Marie Jousse
James Peake
Clara-Lou Pahisa

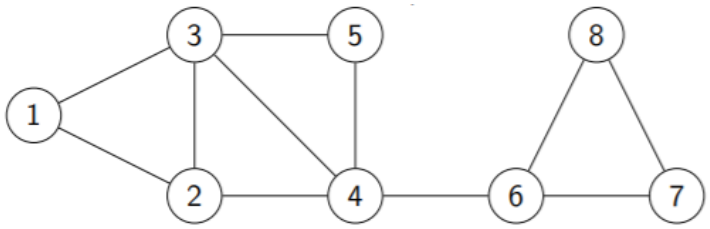
MIN3/MAT2

Rapport de projet INF402 :
Coloration de graphes

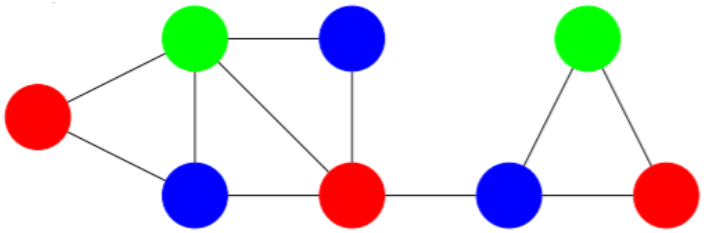
I- Problème

La coloration de graphes est un jeu consistant à attribuer à chaque sommet d'un graphe une couleur, telle que deux sommets reliés n'aient pas la même couleur.
On appelle nombre chromatique le nombre minimum de couleurs nécessaires pour colorier le graphe.

Illustration :
Voici un graphe constitué de 8 sommets. Par exemple, les sommets 1 et 2, ainsi que les sommets 1 et 3 ne doivent pas avoir la même couleur puisqu'ils sont reliés.



Ici, nous avons une résolution possible de cet exemple, avec 3 couleurs différentes.



NB : Lorsqu'un graphe est complet, c'est-à-dire que tous les sommets sont reliés entre eux, alors le nombre chromatique est égal au nombre de sommets du graphe.
(En revanche, si le graphe est planaire, alors le nombre chromatique est inférieur ou égal à 4.)

Il s'agit alors de trouver une manière pour nous de résoudre la coloration d'un graphe à n sommets et m arêtes. Nous allons dans un premier temps exprimer le problème en logique du premier ordre pour mettre en avant la totalité des conditions qu'il faut respecter. Ensuite, nous passerons à l'expression en FNC qui nous sera utile pour la création du programme python.

Marie Jousse
James Peake
Clara-Lou Pahisa

II- Modélisation en logique du premier ordre

Après avoir expliqué le jeu que nous avons choisi en français, on cherche à utiliser la logique du premier ordre.

Avant tout, nous commençons par écrire tout ce qui pourrait nous servir pour la suite, sans forcément appliquer la logique du premier ordre :

BROUILLON :

1_definition des ensembles

Soit S l'ensemble des sommets du graphe
Soit A l'ensemble des arêtes du graphe
Soit C l'ensemble des couleurs du graphe

2_definition de certaines « fonctions »

arete : A → S x S
renvoie les deux sommets formant l'arête

liés : S x A → booléen
pour tout s,s' dans S, liés(s,s') = vrai <=> il existe a dans A tel que arete(a) = (s,s')

couleur : S → C
associe une couleur à un sommet

3_expression des conditions du jeu

existence d'un coloriage si pour tout (s, s') dans S, couleur(s) != couleur(s')
//chaque sommet a une couleur différente

coloriage valide :
valide : S x A x C → B tq pour tout (s,s') dans S, si liés(s,s') = vrai alors couleur(s)!= couleur(s')

EXPRESSION FINALE :

Pour commencer, nous définissons les ensembles qui nous sont utiles :
-soit S l'ensemble des sommets du graphe
-soit C l'ensemble des couleurs du graphe

Ensuite, nous exprimons les prédicats que nous allons utiliser par la suite :

- Le prédicat L(s1, s2) représente le fait que les sommets s1 et s2 sont liés par une arête.
- Le prédicat C(s,c) représente le fait que le sommet s a la couleur c.

