C34211

3è ANNÉE

INFORMATIQUE I

DURÉE : 5 HEURES

Aucun document n'est autorisé

L'usage de toute calculatrice est interdit.

On considère le problème de l'accès de n processus à une unique ressource non partageable. La ressource ne peut être accédée que par un processus à la fois : elle est dite *critique*, et son utilisation se fait au sein de ce que l'on appelle une *section critique*. On souhaite donc définir des *protocoles d'exclusion mutuelle*, garantissant cette propriété d'accès unique.

I Définitions

Processus. On considère un ensemble fini $\mathcal{P}$ de $n > 0$ processus : $\mathcal{P} = \{p_0, p_1, \ldots, p_{n-1}\}$ s'exécutant simultanément.

États. L'ensemble X_i des états possibles du processus p_i est partitionné en

$$X_i = R_i \cup E_i \cup C_i \cup S_i$$

Ces ensembles deux à deux disjoints seront appelés respectivement section *restante*, section *d'essai*, section *critique* et section *de sortie*.

Tournez la page S.V.P.

Variables. $\mathcal{V}$ est un ensemble fini de $m > 0$ variables, partagées entre les processus. La j^e variable de $\mathcal{V}$ a ses valeurs dans l'ensemble V_j.

Configurations. Une *configuration* est un $(n+m)$-uplet

$$q = (x_0, \ldots, x_{n-1}, v_0, \ldots, v_{m-1})$$

où $x_i \in X_i$ pour $0 \leq i < n$ et $v_j \in V_j$ pour $0 \leq j < m$. Une configuration contient donc les états respectifs de tous les processus et les valeurs de toutes les variables à un instant donné. On notera X_i, V_j et V les projections définies par

$$X_i(q) = x_i, \quad V_j(q) = v_j \quad \text{et} \quad V(q) = (v_0, \ldots, v_{m-1})$$

Système. Un *système* est un quadruplet $S = (\mathcal{P}, \mathcal{V}, q_0, \varphi)$ où
- $\mathcal{P}$ est l'ensemble des processus,
- $\mathcal{V}$ est l'ensemble des variables,
- q_0 est la configuration initiale de S,
- φ est la fonction de transition de S, définie ci-dessous.

Transitions. Soit Q l'ensemble de toutes les configurations de S. Alors la fonction de transition φ est une fonction totale de $\{0, \ldots, n-1\} \times Q$ dans Q telle que, si $i \in \{0, \ldots, n-1\}$ et $q \in Q$, alors $q' = \varphi(i, q)$ vérifie
1. $X_j(q') = X_j(q)$ pour $j \in \{0, \ldots, n-1\} \backslash \{i\}$, et,
2. si $X_i(q) \in R_i \cup E_i$ alors $X_i(q') \in E_i \cup C_i$, et,
3. si $X_i(q) \in C_i \cup S_i$ alors $X_i(q') \in S_i \cup R_i$.

On notera $q \xrightarrow{i} q'$ une telle transition, et l'on parlera d'une transition du processus p_i.

Lectures/écritures. On restreint la fonction de transition avec les conditions suivantes : pour tout $i \in \{0, \ldots, n-1\}$, X_i peut être partitionné en $2m$ ensembles disjoints Read_i^k et Write_i^k pour $k \in \{0, \ldots, m-1\}$ tels que, pour toute configuration q :

- Si $X_i(q) \in \mathit{Read}_i^k$, on dit que p_i est *prêt à lire la variable* k, c'est-à-dire que la prochaine transition *du processus* p_i sera une lecture de la variable k. La fonction de transition φ doit donc vérifier les conditions suivantes :
  1. $V(\varphi(i, q)) = V(q)$, et,
  2. pour toute configuration q' avec $X_i(q') = X_i(q)$, si $V_k(q') = V_k(q)$ alors $X_i(\varphi(i, q')) = X_i(\varphi(i, q))$.

  On dit alors que la transition $q \xrightarrow{i} \varphi(i, q)$ est une *lecture* de la k^evariable par p_i.

- Si $X_i(q) \in \mathit{Write}_i^k$, on dit que p_i est *prêt à écrire la variable* k, c'est-à-dire que la prochaine transition *du processus* p_i sera une lecture de la variable k. La fonction de transition φ doit donc vérifier les conditions suivantes :
  1. $V_j(\varphi(i, q)) = V_j(q)$ pour $j \neq k$, et,
  2. pour toute configuration q' avec $X_i(q') = X_i(q)$, on a $V_k(\varphi(i, q')) = V_k(\varphi(i, q))$ et $X_i(\varphi(i, q')) = X_i(\varphi(i, q))$.

  On dit alors que la transition $q \xrightarrow{i} \varphi(i, q)$ est une *écriture* de la k^e variable par p_i.

