Recueil d'exercices et problèmes partiellement solutionnés sur les algorithmes distribués

Elaboré par Pr. Djamel Eddine SAIDOUNI
Laboratoire MISC, Département IFA, Faculté des NTIC, Université Constantine 2 – Andelhamid Mehri
saidounid@hotmail.com

Partie : Problème de la section critique

Exercice 1:
Donner une comparaison entre l'algorithme de Ricard et Agrawala, et celui de Carvalho et Roucairol résolvant chacun le problème de l'accès à la section critique dans un contexte distribué.

Corrigé :
Dans la comparaison on doit mentionner les points qui différencient les deux algorithmes.

	Ricard et Agrawala	**Carvalho et Roucairol**
Nombre de messages de contrôle	Nombre = 2*(n-1)	0 <= Nombre <= 2*(n-1)
Temps d'accès à la SC lorsqu'elle est libre où **T** est le temps moyen de transit d'un message entre deux sites	Temps = 2 * T	0 <= Temps <= 2 * T
Adaptativité	Non adaptatif	Adaptatif
Bornétude des variables	Variables bornées	Variables non bornées

Exercice 2
Remplissez le tableau récapitulatif résumant les propriétés de quelques algorithmes distribués résolvant le problème de la section critique pour un processus Pi. En se fixe les hypothèses suivantes

- **N** étant le nombre de sites.
- Sur chaque site il existe un seul processus.
- **T** est le temps moyen de transit d'un message entre deux sites.

Nom de l'alg.	Nombre Message requête	Nombre message permission	Nombre message retour permission	Temps d'entrée en S.C. lorsqu'elle est libre	Adaptatif
Ricart Agrawala					
Carvalo Roucairol					
Misra Chandi					
Alg. Mixte					

Corrigé :

Nom de l'alg.	Nombre Message requête	Nombre message permission	Nombre message retour permission	Temps d'entrée en S.C. lorsqu'elle est libre	Adaptatif
Ricart Agrawala	N-1	N-1	0	2*T	Non

Carvalo Roucairol	0≤nbr≤N-1	0≤nbr≤N-1	0	0≤temps≤2*T	Oui
Misra Chandi	0≤nbr≤N-1	0≤nbr≤N-1	0	0≤temps≤2*T	Oui
Alg. Mixte	Card(Ri)	Card(Ri)	Card(Ri)	3*T	Non

Exercice 3

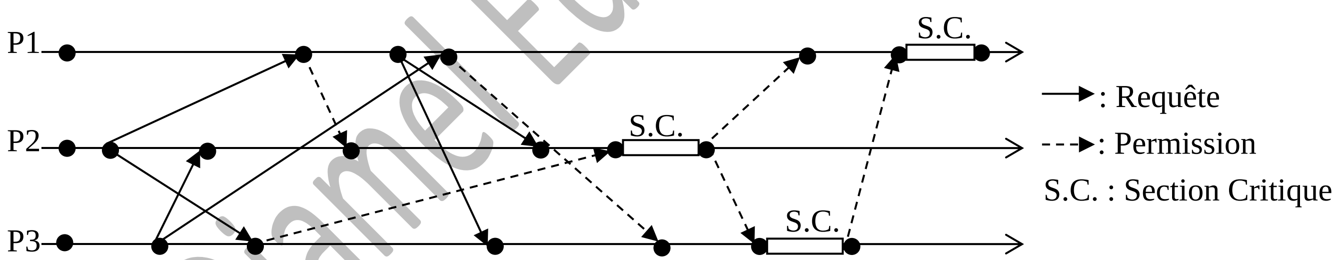
Soit un anneau unidirectionnel reliant **n** sites. On suppose que les liaisons sont fiables.

1. Proposez un algorithme distribué d'exclusion mutuelle basé sur la circulation perpétuelle d'un jeton de contrôle sur l'anneau (**NB** : énumérez d'abord les évènements qui occurrent au niveau d'un site i).
2. Montrez la validité de l'algorithme proposé.

Corrigé :

Exercice 4

1. Quelles sont les variables utilisées dans l'algorithme de Ricart et Agrawala ? donnez le rôle de chacune d'elles.
2. Rappelez
   2.1. Le protocole de mise à jour des horloges logiques.
   2.2. Le contenu d'un message Requête
   2.3. L'expression qui calcule la priorité d'un site **i** recevant une requête de la part d'un site **j**.
3. On considère un système distribué constitué de 3 sites P1, P2 et P3 s'envoyant des messages de façon asynchrone comme représenté par la figure suivante. Les événements d'un processus, représentés par des gros points noirs, sont soit des envois ou des réceptions de messages. Ces événements sont datés par un système d'horloge logique, où chaque processus possède sa propre horloge initialisée à 0.
   En utilisant l'algorithme de Ricart et Agrawala, indiquez au dessus de chaque point de la figure : la **valeur de l'horloge ;** le **contenu du message** ; l'**état du site** ; le **contenu des vecteurs Attendu$_i$ et Différé$_i$**.



4. Peut-on borner les horloges logiques dans cet algorithme ? Justifiez votre réponse.
5. Cet algorithme est de quel type ?
6. Donnez les avantages et les inconvénients de cet algorithme.
7. Citez un algorithme de même type qui surmonte les inconvénients de l'algorithme de Ricard et Agrawala sans pour autant conserver ses avantages. Expliquez son principe.
8. Citez un algorithme de même type qui surmonte les inconvénients de l'algorithme de Ricard et Agrawala tout en conservant ses avantages. Expliquez son principe.
9. Montrez que l'expression qui calcule la priorité d'un site **i** recevant une requête de la part d'un site **j** est équivalente à :

(étati = dedans) ou ((étati = demandeur) et ((lasti,i) < (lastj,j)))

Corrigé :

1. Quelles sont les variables utilisées dans l'algorithme de Ricart et Agrawala ? donnez le rôle de chacune d'elles.

Variable	Rôle
$etat_i$ = {dehors, demandeur, dedans} initialisé à dehors	Indique l'état du processus Dehors : le processus n'est pas intéressé par la section critique Demandeur : le processus désire accéder à la section critique Dedans : le processus est en section critique
h_i : entier croissant initialisé 0	Horloge logique utilisée pour dater les événements
$last_i$: entier croissant initialisé 0	Mémorise la date de la dernière demande du processus i
$attendu_i$: ensemble des sites initialisé à Ø	Ensemble des processus à partir desquels le processus i attend des permissions
$differe_i$: ensemble des sites initialisé à Ø	Ensemble des processus dont le processus i a retardé l'envoi de permission
$priorité_i$:booléen	Utilisé pour calculer la priorité

2. Rappelez

2.1. Le protocole de mise à jour des horloges logiques.

Lors d'un appel à acquérir :

$h_i = h_i + 1$;

Lors de la réception d'une requête de la part d'un site j avec une estampille (k,j) :

$h_i = \max (h_i , k)$;

2.2. Le contenu d'un message Requête :

Le message Requête contient :

$last_i$: la date d'émission de la requête

i : l'identité du site émetteur

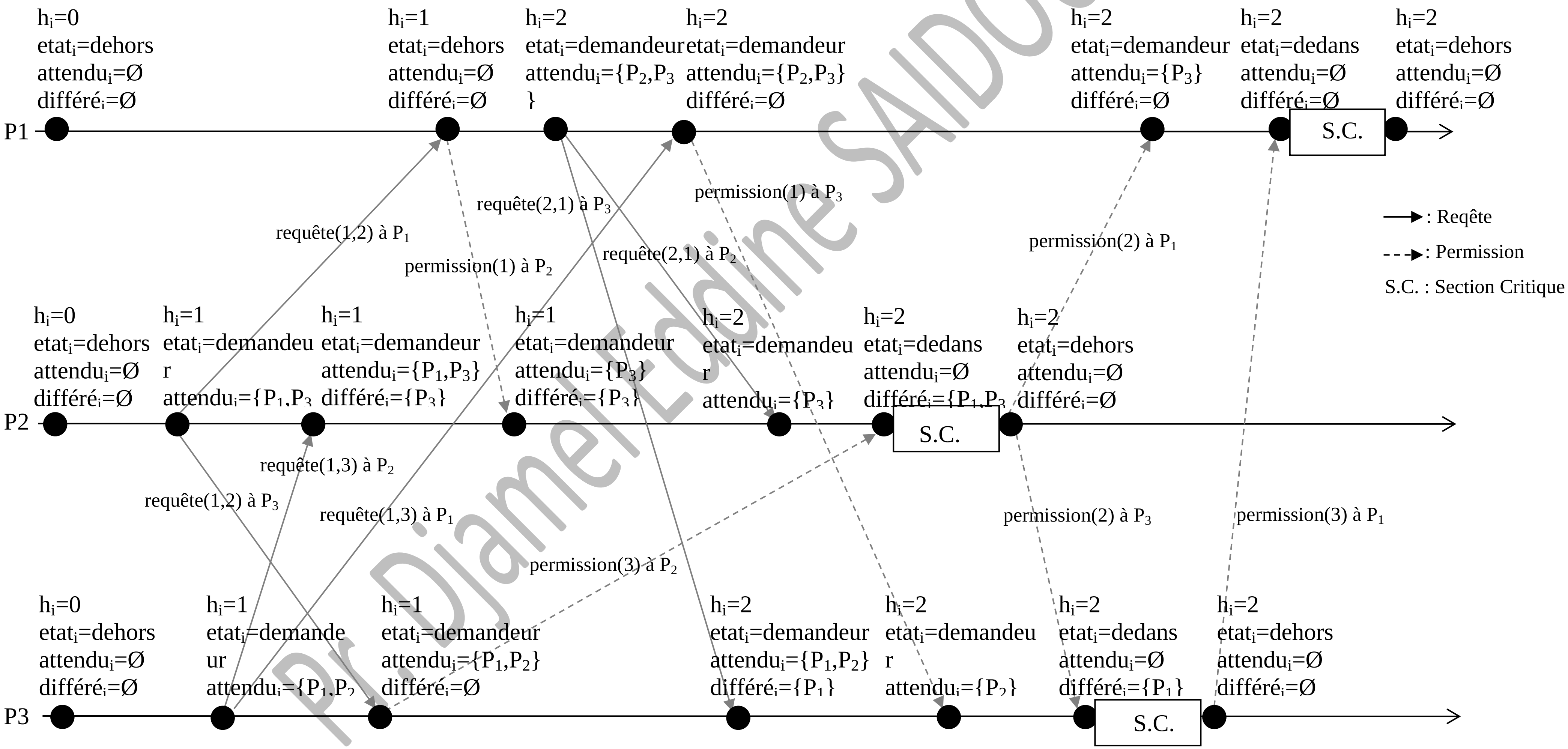
2.3. L'expression qui calcule la priorité d'un site **i** recevant une requête de la part d'un site **j**.

