TD 6 - Élection

Janvier 2008

Contexte

- ✯ les sites ont tous des identités distinctes et le savent ; ils ne sont pas forcément placés sur l'anneau dans l'ordre de leurs identités ;
- ✯ aucun site ne connaît la taille de l'anneau ;
- ✯ les liaisons sont bidirectionnelles ;
- ✯ les communications sont fiables (les messages arrivent au bout d'un temps fini mais quelconque).

Exercice(s)

Exercice 1 – Élection sur un anneau bi-directionnel (Hirschberg et Sinclair)

Le but de cet exercice est de construire un algorithme qui réalise l'élection sur un anneau du site de plus grande identité parmi les initiateurs.

Le principe est le suivant : un site qui lance une élection exécute plusieurs étapes successivement. A l'étape k, le site envoie sur l'anneau dans les deux directions un jeton de type `out` contenant son identité. Ces jetons parcourent une distance 2^k avant de revenir à leur émetteur. On associe un type `in` aux jetons sur le chemin du retour (voir figure 1).

Si un site reçoit ses deux jetons `in`, il passe à l'étape $(k+1)$. Cependant, un jeton peut être détruit au cours de son trajet. En effet, le comportement d'un site est le suivant : si un site j voit passer un jeton `out` contenant l'identité i, alors

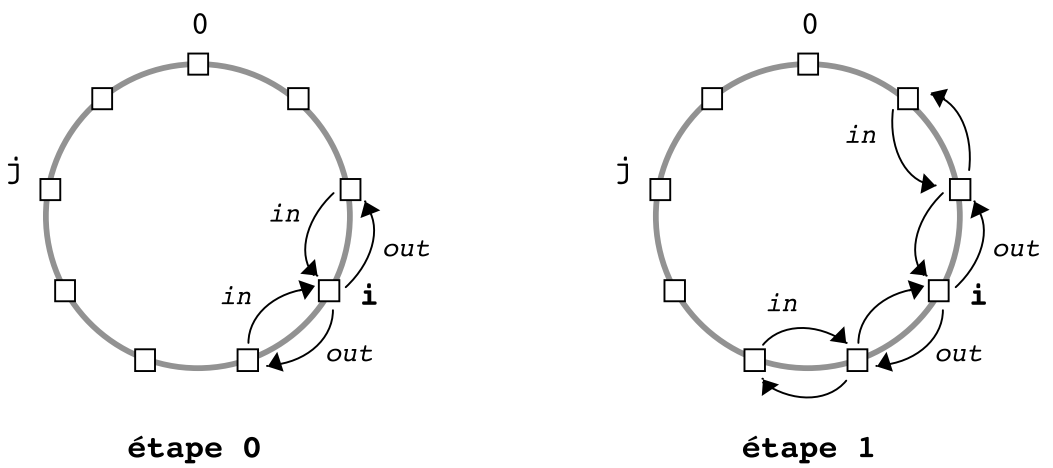

FIGURE 1 – Mécanisme de l'algorithme

– si $i > j$ ou si j n'est pas initiateur, j transmet le jeton à son voisin dans le sens de circulation (ou éventuellement renvoie celui-ci vers i s'il est en bout de parcours) ;
– si $i < j$ et j est initiateur, le jeton est détruit (j ne le transmet pas).

À partir du moment où il voit passer un jeton, un site ne peut plus devenir initiateur.

Question 1

A quel moment un site sait-il qu'il est battu ? Justifiez.

Solution:

Un site est battu

– s'il n'est pas initiateur, dès qu'il voit passer un jeton : en effet, il ne peut plus devenir initiateur et perd donc toute chance d'être élu.
– s'il est initiateur, quand il voit passer un jeton portant une valeur initiateur supérieure à son identité. Il sait alors qu'il ne pourra pas remplir la condition d'élection.

Fin solution

Question 2

A quel moment un site sait-il qu'il est élu ? Justifiez. Qu'en serait-il si la taille de l'anneau était connue ?

