

Algorithmique Répartie

Travaux Dirigés (2), Master 1 Informatique SIAD

Élection et État Global

Exercice 1 Algorithme d'élection de Chang et Roberts

L'algorithme de Chang et Roberts, donné ci-dessous, est une solution au problème d'élection sur un anneau unidirectionnel.

```
var statep ;

begin  if p is initiator then
      begin  statep := cand ; send <tok, p> to Nextp ;
            repeat receive <tok, q> ;
                  if q = p then statep := leader
                  else  if q < p then
                        begin  if statep = cand then statep := lost ;
                              send <tok, q> to Nextp
                        end
                  until statep = leader
            end
      else repeat receive <tok, q> ; send <tok, q> to Nextp ;
            if statep = sleep then statep := lost
            until false
end
```

Un processus peut être soit initiateur soit non-initiateur. L'initiateur de plus petite identité est sélectionné. Chaque initiateur émet sur l'anneau un jeton contenant son identité. Un initiateur p supprime de l'anneau le jeton $\langle \text{tok}, q \rangle$ si $q > p$. Un initiateur devient `lost` quand un jeton avec une identité $q < p$ est reçu, et `leader` lorsque le jeton avec l'identité p est reçu.

Les processus non-initiateurs ne sont pas élus mais tous entrent dans l'état `lost` lorsqu'ils reçoivent le jeton d'un initiateur.

Il est supposé que les canaux sont FIFO.

Questions :

1. Supposer que chaque processus sur l'anneau est un initiateur. Pour quelle distribution des identités sur l'anneau, la complexité en nombre de messages est minimale ?
2. Combien de messages sont échangés dans ce cas ?

Exercice 2 Algorithme d'élection de LeLann

L'algorithme de LeLann, donné ci-dessous, est une solution au problème d'élection sur un anneau **unidirectionnel**.

```
Var      Listp : set of P init {p};
         statep;

begin if p is initiator then
    begin statep := cand ; send <tok, p> to Nextp ; receive<tok, q>;
        while q ≠ p do
            begin Listp := Listp ∪ {q};
                send<tok, q> to Nextp; receive<tok, q>
            end;
            if p = min(Listp) then statep := leader
                else statep := lost
            end
        end
    else repeat receive <tok, q> ; send <tok, q> to Nextp ;
        if statep = sleep then statep := lost
    until false
end
```

Dans l'algorithme de LeLann, chaque initiateur calcule une liste des identités de **tous** les initiateurs, suite à quoi l'initiateur d'identité la plus petite devient l'élu. Chaque initiateur émet un jeton, contenant son identité, sur l'anneau et ce jeton est retransmis par tous les processus. Il est supposé que chaque initiateur a une identité unique, que les canaux sont FIFO et qu'un initiateur doit émettre son propre jeton avant que le jeton d'un quelconque autre initiateur ne soit reçu (quand un processus reçoit un jeton, il ne peut plus initier l'algorithme par la suite). Quand un processus p reçoit son propre jeton, les jetons de tous les autres initiateurs ont déjà atteint p , et p devient l'élu si et seulement si p a l'identité la plus petite parmi tous les initiateurs.

1. Est-ce que cet algorithme fonctionne correctement avec un nombre inconnu de processus ? Expliquez.
2. Est-ce que cet algorithme fonctionne correctement lorsque les processus ne se connaissent pas mutuellement ? Expliquez.
3. Expliquez comment cet algorithme assure qu'un seul processus est élu à chaque élection (initiation) ?
4. Est-ce que **tous** les processus finissent par prendre connaissance de l'identité du processus élu ? Si oui expliquez, sinon proposez une solution qui le permette.
5. Est-ce que tous les processus prennent connaissance de la fin de l'élection ? Si oui expliquez, sinon proposez une solution qui le permette.
6. Donnez la complexité de l'algorithme, en nombre de messages, dans les 2 cas suivants (on supposera un nombre total de N processus) :
 - a. Un seul processus est initiateur,
 - b. Tous les processus sont initiateurs.

Exercice 3 **Algorithme de calcul d'état global de Chandy et Lamport**

Le principe de l'algorithme de calcul d'état global de Chandy et Lamport est très simple. Les processus s'informent mutuellement de la construction d'un état global en s'envoyant des messages de contrôle (marqueurs) mkr sur chaque liaison. Chaque site envoie un marqueur exactement une fois à chacun de ses voisins, au moment où il enregistre son état local. Donc, lorsqu'un site reçoit un marqueur, s'il n'a pas encore enregistré son état local, il le fait et envoie un marqueur à chacun de ses voisins.

L'algorithme exécuté par le site p_i utilise les variables suivantes :

```
voisins(i)                /* table des voisins du site  $p_i$  */  
enregistré(i)            /* indique si l'état local a été enregistré, init à faux */  
site                      /* variable locale, un des voisins */
```

Les fonctions exécutées par le site p_i sont :

Initialiser ()

```
enregistrer l'état local;  
enregistré(i) := vrai;  
pour tout site appartenant à voisins(i) faire  
    envoyer (mkr) à site  
fait
```

Recevoir_mkr ()

```
si enregistré(i) = faux alors  
    enregistrer l'état local;  
    enregistré(i) := vrai;  
    pour tout site appartenant à voisins(i) faire  
        envoyer (mkr) à site  
fait  
fsi
```

1. Cet algorithme ne fonctionne que sous l'hypothèse que les liaisons sont FIFO. Montrer que les marqueurs assurent alors que les deux conditions (propriétés 1 et 2) requises pour un algorithme de calcul d'état global sont remplies. On se place comme d'habitude dans la situation où le réseau est connexe et les communications sont fiables.

Propriété 1 : Si au moins un site initialise l'algorithme, alors tous les sites enregistrent leur état local au bout d'un temps fini.

Propriété 2 : L'algorithme de Chandy et Lamport calcule un état global significatif au bout d'un temps fini dès lors qu'il a été initialisé par au moins un site.

2. Calculer la complexité (en nombre de messages) de l'algorithme.