

Exercices Master Informatique, IFI, Parcours CSSR

Course 1

2009-2010

F. Baude

Exercice 1

Voici 2 versions de l'algorithme d'élection sur anneau unidirectionnel publié par Chang et Roberts en 1979.

Version 1, vue en cours

On rappelle que le modèle de programmation adopté en cours est le suivant :

On discerne 2 types d'actions : les actions **internes**, et les actions **dirigées par les messages**.

Une action interne I d'un processus p est spécifiée selon le schéma suivant :

I_p : Si la condition { garde } est vérifiée

Alors séquence d'instructions utilisant seulement des variables locales à p
 envoie un message à ...
 séquence d'instructions utilisant seulement des variables locales à p

...

Une action peut donc être exécutée si la garde est vraie. Aucun message ne peut être reçu dans une action interne, mais seulement au début d'une action dirigée par les messages.

Les actions dirigées par les messages sont spécifiées de façon similaire :

M_p : Si la condition { message <Mess> arrivé } est vérifiée

Alors ...

...

Typiquement, un processus est constitué de plusieurs actions. Si plusieurs d'entre elles sont éligibles à un moment donné (car leurs gardes sont vraies), une parmi elles est choisie de façon non-déterministe.

On utilise les variables suivantes :

- M : initialisée à 0
- mon_numero : identificateur du nœud

Texte de l'algorithme en P_i :

I_i : Si la condition $\{ M=0 \}$ est vérifiée
 Alors $M := mon_numero$
 envoie $\langle M \rangle$ au suivant dans l'anneau

M_i : Si la condition $\{ \text{message } \langle j \rangle \text{ arrivé} \}$ est vérifiée
 Alors Si $M < j$ Alors
 $M := j$
 envoie $\langle M \rangle$ au suivant dans l'anneau
 Si $j = i$ alors
 "Je suis le leader"

On suppose que "Je suis le leader" déclenche un algorithme de diffusion sur l'anneau pour en informer les autres, qui remettra toutes les variables M à 0, mais il ne s'agit plus alors de l'algorithme d'élection proprement dit.

Version 2, similaire à celle du polycopié de S. Krakowiak

On utilise les variables suivantes :

- *participant* : booléen vérifiant le réveil du noeud. Il est mis à faux au départ.
- *mon_numero* : identificateur du noeud.

Texte de l'algorithme en P_i :

Initialisation en cas de réveil spontané

```
faire
  participant := vrai
  envoie (mon_numero) au suivant dans l'anneau
fin
```

A la réception de (j) (à ce moment là, P_i est réveillé et ne peut donc plus exécuter le code d'initialisation)

```
faire
  si (j > mon_numero)
    alors participant := vrai;
    envoie (j) au suivant dans l'anneau;
  si (j < mon_numero) et (non participant)
    alors participant := vrai;
    envoie (mon_numero) au suivant dans l'anneau;
  si (j = mon_numero)
    alors
      participant := faux;
      "Je suis le leader";
fin
```

Question

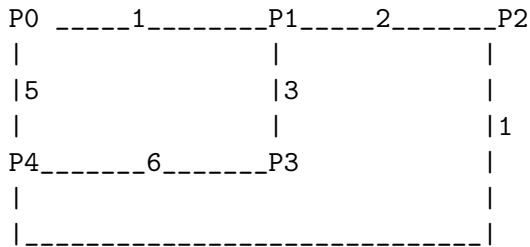
Ce sont bien 2 versions qui résolvent un même problème à savoir élire un leader. Cependant, il y a une différence sur le résultat de l'élection (regardez ce qu'il se produit dans les cas où tous les processus se réveillent spontanément et dans les cas où seulement une portion d'entre eux se réveillent spontanément). Expliquez cette différence.

Exercice 2

On peut calculer de façon distribuée des tables de routage de plus faible coût en s'appuyant sur un **calcul d'approximation**. Les idées d'un tel algorithme sont données dans l'article de Mattern à partir du dernier paragraphe de la page 8.

On vous demande de décrire **une exécution possible** de cet algorithme sur le graphe donné ci-dessous, sans vous soucier du problème de détection de sa terminaison. Il s'agit de dire quels messages sont engendrés, quels traitements ils entraînent, et quelles sont les valeurs successives que prennent les tables de routage sur chaque processus du graphe.

Le graphe est le suivant, les valeurs numériques portées sur les arêtes étant les coûts.



Exercice 3 - Mentionné transparent 21 du cours

A partir de l'algorithme Echo présenté dans l'article de Mattern, écrire un algorithme permettant de résoudre l'élection de leader dans un graphe quelconque.

Ensuite, appliquer votre algorithme sur l'exemple de graphe suivant : 4 sites : a, b, c, f, et les canaux de communication bi-directionnels suivants : ab, ac, af, cf. Supposez que l'ordre lexicographique des identités associées aux sites permet de déterminer le max (à savoir ici, f). Supposez que tous les sites se réveillent spontanément.

Exercice 4, from Ghosh book, chapter 3

A wireless sensor network is being used to monitor the maximum temperature in a region. Each node monitors the temperature of a specific point in the region. Propose an algorithm for computing the maximum temperature and broadcasting the maximum value to every sensor node. Assume that communication is by broadcasting only, and the broadcasts are interleaved (i.e., they do not overlap), so two sensor nodes never send or receive data at the same time.

Exercise 5, from Ghosh book, chapter 3

Consider an anonymous distributed system consisting of N processes. The topology is a completely connected network and the links are bidirectional. Propose an algorithm using which processes can acquire unique identifiers. (Hint: use coin flipping, and organize the computation in rounds).