priorité = ($etat_i \neq$ dehors) et (($last_i$, i) < (k , j))

ou bien :

priorité = ($etat_i \neq$ dehors) et (($last_i$ < k) ou (($last_i$ == k) et (i < j)))

3. On considère un système distribué constitué de 3 sites P1, P2 et P3 s'envoyant des messages de façon asynchrone comme représenté par la figure suivante. Les événements d'un processus, représentés par des gros points noirs, sont soit des envois ou des réceptions de messages. Ces événements sont datés par un système d'horloge logique, où chaque processus possède sa propre horloge initialisée à 0.
   En utilisant l'algorithme de Ricart et Agrawala, indiquez au dessus de chaque point de la figure : la **valeur de l'horloge ;** le **contenu du message** ; l'**état du site** ; le **contenu des vecteurs Attendu$_i$ et Différé$_i$**.

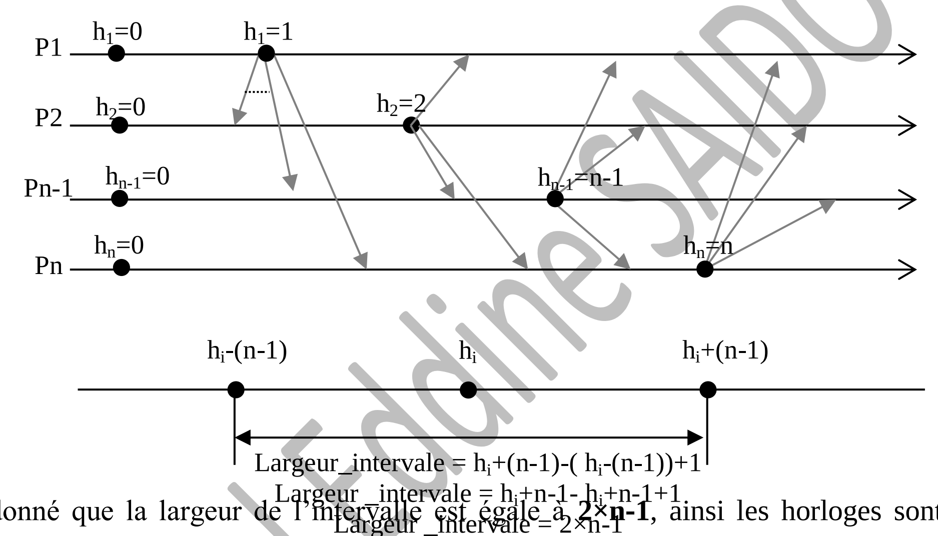


4. Peut-on borner les horloges logiques dans cet algorithme ? Justifiez votre réponse.

Oui, les horloges peuvent être bornées dans cet algorithme. En effet si **h** est la plus grande date associée à une requête, la prochaine date utilisée par une requête ne peut pas être supérieure à **h+1**. En supposant qu'il y a **n** sites et qu'un site ne peut pas faire une nouvelle requête que lorsque la précédente a été satisfaite, l'écart maximal entre les dates de deux requête est borné par **n-1**. Si l'on se place sur un site **i**, on sait donc que les autres horloges ont des valeurs dans l'intervalle :

[hi-(n-1),hi+(n-1)]

Les 2 bornes de l'interval correspondent aux cas extrêmes où **hi** serait respectivement la plus grande et la plus petite valeur de l'horloge.



Etant donné que la largeur de l'intervalle est égale à **2×n-1**, ainsi les horloges sont implémenté **modulo 2×n-1.**

5. Cet algorithme est de quel type ?

Cet algorithme appartient à la classe des algorithmes basés sur les permissions individuelles, puisque chaque site i gère individuellement chacun des conflits qui le mettent en compétition avec les autres sites.

6. Donnez les avantages et les inconvénients de cet algorithme.

Avantage :

- Les horloges bornées

Inconvénients :

- Non adaptatif puisque si le processus **i** est le seul demandeur, alors il doit envoyer ses requêtes aux autres sites (qui sont dans l'état dehors) et il attend leurs permissions, ainsi il y aura une perte de temps vu que la section critique est libre.
- Chaque site doit connaître les identités des autres sites

7. Citez un algorithme de même type qui surmonte les inconvénients de l'algorithme de Ricart et Agrawala sans pour autant conserver ses avantages. Expliquez son principe.

   C'est l'algorithme de Carvalho et Roucairol puisqu'il est adaptatif mais il ne peut pas borner les horloges, son principe est le suivant :

   - Associer à tout couple de site distinct **i** et **j** un et un seul message **permission(i,j)**.
   - Si le message est chez **i**, celui-ci a la permission de **j** et inversement.
   - Initialement **permission(i,j)** est placé indifféremment sur l'un ou l'autre site.
   - Lorsqu'un site **i** invoque l'opération acquérir, il doit demander les permissions qui lui manquent, les sites qui détiennent ces permission constituent donc l'ensemble $\mathbf{R_i}$.
   - Lorsque le site **i** reçoit **permission(i,k)**, il supprime **k** de $\mathbf{R_i}$.
   - Lorsque le site **i** envoie **permission(i,m)**, il rajoute **m** à $\mathbf{R_i}$ et dans le cas où le site **i** est dans l'état demandeur, il doit envoyer un message requête au site **m** afin de la récupérer une fois que le site **m** libérera la section critique.

   L'application de ce principe permet de réduire le nombre de messages nécessaire à l'utilisation de la section critique. Par exemple si le processus i est le seul demandeur et il possède toute les permissions alors il n'envoie pas de requêtes aux autres sites (qui sont dans l'état dehors) et il accède directement la section critique vu qu'elle est libre. Cependant, il n'est pas possible de borner les horloges.

8. Citez un algorithme de même type qui surmonte les inconvénients de l'algorithme de Ricart et Agrawala tout en conservant ses avantages. Expliquez son principe.

   C'est l'algorithme de Chandy et Misra puisqu'il est adaptatif (il utilise le même principe de Carvalho et Roucairol pour la distribution initiale des messages permissions entre les différents sites) et il n'utilise pas des variables de taille bornée (horloges), son principe est le suivant :

   - Associer un état à chaque permission. Ainsi un message permission peut être dans l'état *util* (utilisé) ou **non-util** (non utilisé).
   - Initialement tout message permission est dans l'état util.
   - L'état de ce message est mémorisé dans **permission(i,j).etat**.
   - Lorsqu'un site **i** pénètre en section critique toutes ses permissions passent dans l'état **util**.

- Avant d'envoyer un message **permission(i,j)**, le site émetteur **i** passe l'état de ce message à **non-util**.
- Lorsque le site **i** reçoit un message requête du site **j**, est prioritaire sur **j** si le message **permission(i,j)** est dans l'état **non-util**, il diffère alors l'envoi de cette permission à sa sortie de la section critique. Dans le cas contraire, il envoie la permission par retour de courrier

<u>Exercice 5</u>

Dans l'algorithme de Ricard et Agrawala, un site i recevant une requête(K,j) calcule sa priorité par rapport à cette requête par la formule suivante :

Prioritéi := (étati <> dehors) et (lasti , i) < (K , j)

1. Montrez mathématiquement que cette formule est équivalente à la formule suivante :

   prioritéi := (étati = dedans) ou ((étati = demandeur) et (lasti , i) < (K,j))
2. A l'aide d'un contre exemple, montrez que la première formule est incorrecte pour le calcul de la priorité dans l'algorithme de Carvalho et Roucairol. Quelle est la conséquence de l'utilisation de cette formule dans cet algorithme ? Justifiez vos réponses.

<u>Corrigé :</u>

<u>Exercice 6</u>

On se propose d'étudier la validité d'un algorithme résolvant le problème de la section critique dans un environnement distribué. On admet les hypothèses suivantes :

- **N** un entier non nul qui désigne le nombre de sites. Chaque entier i ($0 \leq i \leq N-1$) désigne un site qu'on note Si.
- Chaque site héberge un et un seul processus. De ce fait un processus est identifié par le site où il est hébergé et inversement.
- A chaque site **Si** sont associées :
  - Une horloge logique **Hi** initialisée à **0**.
  - Une variable **Lasti** qui mémorise la date à laquelle le site est devenu demandeur.
  - Une file d'attente **Filei** ordonnée, initialement vide. Cette file contient les estampilles des sites demandeurs d'accès à la section critique ayant sollicités la permission du site **Si**.
  - **Laststampij** avec $0 \leq j \neq i \leq N-1$: Ensemble de variables. La variable Laststampij, initialisée à (0,j), mémorise la dernière estampille du message reçu par Si en provenance de Sj.