Solution:

Quand la taille de l'anneau n'est pas connue, le retour de deux messages `in` ne permet pas à un site de conclure. En effet, il ne peut pas savoir si tous les sites de l'anneau ont été visités. En particulier, il se peut qu'un site initiateur de plus forte identité n'ait pas vu passer son jeton.

Par contre, s'il voit passer un message `out` portant son identité, c'est que ce message a fait le tour de l'anneau. Il a donc parcouru tous les sites sans rencontrer d'initiateur d'identité supérieure. Le site peut donc en conclure qu'il est élu.

Si la taille de l'anneau est connue, soit N cette taille. La distance maximum entre deux sites est alors $\lfloor \frac{N}{2} \rfloor$($\lfloor\ \rfloor$ = partie entière par valeur inférieure). Si, à l'étape k telle que $2^k \geq \lfloor \frac{N}{2} \rfloor$, un site voit revenir ses deux messages `in`, tous les autres sites ont donc été parcourus sans détruire de message. Le site est par conséquent élu. Avec l'étape 0, on a un total de ($\log_2 \lfloor \frac{N}{2} \rfloor + 1$) (arrondi par valeur supérieure) étapes au plus.

Fin solution

On suppose que les messages qui circulent dans l'algorithme sont du type

```
<emetteur, initiateur, type, distance>
```

où émetteur est l'identité du dernier site qui a relayé le message (donc nécessairement le voisin de gauche ou le voisin de droite du récepteur), initiateur l'identité du premier site qui a émis ce message (et qui est donc forcément un initiateur de l'élection), type peut prendre les valeurs `in` et `out`, et `distance` est la distance qu'il reste à parcourir à un message `out` avant de faire demi-tour (donc de se transformer en message `in`).

Question 3

Pourquoi est-il utile pour le récepteur de connaître l'identité de l'émetteur ?

Solution:

Un site peut recevoir un message de son voisin de gauche comme de celui de droite. Il a besoin de savoir qui l'a émis simplement pour savoir à qui le retransmettre.

Fin solution

Question 4

Comment la variable distance évolue-t-elle au cours de la circulation d'un message `out` ?

Solution:

La variable distance n'a d'utilité que pour les messages `out`. Elle est initialisée par l'initiateur à la valeur correspondant à la distance que doit parcourir le message, soit 2^k à l'étape k. Chaque site qui fait passer le message décrémente cette distance de 1 avant de transmettre le message à son voisin.

Fin solution

Le site i gère les variables VG(i) et VD(i) contenant les identités de son voisin de gauche et de son voisin de droite respectivement, ainsi que toute autre variable que vous jugerez nécessaire. On suppose que les sites disposent en outre de deux primitives `envoyer_VG(mes)` et `envoyer_VD(mes)` permettant à un site d'envoyer un message sur l'anneau à son voisin de gauche et à son voisin de droite respectivement.

Question 5

Ecrivez la procédure `recevoir(<emetteur, initiateur, out, distance>)` exécutée par le site i.

Solution:

Réception d'un message out par le site i : comme dans tout algorithme d'élection, chaque site doit gérer une variable état lui donnant sa situation vis-à-vis de l'élection (élu, battu, ne sait pas (nsp)). Initialement, état = nsp pour tous les sites. De plus, il faut une variable permettant de savoir si le site qui reçoit le message est ou non lui-même initiateur. En effet, dans le deuxième cas il est systématiquement battu alors que dans le premier, cela dépend de l'identité de l'initiateur véhiculée par le message.

```
recevoir (<emetteur, initiateur, out, distance>) {
   si (init[i] == faux ou initiateur > i) {
      etat[i] = battu
      si (distance > 1) {
         si ( emetteur == VD(i) ) {
            envoyer_VG(<i, initiateur, out, distance - 1>)
         } sinon {
            envoyer_VD(<i, initiateur, out, distance - 1>)
         }
      } sinon {
         si ( emetteur == VD(i) ) {
            envoyer_VD(<i, initiateur, in, - >)
         } sinon {
            envoyer_VG(<i, initiateur, in, - >)
         }
      }
   } sinon {
      si ( initiateur ==  i ) {
         etat[i] = elu ;
      }
   }
}
```

Fin solution

Question 6

Ecrivez la procédure `recevoir(<emetteur, initiateur, in, distance>)` exécutée par le site i.