Histoire, calcul, atteignabilité, résultat. Une *histoire* de S est une suite (finie ou infinie) d'indices de processus (éléments de $\{0, \ldots, n-1\}$). Une occurrence de l'indice i dans une histoire est appelée *pas* du processus p_i dans cette histoire.

Si q_1 est une configuration et $h = i_1 i_2 \ldots$ une histoire, alors $q_1 q_2 \ldots$ est un *calcul* par h à partir de q_1, où $q_{j+1} = \varphi(i_j, q_j)$ pour $j = 1, 2, \ldots$. Si h est finie, alors $res(q_1, h)$ est la configuration finale dans le calcul par h à partir de q_1. On dit que la configuration q' est *atteignable* à partir de la configuration q s'il existe une histoire finie h telle que $q' = res(q, h)$. Une configuration est dite *atteignable* si elle est atteignable à partir de q_0.

Un point important est de voir que, puisqu'un calcul est une suite de transitions opérées chacune par un seul processus, un processus peut être prêt à lire (ou écrire) une variable pendant plusieurs transitions au cours d'un calcul avant que cette lecture (ou écriture) ne devienne effective.

Équité, arrêt. Une histoire h est *équitable* à partir de la configuration q si pour tout préfixe fini h_1 de h et pour tout $i \in \{0, \ldots, n-1\}$, $X_i(res(q, h_1)) \notin R_i$ implique que i apparaît dans h_2, où h_2 est défini par $h = h_1 h_2$.

Un processus *s'arrête* dans une histoire si et seulement si il y apparaît un nombre fini de fois. La notion d'arrêt est donc reliée à celle d'histoire, plutôt qu'à l'état d'un processus. Si une histoire h est équitable à partir de q, tout processus qui s'arrête dans h est dans sa section restante après son dernier pas dans le calcul par h à partir de q.

Exclusion mutuelle. Une configuration q *ne satisfait pas la propriété d'exclusion mutuelle* s'il existe deux valeurs distinctes i et j de $\{0, \ldots, n-1\}$ telles que $X_i(q) \in C_i$ et $X_j(q) \in C_i$. Le système S *satisfait la propriété d'exclusion mutuelle* si toute configuration atteignable q de S satisfait la propriété d'exclusion mutuelle.

Progrès global. Le processus p_i *change de section* par h en partant de q s'il existe des préfixes finis h_1 et h_2 de h tels que $X_i(res(q, h_1))$ est dans une section différente de celle de $X_i(res(q, h_2))$. Un système S a la propriété de *progrès global* si pour toute configuration atteignable q et toute histoire non nulle h équitable à partir de q, un processus au moins change de section par h en partant de q. On notera qu'un système ayant la propriété de progrès global peut laisser un processus en situation de famine, c'est-à-dire dans une situation où il tente d'entrer en section critique mais n'y parvient jamais.

II Une borne supérieure

L'algorithme de la figure 1 est exprimé dans un langage impératif de type Pascal. Il opère sur un tableau partagé *drapeau*[] de taille n, à valeurs dans $\{haut, bas\}$. À l'initialisation du système, toutes les valeurs de *drapeau*[] valent *bas*, et chaque processus p_i exécute le code du programme proc_i. En plus des sections restante, d'essai, critique et de sortie, on définit une sous-section d'essai et une section de passage à l'intérieur de la section d'essai, comme indiqué en commentaire dans l'algorithme. La section de passage correspond donc à la fin de la section d'essai.

On veut montrer que le système défini par cet algorithme vérifie les propriétés de progrès global et d'exclusion mutuelle.

```
program proc_i
    local var j : 0..n-1
begin
    while true do begin
        section restante                                          (* section restante *)
        repeat                                            (* début de la section d'essai *)
            drapeau[i] ← bas
            repeat until ∀j ∈ {0,...,i-1} : drapeau[j] = bas   (*sous-section d'essai*)
            drapeau[i] ← bas
        until ∀j ∈ {0,...,i-1} : drapeau[j] = bas
        repeat until ∀j ∈ {i+1,...,n-1} : drapeau[j] = bas       (*section de passage*)
        section critique                                          (* section critique *)
        drapeau[i] ← bas                                         (* section de sortie *)
    end
end.
```

FIG. 1 – Un algorithme d'exclusion mutuelle

Question 1. *Soit q une configuration telle que au moins un processus n'est pas dans sa section restante, et h une histoire équitable telle que aucun processus ne change de section par h en partant de q. Montrer que pour $i \in \{0, \ldots, n-1\}$, soit $X_i(q) \in R_i$ et i n'apparaît pas dans h, soit $X_i(q) \in E_i$ et i apparaît infiniment souvent dans h.*

On dit qu'un processus qui n'est pas dans R_i est *actif*.

