

Algorithmique Distribuée : TD 3, Autostabilisation

Stéphane Devismes

27 mars 2023

1 Le modèle à états

Nous considérons un anneau orienté de $n > 0$ processus : p_0, \dots, p_{n-1} . Chaque processus p_i peut lire l'état de son prédécesseur p_{i-1} , *i.e.*, les valeurs des variables de son prédécesseur (les calculs sur les indices sont modulo n bien entendu!).

Tous les processus ont le même programme, sauf p_0 . Ce dernier est appelé la *racine*. Par suite, les autres sont appelés processus *nonracines*.

Chaque processus i a une unique variable v dont le domaine est $\{0, \dots, K-1\}$ avec $K \geq n$.

Le programme de chaque processus consiste en une unique règle donnée ci-dessous. Une règle se décompose en deux parties : *garde* \mapsto *action*. Dans un anneau unidirectionnel, la *garde* est un prédicat sur les variables du processus et de son prédécesseur. L'*action* consiste ici en une seule affectation sur la variable du processus. L'*action* de la règle ne peut être exécutée que si la *garde* de l'action est vraie.

Programme de la racine p_0 :

$$Token(p_0) \mapsto v_{p_0} \leftarrow (v_{p_0} + 1) \bmod K, \text{ où } Token(p_0) \equiv (v_{p_0} = v_{p_{n-1}})$$

Programme de chaque processus p_i nonracine :

$$Token(p_i) \mapsto v_{p_i} \leftarrow v_{p_{i-1}}, \text{ où } Token(p_i) \equiv (v_{p_i} \neq v_{p_{i-1}})$$

Exécution d'une étape de calcul : Une règle est dite *activable* lorsque sa garde est vraie. Un processus est dit *activable* lorsque sa règle est activable. Dans le cas contraire, le processus est dite *inactivable*.

Une étape de calcul s'exécute toujours de la façon suivante :

1. On évalue qui sont les processus activables.
2. Si aucun processus n'est activable, on arrête.
3. Sinon, un adversaire, appelé *démon*¹, choisit des processus activables, au moins un.
4. Les processus choisis exécutent simultanément leur règle activable.
5. Et on recommence.

2 Exercices

Nous supposons dans cette section que $K = 8$.

1. Le démon modélise l'asynchronisme du système.

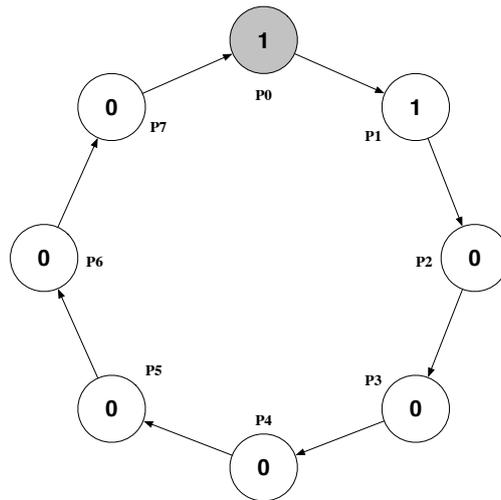


FIGURE 1 – Une configuration possible.

Question 1. Exécutez 8 étapes de calcul à partir de la configuration donnée en figure 1, en supposant que le démon choisit systématiquement tous les processus activables à chaque étape de calcul.

	<i>P0</i>	<i>P1</i>	<i>P2</i>	<i>P3</i>	<i>P4</i>	<i>P5</i>	<i>P6</i>	<i>P7</i>
0	1	1	0	0	0	0	0	0
1								
2								
3								
4								
5								
6								
7								
8								

Question 2. Existe-t-il plusieurs exécutions possibles de l'algorithme à partir de la configuration donnée en figure 1 ?

Question 3. À partir de la configuration donnée en figure 1, combien faut-il d'étapes de calcul pour retrouver à nouveau la configuration donnée en figure 1 ? Est-ce dépendant du choix de K ?

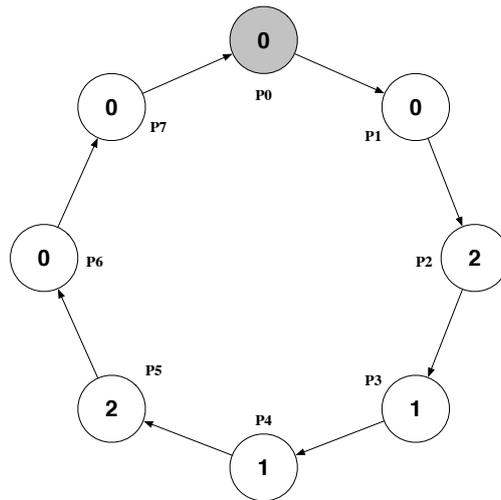


FIGURE 2 – Une configuration possible.

Nous considérons maintenant la configuration donnée en figure 2.

Question 4. Exécutez 15 étapes de calcul à partir de la configuration donnée en figure 2, en supposant que le démon choisit systématiquement tous les processus activables à chaque étape de calcul.

	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
0	0	0	2	1	1	2	0	0
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								

Question 5. Exécutez 8 étapes de calcul à partir de la configuration donnée en figure 2, en supposant que le démon choisit systématiquement le processus activable d'indice minimum à chaque étape de calcul.

	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
0	0	0	2	1	1	2	0	0
1								
2								
3								
4								
5								
6								
7								
8								

Question 6. Exécutez 18 étapes de calcul à partir de la configuration donnée en figure 2, en supposant que le démon choisit systématiquement le processus activable d'indice maximum à chaque étape de calcul.

	<i>P0</i>	<i>P1</i>	<i>P2</i>	<i>P3</i>	<i>P4</i>	<i>P5</i>	<i>P6</i>	<i>P7</i>
0	0	0	2	1	1	2	0	0
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								

Question 7. Quel phénomène remarquez-vous dans les exécutions précédentes? Construisez un pire des cas.

Question 8. Quel est le but de cet algorithme? Décrivez ces propriétés.