Le comportement d'un site (processus) *Si* est donné par les énoncés suivants :

Lors d'un appel à acquérir :

- Hi := Hi + 1
- Lasti := Hi
- Envoyer(Request(Lasti,i)) à tous les autres sites
- Enfiler((Lasti,i)) dans Filei.
- Attendre ((Lasti,i) en tête de Filei **et** ((Lasti,i) < Laststampij pour tout $0 \leq j \neq i \leq N-1$))

Lors d'un appel à libérer :

- Défilet((Lasti,i)) de Filei
- Envoyer (Release((Hi,i))) à tous les autres sites

Lors de la réception de Request (K,j) :

- Hi := max(Hi,K)
- Enfiler((K,j)) dans Filei
- Hi := Hi +1

- Laststampij := (K,j)
- Envoyer(Response(Hi,i) à j

Lors de la réception de Response(K,j)

- Hi := max(Hi,K)
- Laststampij := (K,j)

Lors de réception de Release(K,j)

- Défiler(l'estampille de Sj) de Filei

Questions :

Partie 1 : Compréhension de l'algorithme

1. L'exécution locale de chaque énoncé doit satisfaire une condition particulière, laquelle ? Justifier votre réponse.
2. Donner la fonction booléenne qui compare deux estampilles (L, i) et (K,j) où i et j sont les identificateurs des sites.
3. Les valeurs d'une variables Laststampij peuvent elle décroitre dans le temps ? Justifier votre réponse.
4. A travers un scénario, montrer la **nécessité** de l'hypothèse de canaux FIFO pour la propriété d'exclusion mutuelle.

Partie 2 : Analyse de l'algorithme

5. Sous cette hypothèse, montrer que la propriété d'exclusion mutuelle est assurée par cet algorithme. (**indication** : Procéder par l'absurde en supposant qu'à un moment donné deux sites Si et Sj sont en section critique en même temps)
6. Montrez que la propriété de vivacité locale est assurée (chaque site demandeur arrivera à accéder à sa section critique). (**indication :** Procéder par l'absurde en supposant que Si et Sj sont demandeurs d'accès à la section critique avec l'hypothèse que l'estampille de Si et inferieur à l'estampille de Sj et que Sj peut rentrer à sa section critique avant Si).
7. Dans le cas où un site tombe en panne, que ce passera t-il ?
8. Calculer le nombre de message mis en jeux pour chaque entrée en section critique.
9. Le nombre de message mis en jeux peut être diminué par la diminution des messages « Response ». Donnez cette amélioration ainsi que le nouveau nombre de messages.

Partie 3 : Amélioration de l'algorithme

Un diplômé d'un master en informatique à constaté que pour son application les sites se divisent en deux catégories :

- **Catégorie1** : Sites qui demandent rarement l'accès à la section critique,
- **Catégorie2** : Sites qui demandent très souvent l'accès à la section critique.

De ce fait, il propose une modification de l'algorithme afin de diminuer le nombre de message mis en jeux pour l'accès à la section critique.

Modification : En dehors d'une catégorie, seuls les processus de la catégorie2 sont sollicités par les processus demandeurs d'accès à la section critique.

10. Apporter les modifications nécessaires aux énoncés de l'algorithme.
11. Soit **n1** (respectivement **n2**) le cardinal de Catégorie1 (respectivement Catégorie2).
    a. Quel est le nombre de messages de contrôle mis en œuvre par un processus de Catégorie1 pour accéder à sa section critique.
    b. Même question pour un processus de Catégorie2.
12. Soit **T** une période de temps. On suppose qu'à chaque période **T**, chaque site de Catégorie2 demande l'accès à sa section critique, alors que chaque site de Catégorie1 demande l'accès à chaque période **f*T** (f est un entier positif).

a. Dans la période T, quel est le nombre de message engendrés par les processus de Catégorie2.
b. Dans la période f*T, quel est le nombre de message engendrés par les processus de Catégorie1.
c. Dans la période f*T, quel est le nombre total de message engendré par tous les processus.
d. Discuter le cas où f = 1 et n1 = 0.

13. La sûreté de l'algorithme est-elle préservée pour n1 = n2 = 1 ? Argumenter votre réponse.

<u>Corrigé :</u>

Partie 1 :

1. La condition particulière que chaque énoncé doit satisfaire est la propriété d'atomicité sauf pour l'instruction Attendre qui est intérruptible.
   Justification : Remarque par exemple que la variable locale Hi est manipulée en écriture dans les deux procédure concurrentes « **Appel à acquérir** » et « **Réception d'une requête** » de ce fait, cette variable doit être manipulée en EM.

2. La fonction booléenne qui compare deux estampilles, (L,i) et (K,j) est basée sur l'ordre lexicographique :
   $(L,i) < (K,j)$ ssi $(L < k)$ ou $(L=K$ et $i < j)$
3. Les valeurs de Laststampij ne vont que croitre car les horloges des sites sont synchronisées à chaque évènement et que **les liaisons sont FIFO**.
4. Sans l'hypothèse FIFO des liaisons, l'exclusion mutuelle est non garantie. Exemple : Soient les deux sites i et j respectivement avec $i < j$ demandeurs d'accès à la SC en même temps. Acceptant que la requête du site i arrive très en retard au site j (canaux non FIFO). J décide de rentrer dans sa SC étant donné qu'il est en tête de filej et que Response de i est arrivée. Après l'arrivée de Response de j chez i et comme l'estampille de i est inférieur donc i est en tête de sa file. Par conséquent il rentre dans sa SC.

Deuxième partie :

5. Procédons par l'absurde : Si et Sj en SC. Pour cela la condition de franchissement est vérifiée par i et j. A un instant t, les deux requêtes de Si et Sj sont en tête de filei et filej respectivement. Supposons maintenant que (Lasti,i) < (Lastj,j), à la réception de des messages Response de Si par Sj et de Sj par Si, la tête de file de j est (Lasti,i) ce qui contredit l'hypothèse que (Lastj,j) est aussi tête de file de Sj.
6. Supposons que (Lasti,i) < (Lastj,j) et Sj exécute la Sc avant Si. Pour que Sj exécute la Sc il doit être en tête de sa file d'attente et Laststampji > (Lastj,j). Or (Lasti,i) < (Lastj,j) et donc elle précède la requête de j dans filej, d'où la contradiction. La vivacité locale (individuelle) est vérifiée.
7. Interblocage.
8. 3*(N-1)
9. Amélioration :
   Enoncé Réception d'une requête :
   Si (Lasti,i) not supérieur Laststampij) alors Envoyer Response (Hi,i) à j

   2*(N-1) <= Nombre de message <= 3*(N-1)

Troisième partie :

Convention : Processus de Catégorie 2 numérotés de 0 à n2 -1

10. Enoncé Acquérir : (Juste les sites de la catégorie 2)
    Attendre …… (Lasti,i) < Laststampij pour (0<= j<> i <= n2-1)
11. Reponse
    a. 3*(N-1)
    b. 2*(N2 – 1) + (N -1)
12. Reponse
    a. 3 * (N2 – 1) ou N2 * 3*(N2 -1) si on suppose que la durée de la Sc est negligeable
    b. 3*(N -1) + (N1 -1) Ou bien si on suppose que la durée de la Sc est negligeable (N1 * 3 * (N-1))
    c. f*(11.a) + 11.b =
    d. Dans ce cas N2 = N et en retombe sur le premier algorithme
13. La sûreté n'est pas assurée (voir exemple)

<u>Exercice 7 :</u>
Soient les éléments de réponses suivants :

a) En ce qui concerne les processus qui m'ont sollicité vous pouvez entrer en section critique.
b) Principe de l'égalité de la responsabilité.
c) En ce qui me concerne vous pouvez entrer en section critique.
d) Principe de l'égalité de l'effort.

Pour un algorithme résolvant le problème de la section critique, choisissez la réponse qui convient :

1) Pour Di = ensemble des processus qui sollicitent le processus Pi. La sémantique de |Di|=Constante est : a b c d
2) Si l'algorithme est à permission d'arbitre, la sémantique d'une permission donnée par un processus est : a b c d
3) Pour Ri=ensemble des processus sollicités par le processus Pi. La sémantique de |Ri|=Constante est : a b c d
4) Si l'algorithme est à permission individuelle, la sémantique d'une permission donnée par un processus est : a b c d

<u>Corrigé :</u>

<u>Exercice 7 :</u>
Parmi les problèmes posés par les algorithmes basés sur les jetons et résolvant le problème de la section critique sur un anneau on trouve le problème de la perte du jeton et la nécessité de sa régénération. Dans cet exercice on se propose d'étudier un des algorithmes résolvant ce problème. On suppose l'existence de n processus sur n sites répartis sur un anneau.

Le principe de cet algorithme consiste en la mise en circulation de deux jetons ping et pong. Un site recevant l'un des jetons déclarera que l'autre jeton est perdu si le jeton reçu a fait un tour complet sans que le site en reçoive l'autre jeton entre temps. Dans le cas affirmatif, le site régénère le jeton perdu et le remet en circulation sur l'anneau.

1. En regardant les lignes L3 et L4 ainsi que les lignes L6 et L7 ; quelle est la valeur de Nbping + Nbpong ? Que peut-on conclure ?
2. Que comptent Nbping et |Nbpong| ?
3. Au niveau de la ligne L1, si Pi trouve que m = nbping il déclare que pong est perdu, expliquez pourquoi ?
4. Peut-on borner les variables Nbping et Nbpong ? Argumentez votre réponse.

