

# **COMPLEXITÉ ET CALCULABILITÉ**

## **MASTER 1 INFORMATIQUE**

### **2016/2017**

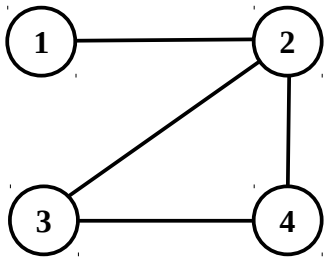
**AL CHAHID Kinda**

**DUBROCA Axel**

**Professeur : Anca Muscholl**

Suite à une perte malencontreuse du rapport d'origine dans les méandres de GitHub, voici une synthèse synthétique des clauses que nous avons implémentées.

Pour générer nos clauses, nous avons considéré chaque arête, et explicité les paires de sommets interdites – celles de même couleur.



Sur notre graphe exemple ci-joint, pour l'arête (1, 2), et 3 couleurs A, B et C, on a donc :

$$(\neg 1_A \vee \neg 2_A) \wedge (\neg 1_B \vee \neg 2_B) \wedge (\neg 1_C \vee \neg 2_C)$$

On a donc juste à continuer pour toutes les arêtes :

$$\begin{aligned} &(\neg 1_A \vee \neg 2_A) \wedge (\neg 1_B \vee \neg 2_B) \wedge (\neg 1_C \vee \neg 2_C) \\ &\wedge (\neg 2_A \vee \neg 3_A) \wedge (\neg 2_B \vee \neg 3_B) \wedge (\neg 2_C \vee \neg 3_C) \\ &\wedge (\neg 2_A \vee \neg 4_A) \wedge (\neg 2_B \vee \neg 4_B) \wedge (\neg 2_C \vee \neg 4_C) \\ &\wedge (\neg 3_A \vee \neg 4_A) \wedge (\neg 3_B \vee \neg 4_B) \wedge (\neg 3_C \vee \neg 4_C) \end{aligned}$$

Et voilà nos clauses. Le souci, c'est qu'il existe un cas où la formule est vraie que l'on doit empêcher : il suffit qu'aucun sommet ne soit coloré. On ajoute donc des clauses permettant de forcer au moins une couleur par sommet :

$$(1_A \vee 1_B \vee 1_C) \wedge (2_A \vee 2_B \vee 2_C) \wedge (3_A \vee 3_B \vee 3_C) \wedge (4_A \vee 4_B \vee 4_C)$$

On obtient donc, pour ce graphe, les clauses suivantes :

$$\begin{aligned} &(1_A \vee 1_B \vee 1_C) \\ &\wedge (2_A \vee 2_B \vee 2_C) \\ &\wedge (3_A \vee 3_B \vee 3_C) \\ &\wedge (4_A \vee 4_B \vee 4_C) \\ &\wedge (\neg 1_A \vee \neg 2_A) \\ &\wedge (\neg 1_B \vee \neg 2_B) \\ &\wedge (\neg 1_C \vee \neg 2_C) \\ &\wedge (\neg 2_A \vee \neg 3_A) \\ &\wedge (\neg 2_B \vee \neg 3_B) \\ &\wedge (\neg 2_C \vee \neg 3_C) \\ &\wedge (\neg 2_A \vee \neg 4_A) \\ &\wedge (\neg 2_B \vee \neg 4_B) \\ &\wedge (\neg 2_C \vee \neg 4_C) \\ &\wedge (\neg 3_A \vee \neg 4_A) \\ &\wedge (\neg 3_B \vee \neg 4_B) \\ &\wedge (\neg 3_C \vee \neg 4_C) \end{aligned}$$

Ici, nous utilisons les indices pour différencier les variables (sommet + coloration). Dans notre implémentation, on a utilisé une bijection réciproque.

Pour passer d'un sommet et de sa coloration à leur valeur associée, on utilise la formule

$$valeur = sommet * nb_{colors} + couleur$$

Et pour l'inverse,

$$\begin{aligned} sommet &= valeur / nb_{colors} && \text{(en utilisant une division entière).} \\ couleur &= valeur \% nb_{colors} \end{aligned}$$

Pour tester le programme, un script – pas très joli – est fourni. Parce que parser la dernière ligne d'un fichier en C était trop fatigant.

Pour vous en servir :

```
./getKcolor.sh [n] [file.c]
```

Avec n le nombre de couleurs à tester, et file.c le fichier C du graphe à utiliser.