sans carré

Chapitre 8

Représentation des Connaissances

8.1 Exercice 21 - Coloration et Homomorphisme

8.1.1 Question 1

Soit $G = (V_G, E_G)$ un graphe quelconque connexe (s'il n'est pas connexe, il suffit d'appliquer la démonstration à chaque composante).
Montrons que G est k-colorable $\leftrightarrow \exists$ un homomorphisme de G dans une k-clique $K_k = (V_K, E_K)$.

$\rightarrow$ G est k-colorable.
Soit $W = (V_W, E_W)$ la plus grande clique de G.
On sait que $\chi(W) \leq \chi(G)$ (cf 2.6). On note $x_1, ..., x_w$ les sommets de W, $x_{w+1}, ..., x_n$ les sommets de $G \setminus W$. Et $y_1, ..., y_k$ les sommets de K_k.
Puisque $\chi(K_n) = n$ on remarque que $w \leq k$.
Soit h une fonction de V_G dans V_K construite de la manière suivante :

- (i) $h(x_i) = y_i \forall i \in [1, w]$
- (ii) $h(x_i) \in V_K \setminus h(x_j) : \exists (x_i, x_j) \in E_G$

Les arêtes de W sont bien conservées par (i), K_k étant complet par définition. Quant à celles ayant une extrémité $x \in V_G \setminus V_W$, on sait qu'il existe au moins un sommet x' de V_W tel qu'il n'existe pas $(x, x') \in E_G$. En effet si tel était le cas, le sommet x ferait partie de W.
Ainsi il suffit de poser $h(x) = h(x')$ et ses arêtes seront également respectées.
La fonction h étant bien définie pour tout $x \in V_G$ et conservant les arêtes, celle-ci est bien un homomorphisme de G dans K_k.

$\leftarrow$ $\exists$ un homomorphisme h de G dans K_k.
Intuitivement, un homomorphisme est une fonction qui "contracte" les sommets du graphe de départ dans celui d'arrivée. Ainsi en appliquant h de G dans K_k, il suffit de colorer chaque y_i avec la couleur i, puis de "redéplier" G à partir de son image dans K_k, et tous les sommets sont colorés avec seulement k couleurs.
Plus formellement, soit la coloration classique de K_k suivante : $couleur(y_i) = i \; \forall i \in [1, k]$.
Attribuons à chaque $x_i \in V_G$ la couleur de $h(x_i)$.
Puisque h conserve les arêtes, il est trivial de voir que la coloration de G est valide. De plus k couleurs ont été utilisées.
Donc G est k-colorable.

8.1.2 Question 2

Nous savons que le problème k-colorable est NP-complet.
De plus, nous pouvons définir l'algorithme suivant :

Algorithm 5 k-colorable ?

Require: G : graphe de n sommets, $k \in N$
Ensure: vrai ssi G est k-colorable, faux sinon
$K_k \leftarrow k - clique$
if $\exists h$ un homomorphisme de G dans K_k **then**
    return vrai
else
    return faux
end if

Sous l'hypothèse que $P \neq NP$, et que la recherche d'homomorphisme dans un graphe est polynomiale, nous avons donc un algorithme polynomial permettant de décider si un graphe est k-colorable. Ce qui est absurde.
HOM appartient donc à la classe de problème NP-complet.

8.2 Exercice 22 - Homomorphisme et Satisfaction de Contraintes

Ici, nous nous intéressons à la réduction polynomiale du problème de l'existence d'une solution d'un réseau de contraintes (Constraint Satisfaction Problem) vers le problème de recherche d'homomorphisme dans un graphe (HOM).

Soit le CSP suivant : C = (X,D,C,R) avec :
- $X = \{x_i : \forall i \in [1, n]\}$ l'ensemble des variables de C,
- $D = \cup_{i \in [1,n]} D_i$ où chaque D_i est le domaine de la variable $x_i \in X$,
- $C = \{C_j : \forall j \in [1, p]\}$ où chaque C_j est une contrainte définie par un ensemble ordonné de variables $x_{ji} \in X_j \in X \forall i \in [1, q]$,
- $R = \{R_j : \forall j \in [1, q]\}$ où $R_j \in \{\times_{i \in [1,q]} D_{ji}\}$ est la définition de la contrainte C_j.

Résoudre C revient à déterminer si ce réseau est consistant.

Nous allons construire deux graphes bipartis différents $G_X = (V_X, E_X, \omega_X)$ (la "requête") et $G_R = (V_R, E_R, \omega_R)$ (la base de connaissance).

Soit G_X défini comme suit :
- $V_X = \{x_i \in X\} \cup \{C_j \in C\}$,
- $E_X = \{(x_i, C_j) : x_i$ est une variable apparaissant dans $C_j\}$,
- $\omega_X : E_X \rightarrow N$ une fonction définie telle que $\omega_X(x_i, C_j) = p$ avec p la position de x_i dans C_j.

Soit G_R défini comme suit :
- $V_R = \{D_{ji} \in R\} \cup \{C_j \in C\}$,
- $E_R = \{(D_{ji}, C_j)\}$,
- $\omega_R : E_R \rightarrow N$ une fonction définie telle que $\omega_R(D_{ji}, C_j) = q$ avec q la position de D_{ji} dans C_j.

C est consistant $\leftrightarrow \exists h$ un homomorphisme de G_X dans G_R.

Preuve ? Bijection entre h et la fonction solution de C...

Réalisé avec LaTeX