5. Que ce passe t-il lorsque les liaisons ne sont pas fifo ?
6. Que ce passe t-il si les deux jetons se perdent ? Quelle solution proposeriez-vous pour remédier à ce problème ?
7. Expliquez comment cet algorithme est utilisé pour résoudre le problème de la section critique ?

L'algorithme :

Var m : entier initialisé à 0

Le comportement du processus Pi est le suivant :
Lors de

La réception de (ping, Nbping) faire
    Si m = Nbping **alors** L1
        Début <pong est perdu, il est régénéré> L2
            Nbping := Nbping + 1 ; L3
            Nbpong := **-** Nbping ; L4
        Fin
    Sinon m := nbping L5
    Fsi
Fait

La réception de (pong, Nbpong) faire
    <traitement analogue en intervertissant les rôles de ping et pong>
Fait

La rencontre des deux jetons faire
    Nbping := Nbping + 1 ; L6
    Nbpong := Nbpong – 1 ; L7
Fait

Corrigé :

Exercice 8
1. Quelles sont les propriétés comportementales des liaisons ?
2. Soit une liaison ne vérifiant pas une propriété Hi et soit un algorithme distribué nécessitant cette propriété pour son bon fonctionnement ; Quelle solution proposeriez vous pour l'utilisation de cet algorithme dans ces conditions ?

Corrigé :

Exercice 9
1. Donnez les avantages et les inconvénients des algorithmes à jeton résolvant le problème de la section critique dans un environnement distribué ?
2. Dans le même contexte, discutez les conditions pour lesquelles une topologie en anneau est avantageuse par rapport aux autres topologies ?
3. Quelles sont les topologies de réseaux que vous connaissez ? Quels sont les avantages et les inconvénients de chacune d'elles ?

Corrigé :

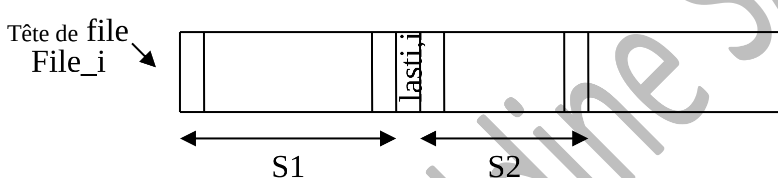
Exercice 10

Dans cet exercice on utilise l'algorithme de Carvalho et Roucairol pour gérer l'accès à la section critique de trois processus P1, P2 et P3 dont les horloges sont toutes égales à 0. On suppose que les ensembles R1, R2 et R3 sont comme suit : R1 = {}, R2 = {1} et R3 = {1,2}.

1. Donnez la répartition des permissions sur les différents sites ?
2. Supposons que le processus P2 appelle la procédure acquérir et arrive à rentrer et sortir de la section critique alors que les états des processus P1 et P3 demeurent égales à « dehors ». Quelles sont les valeurs des horloges H1, H2 et H3 ainsi que la nouvelle répartition des permissions dans l'état résultant ?
3. A partir de l'état résultant, nous supposons que le processus P1 appelle la procédure acquérir et arrive à rentrer et sortir de sa section critique alors que les états des processus P2 et P3 demeurent égales à « dehors ». Quelles sont les valeurs des horloges H1, H2 et H3 ainsi que la nouvelle répartition des permissions dans l'état résultant ?
4. Que peut-on conclure ?

Corrigé :

Exercice 11: Soit l'algorithme mixte vu dans le cours. Soit la configuration suivante d'un processus Pi.



1. Dans le cas où l'état du processus Pi est ***demandeur***. Que représentent les ensembles S1 et S2 ? Discutez l'état des permissions données par Pi ainsi que les valeurs des estampilles des éléments de S1 et de S2 vis-à-vis de celle de la requête de Pi ?
2. Même question dans le cas où l'état du processus Pi est ***dedans*** ?
3. Dans le cas où l'état du processus Pi est dehors. Dessinez la file ***File_i*** ?

Corrigé :

Exercice 12

Soit un système composé de N sites reliés par un anneau unidirectionnel. On suppose que les liaisons sont fiables. On se propose dans cet exercice d'écrire un algorithme résolvant le problème de la section critique. Ce dernier est basé sur l'existence d'un message particulier appelé **Jeton** qui est en mouvement circulaire perpétuel sur l'anneau.

1. Outre le message Jeton, quelles sont les variables (éventuellement une) utilisées par cet algorithme ?
2. Quelles sont les événements qui sont à l'initiative d'un processus ?
3. Quelles sont les événements externes qu'un processus peut subir ?
4. Pour chaque événement identifié précédemment, donnez la procédure à exécuter par le processus ?
5. L'exécution de chaque procédure doit remplir une propriété particulière, laquelle ?
6. Discuter les propriétés d'un tel algorithme en fonction de **N** et **T** (temps moyen de transit d'un message entre deux sites voisins).
7. Discutez les performances de l'algorithme en fonction de **N**.
8. Quels sont les problèmes engendrés par l'absence de la propriété de fiabilité des liaisons ? Sachant que nous n'avons pas le moyen pour rendre les liaisons fiables, quelle solution proposeriez-vous ?

quels problèmes sous-jacents peuvent apparaître et comment remédier à ces problèmes ? Que peut-on conclure ?

Corrigé :

Exercice 13

Soit la donnée de **n** sites, à chaque site **i** est associé un processus **Pi** qui peut à tout moment invoqué l'exécution d'une opération $\mathbf{op_j}$ parmi **m** opérations $\mathbf{op_1, op_2, ..., op_m}$. Le problème qui se pose est induit par la non compatibilité de certaines opérations. En d'autres termes deux opérations incompatibles invoquées chacune par un site doivent être exécutées en exclusion mutuelle.

Les relations d'exclusion entre ces opérations sont définies par une matrice ***Compat***, dite de compatibilité, booléenne et symétrique tel que :

Compat($\mathbf{op_i, op_j}$)$\Leftrightarrow$ $\mathbf{op_i}$ et $\mathbf{op_j}$ ne sont pas exclusives (elles sont compatibles)

Les messages de requêtes émis par un site **j** transportent, outre les estampilles, le nom de l'opération $\mathbf{op_k}$ qu'il veut exécuter. A la réception d'un tel message, un site **i** va retarder le renvoie de sa permission s'il a une opération en cours incompatible avec $\mathbf{op_k}$ et si son opération est dotée d'une estampille inférieure.

Le travail demandé consiste à adapter l'algorithme de Ricart et Agrawala pour la gestion de l'exécution de ces opérations par ces processus distribués. Pour cela chaque site **i** est doté, en plus des variables utilisées dans l'algorithme sus cité, de la variable locale **Last-opi** qui mémorise le nom de la dernière opération invoquée par ce site ($\mathbf{op_1, op_2, ..., op_m}$).

1. Rappelez les variables utilisées dans l'algorithme de Ricart et Agrawala en donnant le rôle de chacune d'elles.
2. Dans le cas de ce problème, quelle est la structure d'un message requête envoyé par un site **i** à un site **j**.
3. Donnez l'expression calculant la priorité d'un processus **i** recevant une requête de la part d'un processus **j**.
4. Donnez les algorithmes des énoncés régissant le comportement d'un site **i**, à savoir : <u>*Lors d'un appel à* $\mathbf{op_k}$</u> , <u>*lors de la terminaison de* $\mathbf{op_k}$</u>, <u>*lors de la réception d'une requête de la part d'un site* ***j***</u>, <u>*lors de la réception d'une permission de la part d'un site* ***j***</u>.

Corrigé :

Partie : Rendez-vous distribués

Exercice 1 :

Un protocole de mise en œuvre des rendez-vous distribués doit vérifier la propriété de coordination synchrone.

1. Donner la définition de cette propriété.
2. Malgré que les rendez-vous sont établis de manière asynchrone, dû à l'aspect distribué des processus intervenant dans les rendez-vous, expliquer comment la propriété de coordination synchrone préserve la sémantique de rendez-vous synchrone des langages de haut niveau, tel que LOTOS, dans lesquels les applications distribuées sont décrites.
3. Dans le protocole vu dans le cours, le problème de non bornétude des variables se pose. Quelle est la solution abordée dans le cours pour ce problème ? Expliquer la nécessité de la propriété de régularité des liaisons.

Solution :

Un protocole de mise en œuvre des rendez-vous distribués doit vérifier la propriété de coordination synchrone.

1. Donner la définition de cette propriété.

   Si un processus qui a invoqué un rendez-vous s'engage dans ce rendez-vous alors tous les processus qui appartiennent à ce rendez-vous l'ont invoqué également et s'engageront aussi dans ce rendez-vous.

2. Malgré que les rendez-vous sont établis de manière asynchrone, dû à l'aspect distribué des processus intervenant dans les rendez-vous, expliquer comment la propriété de coordination synchrone préserve la sémantique de rendez-vous synchrone des langages de haut niveau, tel que LOTOS, dans lesquels les applications distribuées sont décrites.

   Dans les langages de haut niveau, tel que LOTOS, la synchronisation de deux ou plusieurs processus se traduit sémantiquement par l'exécution simultanée d'une action par tous les processus. Si nous considérons que les processus sont distribués, la notion d'exécution simultanée n'a pas de sens (absence d'observateur global). De ce fait il suffit de préserver l'effet de l'exécution simultanée, qui n'est autre que l'exécution de la même action par tous les processus intervenant dans la synchronisation. Ce qui est assuré par la propriété de coordination synchrone qui exige qu'un rendez-vous ne peut avoir lieu qui si tous les processus qui y appartiennent arrivent au point de rendez-vous d'une part, et que si l'un d'entre eux s'engage dans un rendez-vous possible alors tous ces processus s'engageront également d'autre part.

