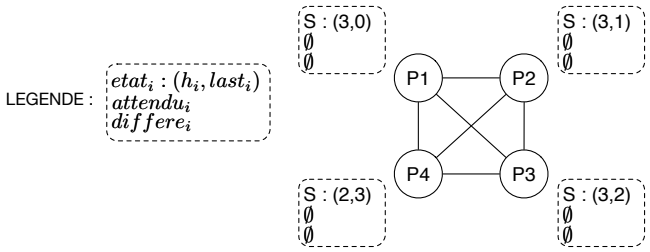
Université de Versailles – Saint-Quentin-en-Yvelines
Algorithmique Distribuée

TD 3 - Exclusion Mutuelle

Exercice 1. Algorithme basé sur les permissions de Ricart-Agrawala

On considère l'algorithme d'exclusion mutuelle de Ricart-Agrawala et la configuration initiale C_0 suivante :



On dit que deux sites P et P' demandent "simultanément" à entrer en SC si, au cours d'une configuration, les sites P et P' sont tous les deux dans l'état de demande de section critique.

1. Donner les grandes lignes d'une exécution qui a pu aboutir à la configuration C_0 et dans laquelle il n'y a pas de demandes simultanées et chaque site fait au plus une demande de SC.
2. Donner les grandes lignes d'une exécution qui a pu aboutir à la configuration C_0 et dans laquelle il n'y a pas de demandes simultanées et P_2 fait deux demandes de SC.
3. En déduire combien de demandes de SC ont fait P_1, P_2, P_3 et P_4 pour arriver à cette configuration ?
4. A partir de C_0, on suppose que les sites P_1 et P_3 demandent en même temps à entrer en section critique.
   (a) Dire qui de P_1 ou de P_3 entre en premier en section critique. Justifier.
   (b) Donner la valeur des variables h_i et $last_i$ pour tous les sites, après que P_1 et P_3 soient sortis de section critique.
   (c) Indiquer combien de messages ont été échangés dans cette exécution.
5. A partir de C_0, comment faire pour qu'un site demande son entrée en section critique, mais voit sa demande différée à cause de son estampille.

Exercice 2. Algorithme basé sur les permissions de Carvalho-Roucairol

L'objectif de cet exercice est d'écrire une version améliorée de l'algorithme de Ricart-Agrawala, version qui améliore la complexité en messages. Le principe de ce nouvel algorithme est donné ci-dessous.

On considère que, pour chaque paire de sites (i, j), il existe un unique jeton nommé $Perm(i, j)$. Un site i ne peut entrer en SC que s'il possède pour tout j le jeton $Perm(i, j)$. Lorsqu'un site veut entrer en SC, il fait donc des requêtes pour tous les jetons qui lui manquent aux sites concernés, et uniquement pour ceux-ci. Lorsqu'un site i reçoit d'un site j la demande de permission $Perm(i, j)$, il l'envoie uniquement s'il calcule qu'il n'est pas prioritaire ; sinon, il enverra le jeton après qu'il soit entré en SC. Le calcul des priorités se fait à l'aide d'estampillages de manière analogue à l'algorithme précédent.

En pratique, on n'utilise pas explicitement ces jetons ; si le site i n'a pas $Perm(i, j)$, on traduit cela sur le site i par $j \in attendu_i$.

Il y a une partie délicate dans cette version : lorsqu'un site i est en train d'*attendre* les permissions qu'il a demandées et qu'il reçoit alors d'un site j une demande *prioritaire* de permission, alors i doit non seulement envoyer sa permission à j, mais il doit en plus faire une demande de permission à j (envoi de deux messages : $Perm$ puis Dem à j). En effet, au moment où i envoie sa permission à j, il *détient* la permission de j, il n'a donc aucune demande de permission à j en cours. Le site i doit donc indiquer à j de lui redonner la permission une fois sorti de SC, afin qu'il puisse lui-même entrer en SC après j.

L'algorithme de Ricart-Agrawala est donné ci-dessous. Insérer les modifications nécessaires pour mettre en place la version améliorée.

——— **Variables et initialisations** ———

...

——— **Algorithme** ———

Sur demande d'entrée en section critique →
$etat_i \longleftarrow E$
$last_i \longleftarrow h_i + 1$
$attendu_i \longleftarrow$ tous les sites sauf i
for all $j \in attendu_i$ **do**
Envoyer $Dem(last_i, i)$ à j

Sur réception de $Perm$ **de** j **→**
$attendu_i \longleftarrow attendu_i \setminus \{j\}$
if $attendu_i = \emptyset$ **then**
$etat_i \longleftarrow SC$

Sur réception de $Dem(h', j)$ **de** j **→**
$h_i \longleftarrow \max(h_i, h')$
$priorite_i \longleftarrow (etat_i = SC) \vee [(etat_i = E) \wedge (last_i, i) < (h', j)]$
if $priorite_i$ **then**
$differe_i \longleftarrow differe_i \cup \{j\}$
else
Envoyer $Perm$ à j
...

Sur sortie de section critique →
$etat_i \longleftarrow S$
for all $j \in differe_i$ **do**
Envoyer $Perm$ à j
...
$differe_i \longleftarrow \emptyset$

Exercice 3. Algorithme basé sur la circulation d'un jeton de Ricart-Agrawala

1. On considère un graphe complet de 5 sites : $P1, P2, P3, P4$ et $P5$.
   (a) Donner l'exécution de l'algorithme dans le scénario suivant : il y a une seule demande d'entrée en SC et celle-ci est faite par le site $P1$.
   (b) Donner un scénario dans lequel le site $P4$ est le deuxième à entrer en SC, alors que c'est le premier à en faire la demande.
2. Quel est le temps d'attente maximal entre 2 entrées successives en SC d'un même site ? Quel est le temps d'attente minimal ?
3. Illustrer au travers d'une exécution la raison pour laquelle un site parcours son tableau et celui du jeton de l'indice $i + 1$ jusqu'à n, puis de l'indice 1 à $i - 1$ et non de l'indice 1 à n, privé de l'indice i.