

Devoir 1

Ce devoir est à rendre pour le 17 février. Vous pouvez travailler en groupe d'au plus trois à la condition que la participation des membres du groupe soit réelle. Il est inutile de recopier un livre (ou le devoir du voisin).

Exercice 1.— Il s'agit d'étudier un algorithme de vague: *l'algorithme de phases*.

On dispose d'un graphe orienté $G = \langle \Pi, V \rangle$ fortement connexe. On notera $d(p, q)$ la longueur du plus court chemin de p à q (attention, en général $d(p, q) \neq d(q, p)$). D le *diamètre* est défini par $D = \max\{d(p, q) \mid p, q \in \Pi\}$. Pour cet algorithme on suppose que D est connu des processus. Pour chaque processus p , $In_p = \{q \in \Pi \mid (q, p) \in V\}$ est l'ensemble des processus qui peuvent envoyer directement à p un message et Out_p ceux qui peuvent recevoir directement de p . \preceq est la relation de causalité entre événements. Pour simplifier on supposera que la communication est FIFO (les messages entre deux noeuds sont reçus suivant leur ordre d'émission).

Dans cet algorithme tous les processus sont des initiateurs.

Initialisations:

1 $\forall q \in In_p : Rec_p[q] := 0$

2 $Phase_p := 0$

CODE POUR LE PROCESSUS p :

3 **forall** $r \in Out_p$ **do** send(<tok>) to r

4 $Phase_p := Phase_p + 1$

5 **while** $Min_q Rec_p[q] < D$ **do**

6 receive (<tok>) from some $q_0 \in In_p$

7 $Rec_p[q_0] := Rec_p[q_0] + 1$

8 **if** $Min_q Rec_p[q] \geq Phase_p$ and $Phase_p < D$ **then**

9 **forall** $r \in Out_p$ **do** send(<tok>) to r

10 $Phase_p := Phase_p + 1$

11 decide

Figure 1: Algorithme de phases

1. On considère le graphe $\Pi = \{1, 2, 3, 4, 5\}$, $V = \{(1, 2), (2, 1), (2, 3), (3, 2), (3, 4), (4, 3), (4, 5), (5, 4)\}$. Ici le diamètre du graphe est 4.

Donner un exemple d'exécution de l'algorithme de phase (Figure 1). Sur cet exemple indiquer les événements suivants: l'émission du premier message envoyé par le processus 1 au processus 2, l'émission du 2ème message envoyé par le processus 2 au processus 3, l'émission du 3ème message envoyé par le processus 3 au processus 4, l'émission du 4ème message envoyé par le processus 4 au processus 5. Dans votre exemple d'exécution quel est l'ordre de ces événements?

Pouvez vous trouver un exemple d'exécution où ces événements apparaîtraient dans un ordre différent?

2. On veut maintenant montrer que l'algorithme de phases termine et décide. Pour cela, on considère une exécution γ de cet algorithme.
 - (a) Combien de messages peuvent être envoyés sur chaque lien de communication? Quel est le nombre maximal de messages qui peuvent circuler?
 - (b) Soit l'instant t_0 à partir duquel dans γ tous les messages ont été reçus et plus aucun message ne sera envoyé.
 - i. Montrer que dans γ à partir de l'instant t_0 , on a:
 - $Phase_p > 0$ pour tout processus p ,
 - $Rec_p[q] = Phase_q$ pour tout processus p, q tel que $q \in In_p$,
 - ii. En déduire que tous les processus décident.
3. On va montrer maintenant que l'algorithme de phases est un algorithme de vague. E_p est l'ensemble des événements du processus p . On note e_p l'événement correspondant au démarrage de l'algorithme en p . Si a est un événement sur p , $ph_p(a)$ est la valeur de la variable $Phase_p$ au moment de cet événement.
 - (a) Montrer la propriété suivante (on pourra faire une induction sur la distance $d(q, p)$).

$$\text{si } d(q, p) = i \text{ alors } \forall a \in E_p : ph_p(a) > i \Rightarrow e_q \preceq a \quad (1)$$
 - (b) En déduire que l'algorithme est un algorithme de vague.
 - (c) Que se passe-t-il si on ne suppose plus que la communication est FIFO?
 - (d) Quel est le nombre de messages échangés avant la décision de tous les processus?
4. Essayer de modifier l'algorithme de phases de façon à obtenir un algorithme de vague même si le diamètre et le nombre de processus sont inconnus.