Question 2. *Avec les mêmes hypothèses, montrer qu'il existe un préfixe de h tel que dans la configuration q' atteinte par ce préfixe en partant de q, tout processus actif est soit dans sa section de passage, ou dans un état à partir duquel il bouclera infiniment dans la sous-section d'essai.*

Question 3. *Toujours avec les mêmes hypothèses, montrer qu'il existe un processus qui, en partant de la configuration q', va changer de section par h.*

Question 4. *Conclure sur la propriété de progrès global.*

Question 5. *Montrer que l'algorithme vérifie la propriété d'exclusion mutuelle.*

III Une borne inférieure

On veut maintenant montrer le résultat suivant : tout système avec $n \geq 2$ processus vérifiant les propriétés d'exclusion mutuelle et de progrès global doit avoir au moins n variables partagées. Dans toute la suite, on désignera par S un système qui satisfait par hypothèse ces deux propriétés.

On commence par donner la définition suivante.

Configurations équivalentes. Soit q et q' deux configurations, et p_i un processus. On dit que q et q' sont équivalentes pour p_i, que l'on note $q \underset{i}{\sim} q'$, si $V(q) = V(q')$ et $X_i(q) = X_i(q')$.

Question 6. *Soit q et q' deux configurations, $\mathcal{P}'$ un sous-ensemble de $\mathcal{P}$, et h une histoire finie de S qui ne contient que des pas de processus de $\mathcal{P}'$. Montrer que si $q \underset{i}{\sim} q'$ pour tout $p_i \in \mathcal{P}'$, alors $V(res(q,h)) = V(res(q',h))$ et pour tout $p_j \in \mathcal{P}', X_j(res(q,h)) = X_j(res(q',h))$.*

Écriture oblitérée. Soit q_1 une configuration, h une histoire, $q_1 q_2 \dots$ étant le calcul de h en partant de q_1. Soit $i \in \{0, \dots, n-1\}, v \in \mathcal{V}$, et $j > 0$ tel que $q_j \xrightarrow{i} q_{j+1}$ est une écriture de v par p_i. S'il existe $k > j$ tel que $q_k \rightarrow q_{k+1}$ est une écriture de v, et pour tout l tel que $j < l < k$, $q_k \rightarrow q_{k+1}$ n'est *pas* une écriture de v par un processus autre que p_i, on dit que l'écriture de v par p_i en q_j est *oblitérée* par h en partant de q_1.

Processus caché. Soit q une configuration, h une histoire, et $i \in \{0, \dots, n-1\}$. On dit que p_i est *caché* de q par h s'il existe des histoires h_1 et h_2, h_1 finie, telles que $h = h_1 h_2$, $X_i(res(q, h_1)) \in R_i$, et chaque écriture de p_i dans le calcul de h_2 en partant de $res(q, h_1)$ est oblitérée par h_2 en partant de $res(q, h_1)$.

Soit q une configuration, h une histoire finie, et $\mathcal{P}'$ un sous-ensemble de $\mathcal{P}$ tel que chaque processus de $\mathcal{P}'$ est caché par h en partant de q. Soit $q' = res(q, h)$. Pour chaque $p \in \mathcal{P}'$, soit h_p le plus long préfixe de h tel que p est dans sa section restante à $res(q, h_p)$, et h_1 défini par $h = h_p h_1$. Soit k_p le nombre de pas de p dans le suffixe de h après h_p. On appelle ces pas des *pas cachés*.

Question 7. *S'il existe $p \in \mathcal{P}'$ tel que $k_p > 0$, soit p_i le processus de $\mathcal{P}'$ qui exécute le dernier pas caché de h. Soit h' l'histoire obtenue en supprimant de h le dernier pas de p_i. Montrer que tous les processus de $\mathcal{P}'$ sont cachés par h'.*

Question 8. *Montrer qu'il existe une configuration atteignable q'' de S telle que chaque p_i dans $\mathcal{P}'$ est dans sa section restante en q'', et $q'' \underset{j}{\sim} q'$ pour tout $p_j \in \mathcal{P}' \backslash \mathcal{P}$.*

Variable couverte. Si une variable est sur le point d'être écrite, tout processus en train d'écrire dans cette variable risque de voir son écriture oblitérée. Plus formellement, on donne la définition suivante : on dit que v est couverte par p_i en q lorsque p_i est prêt à écrire v en q, où q est une configuration, $i \in \{0, \dots, n-1\}$, et $v \in \mathcal{V}$.

