

# Systèmes Distribués : TD 1, circulation d'un jeton dans un réseau quelconque

Alain Cournier

Stéphane Devismes

## 1 Principe de l'algorithme

L'algorithme proposé par Tarry en 1895 résout le problème du labyrinthe. Représentons les intersections dans le labyrinthe par des sommets et les allées entre ces intersections par des arêtes. Un labyrinthe se modélise donc comme un graphe non orienté connexe. Enfin, utilisons un jeton pour symboliser le marcheur qui se déplace. L'algorithme de Tarry devient alors un algorithme de circulation de jeton dans un graphe quelconque. Le principe de l'algorithme de Tarry est le suivant : « Ne reprendre l'allée initiale qui a conduit à un carrefour pour la première fois que lorsqu'on ne peut pas faire autrement ». Ainsi, l'algorithme de Tarry applique le principe du parcours en profondeur d'abord.

## 2 Hypothèses

- Processus et canaux asynchrones.
- Canaux étiquetés de 1 à  $\delta_p$  pour tout processus  $p$ .
- Pas de faute.
- Topologie quelconque (connexe) d'au moins deux nœuds.
- Mono-initiateur.

## 3 L'algorithme

Écrivez le code de l'algorithme. Pour cela, vous utiliserez un seul type de message :  $\langle Jeton \rangle$ ; de plus vous utiliserez les variables locales (à chaque processus) suivantes.

- $Visite[1..\delta_p]$  : tableau de Booléen, toutes les cases sont initialisées à *faux*.
- $Père \in \{\top, \perp\} \cup \{1 \dots \delta_p\}$  : pointeur « parent », initialisé à  $\perp$ . Lors du démarrage « spontané », l'initiateur affecte sa variable  $Père$  à  $\top$ .

## 4 Correction de l'algorithme

**Question 1.** Justifiez pourquoi il existe toujours au plus un jeton dans le réseau.

**Question 2.** Justifiez pourquoi au plus une décision est exécutée.

---

**Algorithme 1** *Circulation* pour tout processus  $p$

---



**Question 3.** Combien de fois est traversée une arête lors de l'exécution ? (Justifiez).

**Question 4.** Pourquoi l'exécution termine<sup>1</sup> toujours ?

**Question 5.** Pourquoi y a-t-il nécessairement une décision lors de l'exécution ?

**Question 6.** Quel processus finit par décider ? Pourquoi ?

**Question 7.** Que peut-on dire des canaux du processus qui a décidé ?

**Question 8.** Une fois la décision exécutée, que peut-on dire des canaux d'un processus visité ? En déduire qu'à l'exécution de la décision tous les processus ont été visité.

---

1. Une exécution termine si elle ne comporte qu'un nombre fini de réception de messages.

**Question 9.** Donnez la complexité en temps et en nombre de messages de l'algorithme.

--

## 5 Circulation perpétuelle

**Question 10.** Réécrivez l'algorithme pour en faire une circulation perpétuelle.

---

**Algorithme 2** *Circulation* pour tout processus  $p$

---

--