Solution:

Il faut pour cette procédure que les sites gèrent une variable leur permettant de savoir si, lorsqu'ils reçoivent un message in portant leur identité, celui-ci est le premier ou le deuxième. En effet, ils ne lancent l'étape suivante que s'ils ont reçu deux messages.

```
recevoir (<emetteur, initiateur, in, distance>) {
   si ( initiateur != i ) {
      si ( emetteur == VD(i) ) {
         envoyer_VG(<i, initiateur, in, distance>)
```

```
      } sinon {
         envoyer_VD(<i, initiateur, in, distance>)
      }
   } sinon {
      nb_in[i]++;
      si (nb_in[i]  ==  2) {
         initier_etape(k) ;
      }
   }
}
```

Fin solution

Question 7

Ecrivez la procédure `initier_etape(k)` exécutée par le site i lorsqu'il débute l'étape k. Quelle procédure le site i exécute-t-il pour lancer une élection ?

Solution:

On décrit explicitement dans cette question la procédure `initier_etape(k)` utilisée à la question précédente.

```
initier_etape(k) {
   si ( k == 0 ) {
      init[i] = vrai ;
   }
   nb_in[i] = 0 ;
   envoyer_VD(<i, i, out, 2^k>);
   envoyer_VG(<i, i, out, 2^k>);
   k++;
}
```

Un site lance une élection en exécutant `initier_etape(0)`.

Fin solution

Exercice 2 – Complexité de l'algorithme de Hirschberg et Sinclair

Le but de cet exercice est de déterminer la complexité en nombre total de messages échangés et la complexité en temps de l'algorithme présenté dans l'exercice précédent. On se place dans le cas où les sites ne connaissent pas la taille de l'anneau.

Question 1

Quel est le nombre d'étapes qu'un site doit exécuter avant de savoir qu'il est élu ?

Solution:

D'après la question **??** de l'exercice précédent, un site sait qu'il est élu dès qu'il voit revenir un de ses messages `out`. Ceci se produit à l'étape k telle que $2^{k-1} < N \leq 2^k$, où N est la taille de l'anneau. A cet instant, le site a exécuté $(k+1)$ étapes, puisqu'il commence par l'étape 0. Le nombre d'étapes est donc :

$$K = \lceil \log_2 N \rceil + 1$$

($\lceil\ \rceil$ = partie entière par valeur supérieure)

Fin solution

Question 2

On considère les deux configurations d'anneau de la figure 2. On suppose que tous les sites sont initiateurs. Donnez, pour chaque cas, le nombre de sites qui exécutent l'étape 1 et l'étape 2.