3. Dans le protocole vu dans le cours, le problème de non bornétude des variables se pose. Quelle est la solution abordée dans le cours pour ce problème ? Expliquer la nécessité de la propriété de régularité des liaisons.

   - **La solution consiste à lancer un protocole de remise à zéro de toutes les variables tout en préservant les états de processus vis-à-vis de valeurs associées aux variables.**
   - **Admettant que les liaisons ne sont pas régulières. Supposons qu'un processus Pi non demandeur reçoit un message raz(CTLk), dans ce cas ni := 0 et wi := 0, et envoie à tous ses contrôleurs le message raz(CTLk, i , 0). Dans le cas où ce même processus devient demandeur, wi := wi +1 (donc 1) et envoie à ses contrôleurs le message prêt(i). Si ce message arrive avant le message raz, son effet sera annulé par le message raz et une incohérence apparaitra entre wi et ses images nbprêtj(i) (décalage de 1) ainsi qu'avec l'image nxi. De ce fait des rendez-vous possibles ne seront pas détectés, ce qui peut engendrer des blocages en avalanche, qui peut mener le système à une situation d'interblocage (absence de vivacité).**

<u>Exercice 2 :</u>

Soit l'expression Basic LOTOS suivante : (P1[a,b] ||| P2[c,d]) |[b,c]| P3[b,c,e] avec

- P1[a,b] = a ; b ; P1[a,b]
- P2[c,d] = d ; c ; P2[c,d]
- P3[b,c,e] = e ; (b ; P3[b,c,e] [] c ; P3[b,c,e])

En admettant que les processus P1, P2 et P3 sont respectivement implémentés sur les sites S1, S2 et S3,

1. Calculez les rendez-vous qui seront mis en œuvre.
2. Soient Ctr1 et Ctr2 deux contrôleurs qui gèrent les rendez-vous.

2.1. Proposez une distribution raisonnable du contrôle de rendez-vous sur les contrôleurs. Construisez en conséquence les ensembles Ctr-pari associés aux processus et les ensembles Proc-ctrj associés aux contrôleurs.
2.2. Pour ce cas d'étude, a-t-on besoin d'implémenter le choix indéterministe entre les rendez-vous conflictuels dans le protocole associé aux contrôleurs ? Justifiez votre réponse.

Solution :

1. On remarque qu'il existe deux rendez-vous à savoir r1 = {P1,P3} et r2 = {P2,P3}. En effet r1 correspond à la synchronisation sur la porte b alors que r2 correspond à synchronisation sur la porte c.
2.
   2.1. On propose la répartition suivante : {P1, P3} contrôlés par Ctr1 et {P2,P3} contrôlés par Ctr2. De ce fait Ctr-par1={Ctr1}, Ctr-par2={Ctr2} et Ctr-par3={Ctr1,Ctr2} alors que Proc-ctr1={P1,P3} et Proc-ctr2={P2,P3}.
   2.2. Non, pour ce cas d'étude on a pas besoin d'implémenter le choix indéterministe entre les rendez-vous conflictuels dans le protocole associé aux contrôleurs. En effet le conflit de rendez-vous concerne l'implication de P3 dans r1 et dans r2 ; cependant les éléments de r1 sont gérés par Ctr1 et les éléments de r2 sont gérés par Ctr2. Le choix indéterministe est assuré par l'arrivée aléatoire du jeton sur chacun des contrôleurs.

Exercice 3 :
Soit un système distribué composé de trois processus distribués P1, P2, P3 et de deux contrôleurs Ctr1 et Ctr2 tel que :
Ctr-par1 = {Ctr1}, Ctr-par2 = {Ctr1, Ctr2}, Ctr-par3 = {Ctr2}
1. Peut-on avoir les deux contrôleurs initiateurs en même temps du protocole de remise à zéro des variables de contrôle ? Justifiez votre réponse ?
2. Est-il opportun de ne remettre à zéro que les variables de contrôle du processus dont les valeurs ont atteint la borne maximale ? Justifiez votre point de vue.
3. Peut-il existé un rendez-vous impliquant les trois processus ? Argumentez votre réponse par l'analyse des images des variables wi dans les contrôleurs.

Solution :

1. Les contrôleurs étant mis sur un anneau et le contrôle est attribué à un contrôleur sous condition qu'il dispose du jeton, alors l'unicité du jeton assure que le lancement du protocole de remise à zéro des variables ne peut se faire que par un seul contrôleur.
2. Non il n'est pas opportun de ne remettre à zéro que les variables de contrôle du processus dont les valeurs ont atteint la borne maximale. En effet dans le cas général, tous les processus coopératifs de l'application sont impliqués dans des rendez-vous qui les lient selon cette application, de ce fait les variables de contrôle progressent tous, même avec des rythmes différents. De cela nous concluons que si la valeur d'une variable atteigne la valeur maximale, les valeurs des autres variables ne tarderont pas à atteigne cette valeur, il est donc plus judicieux de faire leur mise à jour ensemble.
3. Non, il ne peut exister un tel rendez-vous car aucun des deux contrôleurs ne contrôle les rendez-vous des trois processus en même temps.

Exercice 4
Soit la donnée de trois processus P1, P2 et P3 se trouvant respectivement sur les sites S1, S2 et S3. Les codes de ces processus sont comme suit :
Processus P1 : ... Rendez-vous (r1 ; writeln(''l'union'') ou r2 ; writeln(''plus c'est tôt'')) ...
Processus P2 : ... Rendez-vous (r1 ; writeln(''fait'') ou r2 ; writeln(''mieux ça vaut'')

ou r3 ; writln(‘’mieux vaut tard’’) ...

Processus P3 : ... Rendez-vous (r1 ; writeln(‘’la force’’) ou r3 ; writeln(‘’que jamais’’)) ...

Le résultat de ce calcul distribué consiste à juxtaposer les messages affichés sur chacune des consoles des sites S1, S2 et S3 dans cet ordre.

1. Quel est le résultat de ce calcul ? Commentez votre réponse.
2. L’indéterminisme est-il à priori ou à postériori ?

Exercice 5

Soit un système distribué composé de trois processus distribués P1, P2, P3 et de deux contrôleurs Ctr1 et Ctr2 tel que :

Ctr-par1 = {Ctr1}, Ctr-par2 = {Ctr1, Ctr2}, Ctr-par3 = {Ctr2}

1. Peut-il existé un rendez-vous impliquant les trois processus ? Argumentez votre réponse.
2. Peut-on avoir les deux contrôleurs initiateurs en même temps du protocole de remise à zéro des variables de contrôle ? Justifiez votre réponse ?
3. Est-il opportun de ne remettre à zéro que les variables de contrôle du processus dont les valeurs ont atteint la borne maximale ? Justifiez votre point de vue.

Exercice 6 : Dans la littérature plusieurs langages de programmation sont dédiés aux applications distribuées. De manière simplifiée, l’expression P1 |[a]| P2 représente deux processus P1 et P2 se trouvant respectivement sur deux sites S1 et S2. Les deux processus évoluent indépendamment l’un de l’autre sauf lors de l’exécution de l’action a, dans ce cas les deux processus doivent attendre à ce point de rendez-vous, exécutent simultanément l’action **a** et continue chacun son exécution avec la même sémantique.

1. Dans la pratique une telle sémantique n’est pas implantable, quelles sont les raisons ?
2. Quelle est la propriété la plus réaliste à vérifier afin qu’une implantation de cette sémantique soit acceptable ? Expliquez pourquoi ?

Exercice 7

Dans le protocole de remise à zéro des variables du protocole de mise en œuvre des rendez vous distribués, la propriété de fiabilité des liaisons est requise.

1. Quel problème peut apparaître lors que cette propriété n’est pas vérifiée ?
2. Quelle solution proposeriez-vous ?

Exercice 8

Soit le protocole de mise en œuvre des rendez-vous distribués vu dans le cours.

1. Rappeler la procédure exécutée par un contrôleur suite à la réception du jeton.
2. Montrer que l’exécution de cette procédure assure les opérations suivantes :
    a. Pour chaque ensemble de rendez-vous conflictuels gérés par ce contrôleur, la procédure activera un et un seul rendez-vous parmi les rendez-vous possibles.
    b. Plusieurs rendez-vous peuvent être activés durant l’exécution de la procédure.
    c. La propriété de choix aléatoire de rendez-vous parmi des rendez-vous conflictuels possibles n’est pas assurée.
    d. Modifier la procédure pour assurer cette propriété.

Exercice 9

Soit la donnée de trois processus P1, P2 et P3 se trouvant respectivement sur les sites S1, S2 et S3. Les codes de ces processus sont comme suit :

Processus P1 : ... Rendez-vous (r1 ; writeln(‘’text 1’’) ou r2 ; writeln(‘’text 2’’)) ...

Processus P2 : ... Rendez-vous (r1 ; writeln(‘’text 3’’) ou r2 ; writeln(‘’text 4’’) ou r3 ; writln(‘’text 5’’) ...

Processus P3 : ... Rendez-vous (r1 ; writeln(‘’text 6’’) ou r3 ; writeln(‘’text 7’’)) ...

Le résultat de ce calcul distribué consiste à juxtaposer les messages affichés sur chacune des consoles des sites S1, S2 et S3 dans cet ordre.