Enfin, en utilisant la logique de premier ordre, nous exprimons les conditions essentielles pour répondre au problème de la coloration de graphes :

1. Chaque sommet doit être coloré :

$\forall s \in S, \exists c \in C$ tel que C(s,c)

2. Tous les sommets liés ont une couleur différente:

$\forall s1,s2 \in S, \forall c1,c2 \in C$ tq $c1 \neq c2$, L(s1,s2) $\Rightarrow$ C(s1,c) ET C(s2,c2)

3. Chaque sommet a au plus une couleur:

$\forall s \in S, \exists !c \in C$ tel que C(s,c)

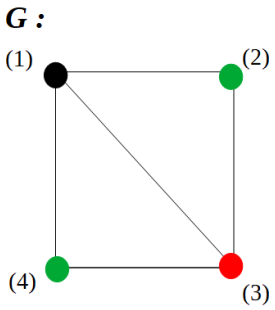
III - Modélisation en forme normale conjonctive

Maintenant que nous avons écrit les conditions liées au problème choisi, nous les exprimons en Forme Normale Conjonctive :

De la même manière que précédemment, on commence par définir les variables et les ensembles que nous allons utiliser.

$x_{i,j}$ est le variable "le sommet i est de couleur j"
$S = \{i, \forall i \in \{1, \dots, n\}\}$ = l'ensemble de sommets
$A = \{(l, k) \in S^2 \mid lié(l, k) = vraie\}$ = l'ensemble d'aretes
C = { j, $\forall j \in \{1, \dots m\} \subseteq \mathbb{N}\}$ = l'ensemble de couleurs
où $m = \chi(G)$ pour G un graphe donné

Pour donner un exemple, on dessine le graphe G ci-joint avec 4 sommets, 5 arêtes et 3 couleurs.
On obtient alors la modélisation suivante :



$S = \{1, 2, 3, 4\}$
A = { (1, 2), (1, 4), (1, 3), (2, 3), (3, 4) }
$C = \{1, 2, 3\}$
où $\chi(G) = 3$ et la couleur 1 est noire, 2 verte et 3 est rouge

Pour un graphe donné, on considère que A, l'ensemble des arêtes, est connu (un graphe étant défini par l'ensemble de ses sommets et les arêtes qui les relient).
En itérant sur l'ensemble de couleurs C, et sur l'ensemble d'arêtes A, on obtient le produit de clauses suivant :

$$\prod_{s=1}^{4}(x_{s,1}+x_{s,2}+x_{s,3}) \times \prod_{s=1}^{4}(\bar{x}_{s,1}+\bar{x}_{s,2}) \cdot (\bar{x}_{s,1}+\bar{x}_{s,3}) \cdot (\bar{x}_{s,2}+\bar{x}_{s,3})$$

$$\times \prod_{c=1}^{3}(\bar{x}_{1,c}+\bar{x}_{2,c}) \cdot (\bar{x}_{1,c}+\bar{x}_{3,c}) \cdot (\bar{x}_{1,c}+\bar{x}_{4,c}) \cdot (\bar{x}_{2,c}+\bar{x}_{3,c}) \cdot (\bar{x}_{3,c}+\bar{x}_{4,c})$$

$$= \prod_{s=1}^{4}(x_{s,1}+x_{s,2}+x_{s,3}) \times \prod_{s=1}^{4} \prod_{(u,v)\in C^2} (\bar{x}_{s,u}+\bar{x}_{s,v}) \times \prod_{c=1}^{3} \prod_{(j,k)\in A} (\bar{x}_{j,c}+\bar{x}_{k,c})$$

$$= \prod_{s\in S}(x_{s,1}+x_{s,2}+x_{s,3}) \times \prod_{\substack{s\in S \\ (u,v)\in C^2}} (\bar{x}_{s,u}+\bar{x}_{s,v}) \times \prod_{\substack{c\in C \\ (j,k)\in A}} (\bar{x}_{j,c}+\bar{x}_{k,c})$$

