

Gestion distribuée des ressources

TP 1 & 2 : Algorithmes auto-stabilisants & pallier la perte d'une information locale

Exercice 1 : algorithmes auto-stabilisants — cas de la 3-Coloration d'un anneau

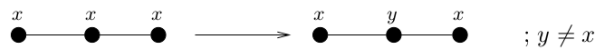
Un algorithme est dit auto-stabilisant si, quel que soit son état initial, il garantit l'atteinte d'un état donné en un nombre d'étapes fini. Ce type d'algorithmes se termine correctement malgré les perturbations qui pourraient survenir dans le système.

Soit un anneau composé d'au moins trois nœuds. L'on souhaite attribuer des couleurs (parmi 3 possibles) de telle sorte que deux nœuds voisins aient des couleurs différentes.

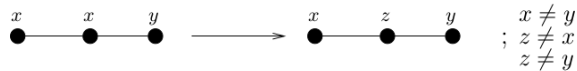
La coloration des sommets est un problème qui s'apparente à l'allocation de ressources et trouve de nombreuses applications en réseaux (allocation des canaux dans les réseaux sans fils) ou gestion des systèmes.

Soient x , y et z les trois couleurs à utiliser. Pour réaliser la 3-coloration, on peut utiliser deux règles :

1. si un nœud a la même couleur que ses deux voisins, alors il choisit une couleur différente ;



2. si un nœud a la même couleur qu'un seul de ses voisins, il choisit une autre couleur en évitant celle de son second voisin.



Travail à faire :

1. Quel est le type de synchronisation le mieux indiqué pour l'écriture de ces règles ?
2. Les sommets doivent-ils, au départ, avoir obligatoirement l'une des trois couleurs choisies ?
3. Réalisez l'implémentation de l'algorithme et vérifiez qu'il se termine correctement.
4. Testez le passage à l'échelle de votre algorithme.

Exercice 2 : pallier la perte d'une information locale

Dans cet exercice, nous supposons que le graphe d'étude est connexe et que chaque nœud a un degré supérieur ou égal à deux. Au départ, un et un seul nœud a une étiquette "A". Tous les nœuds ont une étiquette "N". La figure 1 illustre un exemple d'un tel graphe.