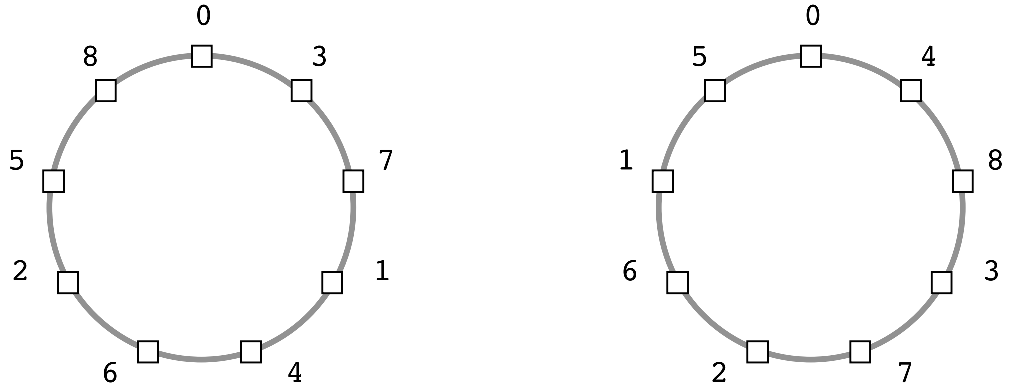

FIGURE 2 – Configurations d'anneaux

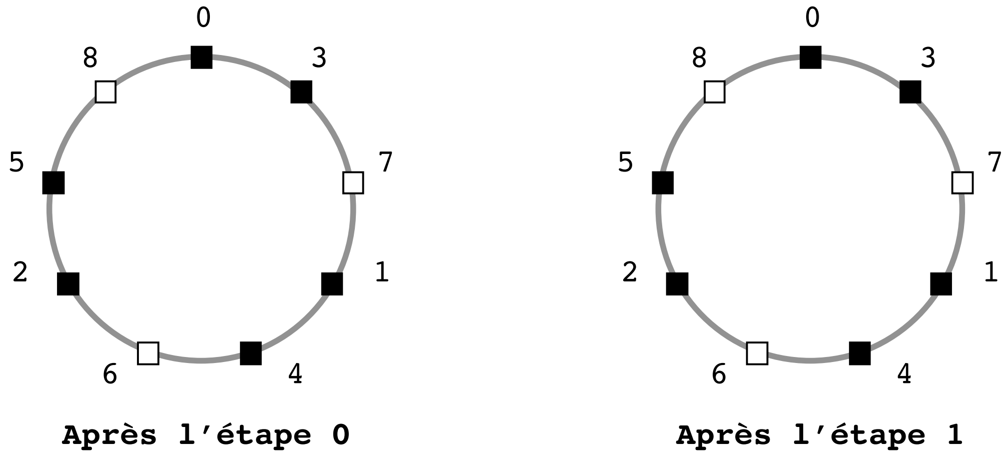

FIGURE 3 – Exécution pour la configuration 1

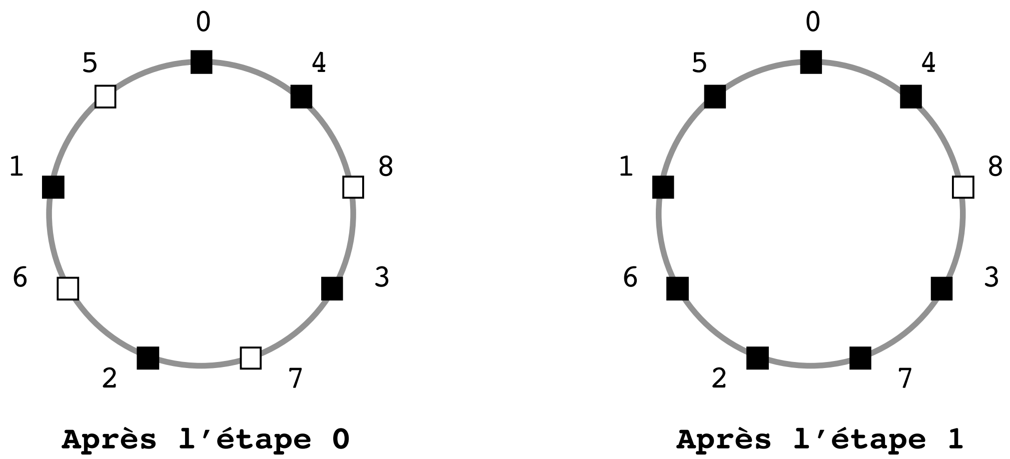

FIGURE 4 – Exécution pour la configuration 2

Solution:
On ne représente pas ici les messages qui circulent à chaque étape mais uniquement les sites éliminés (en noir). Les résultats sont donnés dans les figures 3 et 4.
Fin solution

Question 3

En raisonnant sur le nombre de sites éliminés à chaque étape, en déduire, dans le cas général, le nombre maximum de sites qui exécutent :
- l'étape 1
- l'étape 2
- l'étape k

Solution:
Soit A_k le nombre maximum de sites actifs à l'étape k.