De manière générale :

$$\prod_{s\in S}\sum_{c\in C} x_{s,c} \cdot \prod_{\substack{s\in S \\ (u,v)\in C^2}} (\bar{x}_{s,u}+\bar{x}_{s,v}) \cdot \prod_{\substack{c\in C \\ (j,k)\in A}} (\bar{x}_{j,c}+\bar{x}_{k,c})$$

$$= \prod_{s\in S}\sum_{c\in C} x_{s,c} \cdot \prod_{\substack{s\in S \\ (u,v)\in C^2}} \sum_{i\in\{u,v\}} \bar{x}_{s,i} \cdot \prod_{\substack{c\in C \\ (j,k)\in A}} \sum_{i\in\{j,k\}} \bar{x}_{j,c}$$

En algèbre de Boole, la clause ($-x_{l,j} + -x_{k,j}$) équivaut au monôme $-(x_{l,j}.x_{k,j})$ par les lois de Morgan. C'est-à-dire que, pour chaque couleur j de C, on a la réunion des sommets qui sont colorés par j, inter la négation des intersections des sommets de chaque arête pour la couleur j.

Chaque gros produit correspond à une des conditions dans la modélisation en logique de premier ordre :

1. à gauche: chaque sommet doit être coloré
2. au milieu: chaque sommet a au plus une couleur
3. à droite: tous les sommets reliés par une arête ne peuvent pas avoir la même couleur.

En français : on suppose que chaque sommet doit être coloré avec une et une seule couleur, mais en interdisant que les sommets l et k soient de la même couleur s'ils constituent une arête.

IV – Implémentation

Nous passons maintenant à la partie code du problème de la coloration de graphe. Nous hésitions entre le langage C et le langage python mais notre choix final a été le python selon les conseils de notre chargé de TD.

Notre programme est divisé en 3 parties majeures :
-*generer.py* : fonctions de générations des graphes, de l'ensemble des arêtes et de l'ensemble des couleurs
-*dessin.py* :

1_SOMMETS : représentés par des cercles, nous avons une fonction qui calcule les coordonnées du sommet sur un plan, une fonction qui les dessine selon les coordonnées (stockées dans un dictionnaire) et enfin une fonction qui dessine les sommets en les colorant

2_ARETES : nous avons écrit une fonction qui dessine les arêtes en reliant deux sommets

3_GRAPHE : grâce aux fonctions précédentes, nous avons une fonction qui dessine un graphe avec ses sommets et ses arêtes ainsi qu'une fonction qui donne la solution, c'est-à-dire, un graphe coloré de manière valide.

-*fnc.py* : fonctions qui utilisent nos clauses et créent un format DIMACS (explication détaillée plus bas).

Enfin, nous avons un fichier principal qui dépend des autres et permet de définir les variables utilisées et avoir l'algorithme de résolution de la coloration d'un graphe.

PRÉCISONS / PROBLÈMES RENCONTRÉS :

Avant de commencer à coder la coloration de graphes, nous ne savions pas si nous devions partir d'un graphe connu, c'est-à-dire avec un nombre de sommets et d'arêtes précis. Nous avons fait le choix, avec la validation de l'enseignant, de partir de graphes connus sans les générer aléatoirement à chaque fois. Cela a entrainé la question des arêtes aléatoires ou non. Nous avions commencé par une fonction de génération de graphe avec un nombre d'arêtes généré aléatoirement. Finalement, nous avons préféré réécrire cette fonction en prenant à la fois un nombre n1 de sommets et un nombre n2 d'arêtes en arguments.

Pour ces valeurs nous avons fait le choix d'un programme interactif. En effet, c'est à l'utilisateur de rentrer le nombre de sommets et d'arêtes. Cependant, notre programme ne prend en compte que des graphes à 10 sommets maximum. Le nombre d'arêtes est limité à n*(n-1)/ 2, avec n le nombre de sommets, puisqu'avec un tel nombre d'arêtes, le graphe est connexe (ou complet).

PASSAGE AU FORMAT DIMACS :

Nous allons expliquer la méthode de transformation en fichier de format DIMACS.

