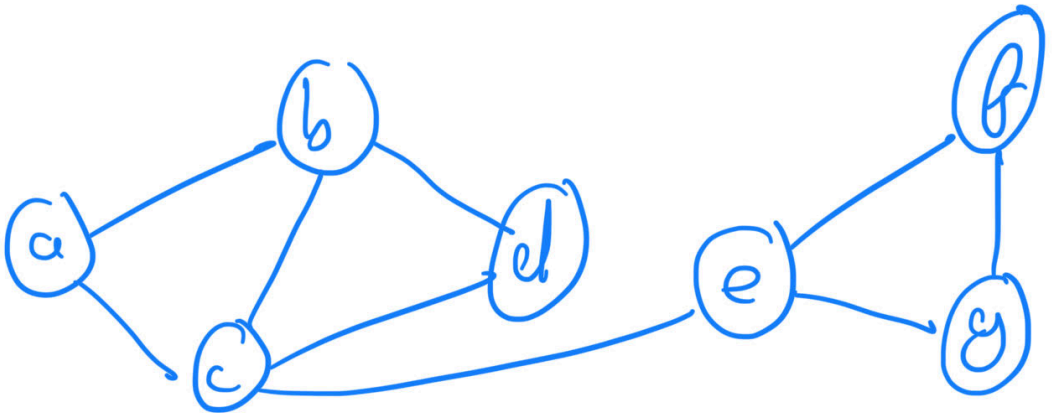
Volodya Aleksanyan
Clément Cournil–Rabeux
LDD 3 IM

Partie 1 - Allocation de registres

1.



2. Minimiser le nombre de registres revient à attribuer un registre à chaque variable, sans avoir de conflit avec les autres variables utilisées au même moment. Cela correspond à effectuer un k-coloriage - où un registre correspond à une couleur- à chaque variable dans le graphe d'intervalles .

Ainsi, la condition qu'un graphe est "bien colorié" signifie que deux sommets connectés n'ont pas la même couleur, ce qui équivaut ici à ce que deux variables ne soient pas affectées au même registre si celles ci sont utilisées à un certain moment en même temps (ce qui correspond au fait d'être connecté dans le graphe d'intervalles).

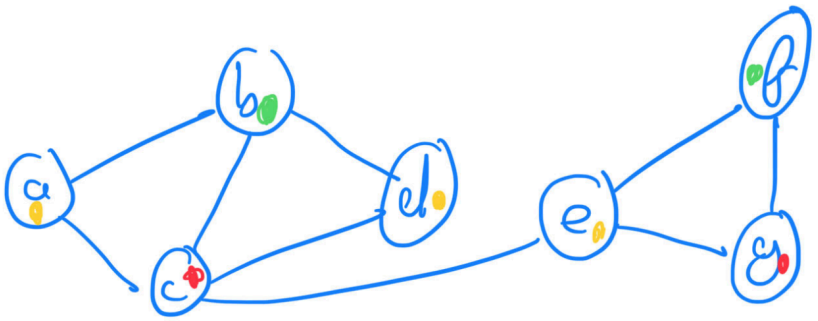
Trouver un k-coloriage valide revient donc à trouver une allocation de k registres valide. Et donc trouver le nombre chromatique du graphe d'intervalles revient à minimiser le nombre de registres à allouer .

3. Soit G = (S,A) un graphe et S' une clique de G et soit n la taille de cette clique Comme tous les sommets de S' sont tous reliés deux à deux, lors d'un k-coloriage de G, ils auront tous obligatoirement une couleur différente, et donc le k-coloriage de G aura au moins n couleurs différentes .

De même, si le k-coloriage est minimal, les sommets de la clique auront tout de même chacun une couleur différente, et donc le coloriage minimal aura au moins n couleurs .

On vient de montrer que pour toute clique S' de G, le k-coloriage minimal de G aura un nombre de couleurs au moins aussi grand que la taille de la clique , et donc cela s'applique aussi à la clique de la taille la plus grande : on a bien que le nombre chromatique de G est plus grand ou égal à la taille de la plus grand clique.

4.



Tri : a, b, c, d, e, f, g

1 2 3 1 1 2 3

(le tri suivant les dates de début d'intervalles est ici dans l'ordre alphabétique)

5. Après l'application de l'algorithme, chaque sommet a été coloré de la couleur "minimale" possible, c'est-à-dire qu'il ne peut être coloré de couleurs plus petites, car il y a un autre sommet avec cette couleur . Donc pour x un sommet de couleur k, il ne peut avoir moins de k-1 voisins, car sinon il existerait alors une couleur l comprise entre 1 et k-1 qui n'est pas présente chez ses voisins, et cela contredirait le fait que x a choisi une couleur "minimale" (l < k).
Donc x a au moins k-1 sommets voisins .

6. Pour t étant le début de l'intervalle correspondant au sommet x, comme x a au moins k-1 sommets voisins, on a que t appartient à au moins k intervalles (les k-1 voisins de x + lui même) .

De plus, si t appartenait à strictement + de k intervalles, cela voudrait dire que x est connecté à plus de k-1 voisins . On aurait donc au moins un voisin qui aurait une couleur plus élevée .
Cependant, ce voisin ayant une couleur plus élevée a forcément été colorié après x dans l'algorithme du 4., qui colorie selon un tri via les dates de début des intervalles . et donc on aurait que ce voisin a une date de début située après celle de x, et donc ne chevauche pas le début de l'intervalle de x, càd t , ce qui aboutit à une contradiction .

Donc on a montré que t appartient à au moins k intervalles et qu'il n'appartient pas à strictement plus de k intervalles : t appartient exactement à k intervalles .
Remarque : Cela n'est vrai que quand tous les débuts d'intervalles sont distincts (on ne peut avoir deux débuts étant exactement les mêmes) .
7. En ne prenant que les sommets xi correspondant aux variables utilisées à l'instant t (du 6.), on a que tous les sommets correspondants sont connectés deux à deux.

On a donc une clique de taille k. On a que la taille de la plus grande clique est plus grande ou égale à celle de celle-ci, càd k, ce qui conclut .