Un site n'exécute l'étape 1 que s'il a reçu deux messages `in` à l'étape 0. Dans ce cas, ses deux voisins sont éliminés. Par conséquent, au plus un site sur deux atteint l'étape 1. On a donc

$$A_1 = \left\lfloor \frac{N}{2} \right\rfloor$$

Un site n'exécute l'étape 2 que s'il est sorti vainqueur de l'étape 1, ce qui signifie qu'il a éliminé ses voisins jusqu'à une distance 2. Sur une séquence de 3 sites, on en a donc au plus un actif.

$$A_2 = \left\lfloor \frac{N}{3} \right\rfloor$$

Un site n'exécute l'étape k que s'il est sorti vainqueur de l'étape $(k-1)$, ce qui signifie qu'il a éliminé ses voisins jusqu'à une distance 2^{k-1}. Sur une séquence de $(2^{k-1}+1)$ sites, on en a donc au plus un actif.

$$A_k = \left\lfloor \frac{N}{2^{k-1}+1} \right\rfloor$$

Fin solution

On considère qu'un message circule d'un site vers l'un de ses voisins. Lorsque ce dernier relaie le message, cette retransmission est considérée comme une nouvelle émission.

Question 4

Quel est le nombre total maximum de messages émis à l'étape k ? En déduire la complexité de l'algorithme en nombre de messages.
Solution:
A l'étape k, il y a au plus A_k sites actifs. Chacun de ces sites envoie 2 messages qui parcourent chacun une distance de 2^k, ce qui représente un total de 2^{k+1} messages. Comme de plus, il s'agit d'un parcours aller-retour, on a un nombre total de messages à l'étape k :

$$M_k = 2^{k+2}.A_k = 2^{k+2}.\left\lfloor \frac{N}{2^{k-1}+1} \right\rfloor \leq 2^{k+2}.\frac{N}{2^{k-1}+1} \leq \frac{2^{k+2}.N}{2^{k-1}} \leq 8.N$$

On peut noter que ce nombre de messages est en fait indépendant de l'étape. On a donc pour l'algorithme une complexité globale en nombre de messages $\leq 8NK$, soit une complexité en $O(N.\log_2 N)$.
Fin solution

On suppose que la durée de transmission d'un message d'un site vers un de ses voisins est d'une unité de temps.

Question 5

Quelle est la durée totale nécessaire au site qui gagne l'élection pour accomplir

- la dernière étape ?
- l'avant-dernière ?
- l'ensemble de toutes les autres étapes ?

En déduire une borne au temps nécessaire à ce site pour accomplir toutes les étapes.

Solution:

Durée nécessaire à l'accomplissement de la dernière étape : celle-ci est terminée lorsqu'un message a fait le tour de l'anneau, soit après un temps N.

Durée nécessaire à l'accomplissement de l'avant-dernière étape : la durée nécessaire à l'exécution de toute étape k est de 2.2^k (durée nécessaire pour l'aller-retour d'un message envoyé à une distance 2^k). L'avant-dernière étape est telle que $k = \lceil \log_2 N \rceil - 1$. Donc la durée nécessaire est de $2^{\lceil \log_2 N \rceil}$.

Durée nécessaire à l'accomplissement de toutes les étapes précédentes :

$$\sum_{k=0}^{\lceil \log_2 N \rceil - 2} 2^{k+1} = \sum_{k=1}^{\lceil \log_2 N \rceil - 1} 2^k = 2.(2^{\lceil \log_2 N \rceil - 1} - 1) \leq 2.^{\lceil \log_2 N \rceil}$$

Or,

$$\log_2 a \leq \lceil \log_2 a \rceil < \log_2 a + 1$$

d'où $a \leq 2^{\lceil \log_2 a \rceil} < 2a$.

La durée totale nécessaire au site qui gagne l'élection pour savoir qu'il a gagné est donc bornée par $N + 2N + 2N = 5N$.

Pour obtenir la complexité globale de l'algorithme en temps, il faudrait en plus calculer le délai maximum entre la première et la dernière initialisation, le pire des cas correspondant au cas où le site vainqueur est le dernier initiateur. On peut montrer (mais le calcul est fastidieux) que cette valeur est bornée par $3N$. On a donc une complexité globale en $O(N)$.

Fin solution