1. Quel est le résultat de ce calcul ? Commentez votre réponse.
2. L’indéterminisme est-il à priori ou à postériori ?

<u>Exercice 10</u>

1. Etant donné un algorithme de mise en œuvre de rendez-vous distribués, rappelez :
2. La propriété de coordination synchrone.
3. La propriété d’exclusion.
4. La propriété de vivacité.
5. L’invariant exprimant la propriété de coordination synchrone en fonction des variables abstraites.
6. L’invariant exprimant la propriété d’exclusion en fonction des variables abstraites.
7. L’invariant exprimant la propriété d’exclusion en fonction des variables des processus et du jeton.

<u>Exercice 11</u>

Soit un système distribué composé de trois processus distribués P1, P2, P3 et de deux contrôleurs Ctr1 et Ctr2 tel que :

Ctr-par1 = {Ctr1}, Ctr-par2 = {Ctr1, Ctr2}, Ctr-par3 = {Ctr2}

Montrez par l’absurde qu’il ne peut exister de rendez-vous impliquant les trois processus en même temps ?

Partie : Etat global

<u>Exercice 1</u>:

L’utilisation du temps virtuel nécessite l’assurance de la contrainte de cohérence.

1. Donner la définition de cette propriété.
2. L’assurance de cette propriété peut se réaliser selon deux approches, lesquelles ? Expliquer le principe de chacune d’elles.
3. Laquelle des deux approches nécessite le calcul du temps virtuel global (GVT) ? Expliquer pourquoi ? Quelles sont les propriétés que doit avoir tout protocole calculant le temps virtuel global ?

<u>Solution</u> :

L’utilisation du temps virtuel nécessite l’assurance de la contrainte de cohérence.

1. Donner la définition de cette propriété.

   Les messages délivrés à un site le sont dans l’ordre non-décroissant de leurs estampilles.

2. L’assurance de cette propriété peut se réaliser selon deux approches, lesquelles ? Expliquer le principe de chacune d’elles.

   - **Approche pessimiste : Dans cette approche on agit en amont. En effet avant qu’un message soit délivré à un site, le protocole assure l’impossibilité qu’un message d’estampille inférieur sera reçu par la suite par le processus.**

- **Approche optimiste : Dans cette approche on suppose que les messages arriveront à destination dans l'ordre de leurs estampilles. Ce n'est que lorsqu'une incohérence sera détectée qu'une annulation des opérations touchées par l'incohérence sera faite. Evidement une reprise des traitements associés sera lancée.**

3. Laquelle des deux approches nécessite le calcul du temps virtuel global (GVT) ? Expliquer pourquoi ? Quelles sont les propriétés que doit avoir tout protocole calculant le temps virtuel global ?

- ***Il est évident, d'après l'explication du principe de chaque approche dans la question précédente, que l'approche optimiste nécessite le calcul du temps virtuel global (GVT) afin de défaire et de refaire des opérations de calcul associées aux livraisons incohérentes de messages.***
- ***Pour un protocole calculant le temps virtuel global (GVT), il doit vérifier les deux propriétés suivantes :***
  - ***Propriété de sûreté : La livraison de tout message d'estampille inférieur à GTV est cohérente. Autrement dit, toute incohérence ne concernera pas les messages d'estampilles inférieures à GVT.***
  - ***Propriété de vivacité : Lorsque le protocole de calcul du temps virtuel global (GVT) est lancé, il finira par trouver la valeur de GVT au bout d'un temps fini.***

<u>Exercice 2 :</u>
Les protocoles basés sur le temps virtuel global doivent assurer la contrainte de cohérence.

1. Rappelez la contrainte de cohérence.
2. Soient 3 processus P1, P2 et P3. On suppose que P1 a envoyé un message (m1,d1) à P3 et que P2 a envoyé deux messages (m2,d2) et (m3,d3) à P3. On suppose aussi que d2 < d1< d3.
   a. Dessinez un schéma descriptif de ce scénario.
   b. Montrez que le respect de la contrainte de cohérence peut engendrer un interblocage.

<u>Solution</u> :

Partie : Problème d'élection

<u>Exercice 1 :</u>
Soit l'algorithme d'élection par déclaration donné dans la page suivante :
Hypothèses :
- Anneau virtuel fiable.
- Chaque processus a un numéro.
- Tous les numéros sont différents.

On veut élire le processus de plus grand numéro. Donc tous les processus de l'anneau participent à la compétition.
N'importe quel processus (éventuellement plusieurs processus) lance l'exécution de l'algorithme.

Remarque : On suppose qu'il y a un processus par site.

1) Etant donné un algorithme d'élection ?
   a. Quelles sont les propriétés assurant la validité d'un tel algorithme ?

b. Décomposez ces propriétés selon les deux classes classiques à savoir la classe des propriétés de sûreté et la classe des propriétés de vivacité ?

2) Commentez les instructions L1 jusqu'à L10 (Chaque commentaire ne doit pas dépasser une ligne) ?
3) Peut-on remplacer l'instruction L13 par « Si num < > monnum alors » ? Justifiez votre réponse ?
4) Dans le cas d'un seul initiateur
   a. En prenant comme exemple un anneau virtuel de taille 4. Quel est le nombre d'émissions de messages de type «(élection,num) » durant l'opération d'élection ?
   b. Généralisez le résultat à un anneau virtuel de taille n (n > 0) ?
5) Dans le cas où tous les sites seraient des initiateurs en même temps avec l'hypothèse que les liaisons sont de type fifo (les messages ne se dépassent jamais)
   a. Quelle est la configuration de l'anneau qui conduit à un nombre maximal d'émissions de messages de type « (élection,num) » ? Quel est ce nombre ?
   b. Quelle est la configuration de l'anneau qui conduit à un nombre minimal d'émissions de messages de type « (élection,num) » ? Quel est ce nombre ?
   c. Etant donnée une configuration quelconque de l'anneau, quel est le nombre d'émissions de messages de type « (élection,num) » mis en œuvre lors du processus d'élection ?
   d. Quel est l'impact de l'hypothèse précédente (liaison fifo) sur le nombre de messages de type « (élection,num) » ?
6) Dans cette question on vous demande de :
   a. Montrez que s'il y a un numéro élu, c'est le maximum ?
   b. Montrez que le maximum sera élu (La réponse à cette question peut être divisée en deux étapes, dans la première étape on montre que le maximum sera émis, dans la deuxième étape on montre que le maximum fera le tour de l'anneau) ?
   c. Concluez ?
   d. Discutez la validité de l'algorithme étudié ?

L'algorithme :

```
Pour chaque processus Pi
        Var     monnum : entier := « valeur satisfaisant les hypothèses »
                Candidat : booléen := faux ;
                Max : entier ;
Lancer l'élection
        Candidat := vrai ;
        Emettre (élection, monnum) ;
Réception de (élection, num)
    •   Cas     num > monnum                        L1
                Emettre (élection, num) ;                   L2
                Candidat := vrai ;                          L3
    •   Cas num < monnum                                    L4
                Si non candidat alors                       L5
                        Emettre (élection, monnum) ;  L6
                        Candidat := vrai ;                  L7
                Sinon
                        Ne rien faire ;                     L8
                Finsi
    •   Cas num = monnum                                    L9
                Emettre (élu, monnum) ;                     L10
                Candidat := faux ;                          L11
```

```
                Max := monnum ;                          L12
Réception de (élu,num)
        Si Candidat alors                                L13
                Max := num ;                             L14
                Candidat := faux ;                       L15
                Emettre(élu, num);                       L16
        Sinon
                Ne rien faire ;                          L17
        Finsi
```

Solution :

1.

a. Propriétés assurant la validité de l'algorithme

P1 : Unicité de l'élu.

P2 : S'il y a des candidats pour l'élection, il y aura sûrement un élu.

b. Propriété de sûreté : Quelque chose de mauvais n'arrivera jamais.

P1.

Propriété de vivacité : Quelque chose de bon arrivera sûrement

P2 : Absence d'interblocage

2. L1 : Je suis perdant, num est peut être gagnant.
L2 : Laisser num continuer sur l'anneau.
L3 : Je suis candidat
L4 : num est perdant, je suis peut être gagnant
L5 : Je n'étais pas encore candidat
L6 : Envoyer mon numéro sur l'anneau
L7 : Je deviens candidat
L8 : J'étais déjà candidat, mon numéro a déjà été émis
L9 : Je suis vainqueur
L10 : Je le signale aux autres processus

3. Oui on peut remplacer « Si candidat alors » par « Si num <> monnum alors »
Justification : Le message « (élu,num) » est envoyé après que tous les sites sont devenus candidats (spontanément ou par provocation). Le vainqueur a positionné sa variable « candidat » à faux et c'est le critère d'arrêt, étant donnée que c'est son numéro qui circule, « num = monnum » constitue également un critère d'arrêt.

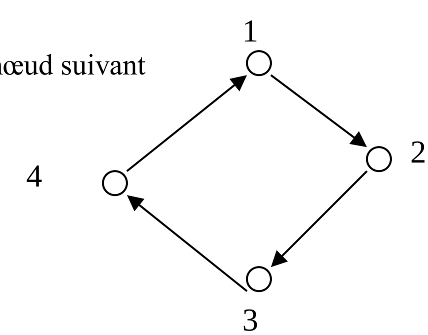
4. Hypothèse : Un seul initiateur

a. Anneau de taille 4 :

Premier cas : Initiateur = site 4 (futur élu)

Chaque nœud recevant le message le transmet vers le nœud suivant

➔ 4 émissions du méssage d'élection.



Deuxième cas : Initiateur = site 1. Même raisonnement

➔ 3 + 4 = 7 émissions

3 = distance séparant l'initiateur au futur élu.

Troisième cas : Initiateur = site 2.

➔ 2 + 4 = 6 émissions

2 = distance séparant l'initiateur au futur élu.