Question 9. *Soit h une histoire finie de S, $q = res(q_0, h)$, et $i \in \{0, \dots, n-1\}$. Supposons que S a au moins deux processus, et que p_i est caché par h en partant de q_0. Montrer que si p_i atteint sa section critique à partir de q par une histoire finie $h_1 = i\ i\ \dots i$, alors p_i doit écrire au moins une variable dans le calcul de h_1 à partir de q qui n'est couverte par aucun autre processus en q.*

Question 10. *Pour un système vérifiant la propriété de progrès global et une configuration Q atteignable, montrer que l'on peut définir une histoire $lat(q)$ telle que $res(q, lat(q))$ est une configuration dans laquelle tous les processus sont dans leur section restante. On appellera* latente *une telle configuration.*

Tournez la page S.V.P.

Variable annulée. La combinaison des notions de variable couverte et de processus caché va nous mener au but poursuivi. Informellement, une variable est annulée par un processus si celui-ci ne "communique" pas avec d'autres processus, et couvre cette variable. Plus formellement : si q est une configuration, h une histoire finie et v une variable, on dit que v est *annulée* par h en partant de q s'il existe un processus caché par h en partant de q qui couvre v en $res(q, h)$.

Question 11. *Si S a au moins deux processus, montrer que pour toute configuration atteignable latente q_1, il existe une histoire finie n'utilisant que le processus p_0 et une variable w telles que w est annulée par h en partant de q_1.*

Question 12. *Soit k tel que $0 < k < n$, et tel que pour toute configuration atteignable latente q_1, il existe une histoire finie h utilisant seulement $p_0, \ldots, p_{k-1}$ et telle que k variables distinctes sont annulées par h en partant de q_1. Montrer que l'on peut construire des suites infinies $q_1, q_2, q_3, \ldots$, $h_1, h_2, h_3, \ldots$, et $W_1, W_2, W_3, \ldots$ telles que pour tout i*

1. *W_i est un ensemble de k variables distinctes,*
2. *h_i utilise seulement $p_0, \ldots, p_{k-1}$,*
3. *les variables de W_i sont annulées par h_i en partant de q_i.*

Question 13. *Avec les hypothèses de la question précédente, montrer qu'il existe une histoire h' utilisant seulement $p_0, \ldots, p_k$ telle que $k + 1$ variables distinctes sont annulées par h en partant de q_1.*

Question 14. *En déduire le résultat final attendu : tout système avec $n \geq 2$ processus vérifiant les propriétés d'exclusion mutuelle et de progrès global doit avoir au moins n variables partagées.*

IV Bornes sur l'attente

On appelle *tour* une utilisation de la section critique par un processus. Lorsqu'un processus p_i est désireux d'utiliser la ressource (donc qu'il se trouve dans sa section d'essai), on appelle *attente* de ce processus le nombre de tours effectués par les autres processus avant que p_i ne puisse entrer lui-même en section critique. Le but de cette partie est de donner des bornes supérieures sur l'attente pour certains protocoles d'exclusion mutuelle.

Question 15. *On considère le protocole d'exclusion mutuelle suivant.*

```
repeat
        drapeau[i] ← demande
        j ← tour
        while j ≠ i do begin if drapeau[j] ≠ passif then j ← tour
                                                    else j ← (j − 1) mod n
                             endif
                       end
        drapeau[i] ← dans_sc
until seul(i)
tour ← i
section critique
tour ← (i − 1) mod n
drapeau[i] ← passif
end.
```

où seul(i) *est le prédicat défini par*

$$(\forall j \neq i \ : \ drapeau[j] \neq dans\_sc) \wedge drapeau[i] = dans\_sc$$

Montrer que l'attente d'un processus est bornée par $2^{n-1} - 1$ *tours.*

Question 16. *On considère la variante suivante.*

```
repeat
        drapeau[i] ← demande
        j ← tour
        while j ≠ i do begin if drapeau[j] ≠ passif then j ← tour
                                                    else j ← (j − 1) mod n
                             endif
                       end
        drapeau[i] ← dans_sc
until seul(i)
section critique
if drapeau[tour] = passif ∨ tour = i
  then tour ← (tour − 1) mod n
endif
drapeau[i] ← passif
end.
```

Montrer que dans un intervalle de temps pendant lequel la variable tour est constante, tout processus peut passer au plus une fois en section critique.

Question 17. *Si $j \in \{1, \ldots, n-1\}$, montrer que dans un intervalle de temps où drapeau*$[0] \neq$ *passif, j peut pénétrer au plus $(n-j)$ fois en section critique.*

Question 18. *En déduire que l'attente d'un processus quelconque est bornée par $\frac{n(n-1)}{2}$ tours. Cette borne peut-elle être atteinte ?*

Question 19. *Proposer un protocole d'exclusion mutuelle sans interblocage pour lequel l'attente d'un processus est bornée par $(n-1)$.*