Nos variables sont des couples (un sommet, une couleur) c'est pourquoi il a fallu les transformer en chaines de caractère. Ces dernières sont alors sous la forme de nombre relatifs à deux chiffres, le premier représentant le sommet et le deuxième une couleur. Si ce nombre est positif alors la couleur est associée au sommet sinon cela veut dire que le sommet n'est pas coloré avec cette couleur.

Nous avons écrit une fonction qui donne la FNC dans un fichier DIMACS puis une fonction qui permet d'utiliser le SAT-solveur.

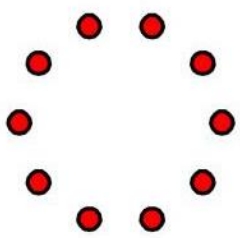
En ce qui concerne les solutions, nous avions commencé en utilisant le SAT-solveur *pycosat*, mais en essayant avec un graphe à 10 sommets, le temps de résolution est très long en raison d'une complexité de notre algorithme trop élevée. De ce fait, nous avons essayé en utilisant un autre SAT-solveur, *minisat*, qui nous propose une réponse valide puisqu'il est plus puissant. L'utilisation de ce dernier se fait de manière manuelle, elle n'est pas incluse dans le programme.

V – Exemples pertinents

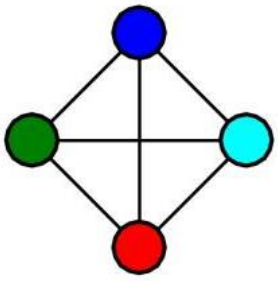
Pour tester notre programme final, nous allons proposer quelques exemples qui nous semblent pertinents :
-un graphe complet à n sommets et n*(n-1)/2 arêtes
-un graphe à n sommets et 1 arête
-un graphe à n sommets et 0 arête
-un graphe à 0 sommet et 0 arête

AFFICHAGE DE CERTAINS RESULTATS :

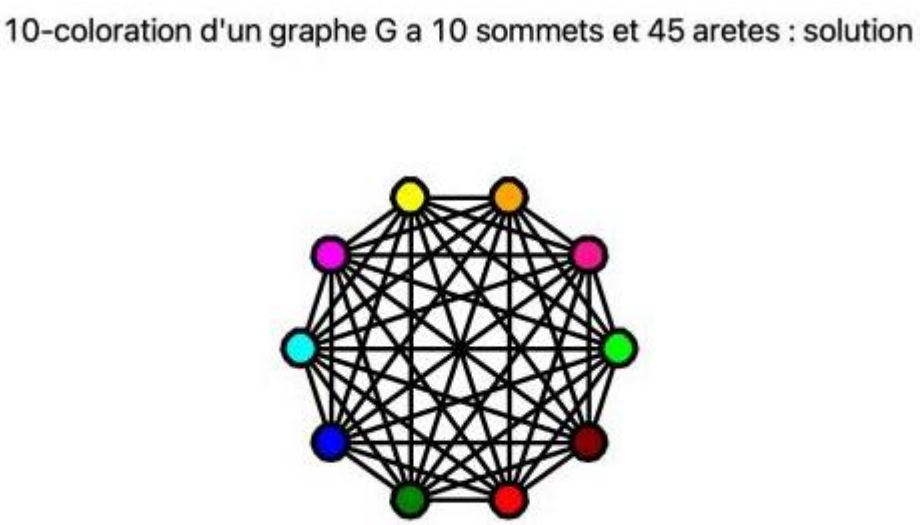
1-coloration d'un graphe G a 10 sommets et 0 aretes : solution



4-coloration d'un graphe G a 4 sommets et 6 aretes : solution



Nous avons testé avec un graphe complet :



NB : Nous avons été obligés de passer par le SAT-solveur minisat pour avoir l'affichage de la solution.

ANALYSE DE NOTRE CODE :

Notre code n'est pas optimal, nous aurions pu utiliser plus de listes chaînées. Un code en langage C aurait pu être mieux optimisé mais le langage python correspondait plus au type de programmes à coder.

Aussi nous utilisons beaucoup de listes pour les différents variables de notre code.