Quatrième cas : Initiateur = site 3.

➔ 1 + 4 = 5 émissions

1 = distance séparant l'initiateur au futur élu.

b. Généralisation

Soit I le site initiateur.

D étant la distance séparant I du futur initiateur.

Nombre d'émissions = n + D

Entre autre

n <= nombre d'émissions <= 2 n – 1

5. Hypothèses : Tous les sites sont des initiateurs en même temps.
Les messages ne se dépassent pas.

a. C'est la configuration dans laquelle les suivants du futur élu (nœud ayant l'identificateur le plus élevé) sont ordonnés par ordre décroissant.

Nombre d'émissions = $\sum^{n}_{i=1} i = n*(n+1) / 2$

b. C'est la configuration dans laquelle les suivants du futur élu sont ordonnés par ordre croissant.

Nombre d'émissions = n + (n – 1) = 2 n – 1

c. Pour une configuration quelconque

2 n – 1 <= nombre d'émissions <= n*(n+1) / 2

d. Pour une configuration quelconque, la considération de liaisons non fifo entraîne une diminution du nombre d'émissions de messages de type « (élection,num) ».

6.

a. S'il y a un élu, c'est le max ?

L'algorithme ➔

- Est élu tout numéro ayant fait le tour de l'anneau, et seuls ces numéros sont élus.
- Un numéro ayant fait le tour a été ré-émis par chaque site ➔il est le plus grand que tous les numéros de tous les autres sites ➔ c'est le max.

Hypothèse ➔ chaque site a un numéro différent ➔ le max est unique

Donc s'il y a un numéro élu c'est le max.

b. Le max sera élu ?

α) Le max sera t-il émis ?

- Ou bien il sera émis spontanément.
- Ou bien son émission est provoquée.

Hypothèse : Un processus a lancé l'élection ➔ un numéro circule sur l'anneau ➔ ce numéro rencontrera le site du max ➔ le max sera émis.

β) Le maximum fera t-il le tour de l'anneau ?

Le maximum ne peut être éliminé par aucun processus qui n'est pas son émetteur ➔ il fera le tour de l'anneau.

Le maximum sera donc élu.

c. Si un ou plusieurs processus quelconques lance(nt) l'élection, alors le numéro max sera élu, et sera le seul élu.

(c) ➔ P1 et P2.

<u>Exercice 2 :</u>

Dans ce qui suit, nous proposons l'étude de problèmes liés aux réseaux dont la topologie est un anneau asynchrone anonyme.

L'asynchronisme du réseau signifie que les délais de communication sont finis mais indéterminés. Alors que le réseau est dit anonyme si les sites qui le composent sont structurellement identiques et ne sont pas forcément distinguables deux à deux. Il est à noter qu'à chaque site est associée une valeur d'entrée.

Première partie : Compréhension du problème

Soit R = {r_0, … , r_{n-1}}un anneau asynchrone anonyme composé de n sites. Initialement, à chaque site ri est associée une valeur **Si ∈ V**, tel que **V** est un ensemble totalement ordonné.

Les valeurs d'entrée **Si** (i ∈ {0, … , n –1}) ne sont pas nécessairement distinctes. Ces valeurs forment un super-ensemble (multi-set) **S = {S0, S1, … , Sn-1}**.

Soit **Δ(S)** la cardinalité de **S** exprimant le nombre de valeurs différentes dans **S**.

Q1) (0,5pts) Quelle est la relation reliant **Δ(S)** et **n** ?

Q2) (2pts) Dans le cas où **Δ(S) = n** :

1. Quelle est la nature de **S** ?
2. Peut-on considérer que l'anneau est anonyme ? Justifiez votre réponse ?
3. Sur un tel anneau, proposez un critère pour l'élection d'un site parmi tous les sites qui composent l'anneau ?

Q3) (1,5pts) **Dans le cas où** Δ(S) < n **:**

1. Quelle est la nature de S ?
2. Peut-on considérer que l'anneau est anonyme ?

L'un des problèmes à résoudre sur un anneau asynchrone anonyme consiste à trier les valeurs d'entrée des sites tel que à la fin de l'opération de tri, il existe deux sites adjacents dont l'un a la plus petite valeur et l'autre a la plus grande valeur et le parcoure de l'anneau du site de plus petite valeur vers le site de plus grande valeur peut se faire en passant par tous les autres sites tout en ne rencontrant que des valeurs croissantes.

Remarque : Il est à noter que les sites sont physiquement figés, l'opération de tri ne fait que changer l'affectation des valeurs d'entrée aux différents sites.

Q4) (2pts) Soit l'anneau asynchrone anonyme de la Figure 1. Donnez le résultat du tri tel que r0 a la plus grande valeur et le parcoure de l'anneau dans le sens inverse des aiguilles d'une montre passe par des valeurs décroissantes jusqu'à la rencontre d'un site dont la valeur est égale à celle de r0?

Q5) (1pt) Soit D_u(S) la multiplicité de u dans S. Sur l'exemple de la Figure 1, quelles sont les valeurs de D_u(S) ?

Deuxième partie : Problème d'élection sur un anneau asynchrone anonyme

Soit S **un anneau asynchrone anonyme binaire bi-directionnel (** Δ(S) = 2, V={0,1}**).**

Idée de l'algorithme : Assigner dynamiquement à chaque site actif une valeur et de réduire étape par étape le nombre de sites actifs par la comparaison de la valeur de tout site actif avec les valeurs des sites voisins. Jusqu'à l'état dans lequel il ne reste qu'un seul site actif qui se déclare élu et informe les autres sites qu'ils sont perdants.

Initialement, tous les sites sont actifs. Soit **P** un site dont **b** est sa valeur d'entrée. Si **P** est initiateur de l'opération d'élection alors il envoie sa valeur au site de droite et se met dans l'état ***SeenOnlyEqual***. P reste dans cet état aussi longtemps qu'il reçoit sur sa gauche des valeurs égales à **b**. Le nombre de valeurs **b** reçues déterminera éventuellement sa nouvelle valeur, qui sera utilisée dans l'état suivant. Plusieurs cas peuvent se présenter.

(Cas 1) : **P** reçoit sur sa gauche **n-1** valeurs, toutes égales à **b**. Dans ce cas il bascule dans l'état ***AllEqual***.
(Cas 2) : **P** envoie des valeurs à droite et reçoit des valeurs à gauche jusqu'à ce qu'il reçoive une valeur différente de **b**. Dans ce cas il choisit comme nouvelle valeur **v** le nombre de **b** qu'il a collecté sur sa gauche plus sa propre valeur ; envoie cette valeur à droite et à gauche dans un message **<*SOE,v*>** et se met dans l'état ***Electing***.
(Cas 3) : **P** reçoit un message **<*SOE,z*>** du site voisin. Il sauvegarde la nouvelle valeur **z** de son voisin. Dans ce cas, si le message provient du voisin de gauche il choisit sa nouvelle valeur égale à **z+1** et se met dans l'état ***Electing*** (ce cas est équivalent à la réception, de son voisin de gauche, d'une valeur différente de **b**).
Intuitivement, dans l'état d'élection, le nombre de sites actifs diminue. En effet, un site **P** reçoit en premier les messages **<*SOE,x*>** et **<*SOE,y*>** contenant les nouvelles valeurs **x** et **y** de ces voisins actifs de gauche et de droite (à moins que sa soit déjà fait dans l'état précédent). Dans ce cas il compare sa propre valeur **v** avec **x** et **y**. Si **v** et tels que **(x<=v et y<v) ou (x<v et y<=v)** alors **P** reste ***actif***, dans le cas contraire il bascule dans l'état ***Passive***.
Si **P** reste actif, il envoie un compteur (initialisé à 1) au site de droite et bascule dans l'état ***Counting***. Durant cet état les autres sites actifs mettent à jour leur valeur comme suit :

- Un site passif qui reçoit le compteur, l'incrémente de 1 et l'envoie à son site de droite.
- Un site actif **P** qui reçoit un compteur **d** du site de gauche vérifie si **d = n**. Dans ce cas **P** sait que c'est son propre compteur. Il devient donc gagnant et bascule à l'état ***Elected***. Dans le cas contraire, **d < n**, **P** choisit **d** comme étant sa nouvelle valeur et envoie le message **<*SOE,d*>** aux sites de gauche et de droite et revient à nouveau à l'état ***Electing***.

Comme le nombre de sites actifs décroît à chaque itération des états ***Electing*** et ***Counting***, à un certain point il y'aurait un gagnant unique qui basculera à l'état ***Elected***. L'élu envoie sa valeur aux autres sites dans un message ***Election*** et se met dans l'état final ***Leader***. A la réception de ce message, tous les autres sites basculent de l'état ***Passive*** à l'état ***Defeated***.

Q6) (1,5pts) Que peut-on conclure dans le cas 1 ?
Q7) (1,5pts) Dans le cas 3, pour quelle raison, lorsque P reçoit un compteur égal à n se déclare élu ?
Q8) (4pts) La Figure 2 représente l'automate décrivant les états et les transitions d'un site de l'anneau. En s'appuyant sur la description de l'algorithme ci dessus, complétez l'automate de la Figure 2 par l'ajout à chaque transition de la condition qu'il faut. A titre d'exemple, le passage de l'état ***Passive*** à l'état ***Defeated*** est conditionné par la réception du message ***Election*** (Voir Figure 2).

