Algorithmique Distribuée : TD 4, algorithmes silencieux

Alain Cournier Stéphane Devismes

8 juin 2026

1 Ensemble Indépendant Maximal

Soit $G = (V, E)$ un graphe simple et non-orienté où V est l'ensemble des sommets et E l'ensemble des arêtes. Soit $I \subseteq V$.

— I est un *ensemble indépendant* (de G) si et seulement si pour toute paire de sommets distincts $\{p, q\}$ de I on a p et q qui ne sont pas voisins, c'est-à-dire, $\{p, q\} \notin E$.

— I est un *ensemble indépendant maximal* (de G) si et seulement si I est indépendant et tout sur-ensemble (propre) de I ne l'est pas.

Question 1. Nous considérons le graphe donné en figure 1. Pour chacun des sous-ensembles suivants dites s'il est indépendant et indépendant maximal. Justifiez les réponses négatives.

	Indépendant	Indépendant maximal	Justification
$\emptyset$			
$\{5\}$			
$\{5, 6\}$			
$\{5, 6, 8\}$			
$\{1, 2, 5\}$			
$\{3, 6, 7\}$			
$\{2, 5, 6, 8\}$			
$\{1, 2, 8, 9\}$			
$\{1, 3, 7, 9\}$			
$\{2, 5, 8, 9\}$			
$\{1, 2, 4, 8, 9\}$			
$\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$			

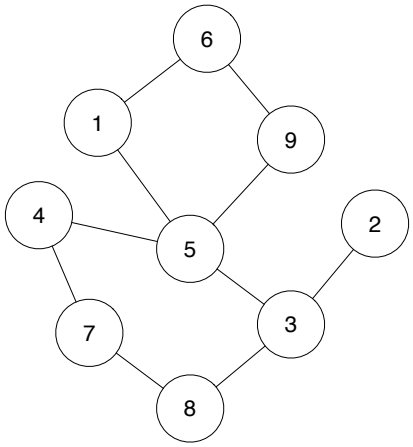

FIGURE 1 – Réseau quelconque.

2 Algorithme Autostabilisant calculant un Ensemble Indépendant Maximal

2.1 Dans un Réseau Anonyme

Nous rappelons que dans un réseau anonyme les processus sont indistinguables sauf lorsque leur degré est différent. Donc, ils ont le même algorithme avec en particulier les mêmes variables.

Question 2. Proposez un exemple de réseau anonyme et de démon pour lesquels il n'existe aucun algorithme déterministe autostabilisant (dans le modèle à états) calculant un ensemble indépendant maximal. Justifiez.

Question 3. Généralisez la réponse précédente.

2.2 Dans un Réseau Identifié

Nous considérons maintenant des réseaux bidirectionnels (l'hypothèse connexe n'est pas nécessaire) $G = (V, E)$ avec identité. $\forall p \in V$, id_p désigne l'identité de p.

2.2.1 Les Algorithmes Silencieux

Dans le modèle à états, un algorithme autostabilisant est *silencieux* si toutes ses exécutions atteignent une configuration dites *terminale* où plus aucun processus n'est activable. Généralement, les algorithmes silencieux sont utilisés pour calculer des structures sur le réseaux, comme par exemple un arbre couvrant.

2.2.2 Algorithme Silencieux calculant un Ensemble Indépendant Maximal

Nous considérons le modèle à états avec un démon distribué inéquitable. L'algorithme consiste à calculer une variable de sortie Booléenne $S_p \in \{Dominant, dominé\}$ de telle manière que l'ensemble $I = \{p \in V, S_p = Dominant\}$ est un ensemble indépendant maximal.

L'idée générale de l'algorithme est la suivante : Un processus doit être *Dominant* si et seulement si chacun de ses voisins est soit *dominé*, soit d'identité plus grande.

L'algorithme comporte deux règles *Leave* et *Join*. Un processus exécute *Leave* pour quitter l'ensemble indépendant. Un processus exécute *Join* pour entrer dans l'ensemble indépendant. L'ensemble des voisins d'un processus p donné sera noté $\mathcal{N}_p$.

Question 4. Écrivez les deux règles de l'algorithme.

Question 5. Les actions *Leave* et *Join* d'un processus donné sont elles mutuellement exclusives ? Justifiez.

2.2.3 Preuve de l'algorithme

La preuve de correction d'un algorithme silencieux en supposant un démon distribué inéquitable comporte deux étapes majeures :

1. D'abord, montrer que dans toute configuration terminale, les variables définissent une solution du problème considéré (ici les variables doivent définir un ensemble indépendant maximal).
2. Puis, montrer que toute exécution termine, c'est-à-dire, atteint une configuration terminale, en un nombre fini d'étapes de calcul.

Question 6. Montrez, par contradiction, que dans toute configuration terminale, I est un ensemble indépendant.

Question 7. Montrez, par contradiction, que dans toute configuration terminale, I est un ensemble indépendant maximal.

D'où

Théorème 1. *Dans toute configuration terminale, I est un ensemble indépendant maximal.*

Nous allons maintenant démontrer que toute exécution de notre algorithme termine. Pour cela, nous allons utiliser la notion de $RANG$.

Soit $RANG(p) = |\{q \in V, id_q \leq id_p\}|$.

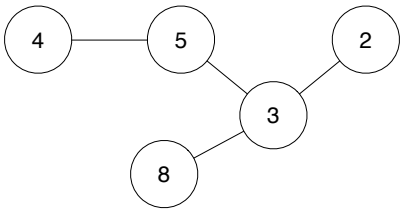

FIGURE 2 – Réseau identifié quelconque. (les identités sont les numéros à l'intérieur des nœuds.

Question 8. Dans la figure 2, qui est le processus de rang 1 et quel est le rang du processus d'identité 5 ? Expliquer informellement la notion de rang. Pourquoi est-ce un bonne mesure pour faire une récurrence.

Question 9. Dans le pire des cas, combien d'actions exécute le processus de rang 1 ?

Question 10. Prouvez par récurrence sur le rang, que tout processus ne peut changer d'état qu'un nombre fini de fois.

D'après la question précédente, on a :

Théorème 2. *À partir d'une configuration quelconque, le système atteint une configuration terminale en un nombre fini d'étapes de calcul.*

Question 11. Proposez une exécution possible comportant $\Omega(n^2)$ étapes de calcul.

Question 12. Quel est le temps de stabilisation en rondes de l'algorithme ? Donnez un exemple d'exécution réalisant ce temps. Prouvez par récurrence sur le rang cette borne.