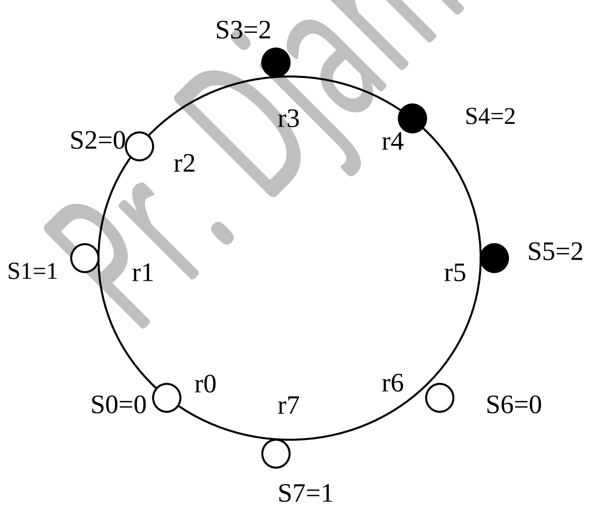
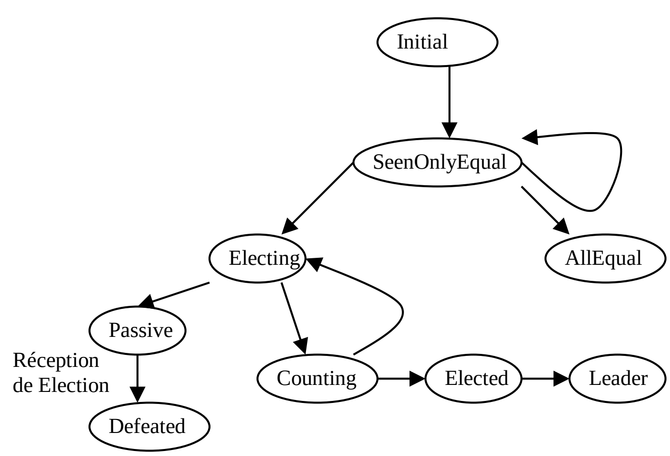


Figure 1



Figure

Solution :

Première partie : Compréhension du problèm

Q1) Quelle est la relation reliant Δ(S) et n ?

Rep : **Δ(S) <= n**

Q2) Dans le cas où Δ(S) = n :

4. Quelle est la nature de S ? **Rep : S est un ensemble**.
5. Peut-on considérer que l'anneau est anonyme ? **Rep : L'anneau n'est pas anonyme** Justifiez votre réponse ? **Rep : Parce que tous les sites distinguables par les valeurs qui leur sont associées.**
6. Sur un tel anneau, proposez un critère pour l'élection d'un site parmi tous les sites qui composent l'anneau ? **Rep : Un critère d'élection pourrait consister à élire le site dont la valeur est maximale ou bien minimale.**

Q3) **Dans le cas où Δ(S) < n :**

3. Quelle est la nature de S ? **Rep : S n'est pas un ensemble care il existe des valeurs redandantes.**
4. Peut-on considérer que l'anneau est anonyme ? **Rep : Oui l'anneau est anonyme.**

Q4) Soit l'anneau asynchrone anonyme de la Figure 1. Donnez le résultat du tri tel que r0 a la plus grande valeur et le parcoure de l'anneau dans le sens inverse des aiguilles d'une montre passe par des valeurs décroissantes jusqu'à la rencontre d'un site dont la valeur est égale à celle de r0?

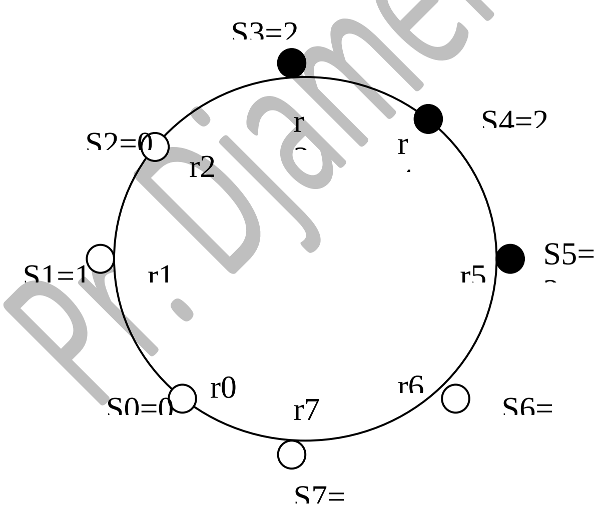


Figure 1

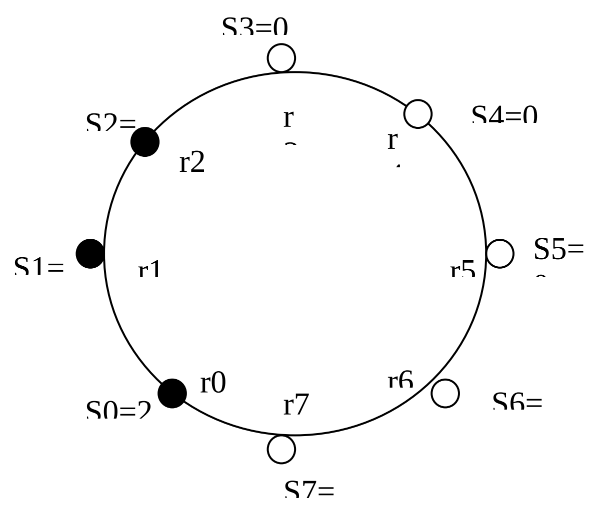


Figure 1'

Q5) Soit $D_u(S)$ la multiplicité de u dans S. Sur l'exemple de la Figure 1, quelles sont les valeurs de $D_u(S)$?

Rep : D0(S) = 3

D1(S) = 2

D2(S) = 3

Deuxième partie : Problème d'élection sur un anneau asynchrone anonyme

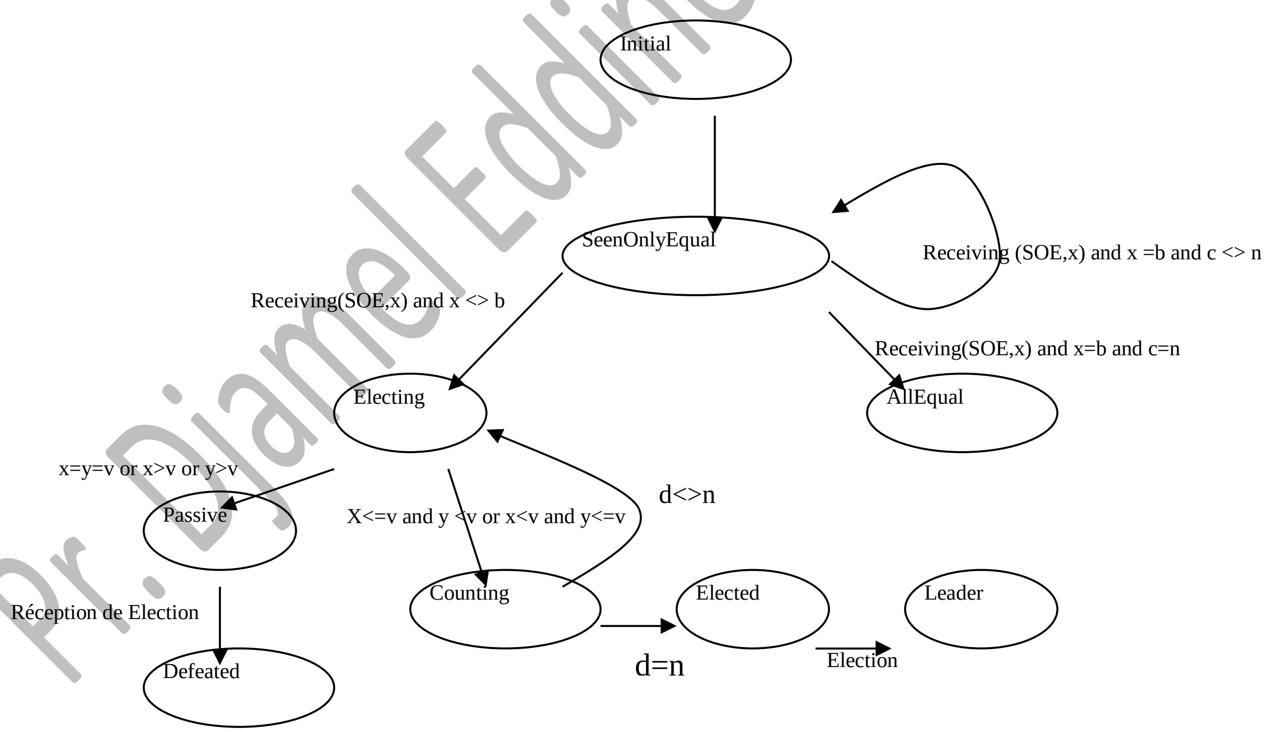
Q6) Que peut-on conclure dans le cas 1 ?

Rep : Aucun site ne peut être élu care tous les sites ont les mêmes valeurs d'entrée.

Q7) Dans le cas 3, pour quelle raison, lorsque P reçoit un compteur égal à n se déclare élu ?

Rep : Care, d'une part il déduit que c'est sont propre compteur et d'autre part les sites rencontrés sont tous dans l'état Passive.

Q8) La Figure 2 représente l'automate décrivant les états et les transitions d'un site P de l'anneau de valeur d'entrée b. En s'appuyant sur la description de l'algorithme ci dessus, complétez l'automate de la Figure 2 par l'ajout à chaque transition de la condition qu'il faut. A titre d'exemple, le passage de l'état Passive à l'état Defeated est conditionné par la réception du message Election (Voir Figure 2).



X est la valeur du voisin de gauche
Y est la valeur du voisin de droite
V est la valeur courante de P
d est la valeur du compteur