

Conception des systèmes répartis

3^{ème} Année Informatique et Mathématiques Appliquées
Section spéciale MAAI

Durée : 2 heures

Documents autorisés : Précis et notes de cours

14 octobre 2003

Remarque préliminaire : Toutes les questions valent 2 points.

1 Causalité (4 points)

On considère le chronogramme suivant :

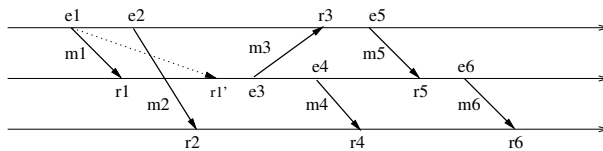


FIG. 1 – Chronogramme 1 : Causalité

Questions

1. Précisez si les événements r_4 et r_5 sont causalement liés ou pas. Justifiez votre réponse.
2. Montrez que la réception du message m_1 à la date réelle r'_1 au lieu de r_1 ne change pas les relations de causalité qui existaient. Autrement dit, les relations de causalité directe entre l'événement r_1 et les autres événements sont les mêmes que celles existant entre r'_1 et ces mêmes événements.

2 Une horloge minimaliste (8 points)

On considère un schéma de datation minimaliste utilisant simplement un compteur croissant sur chaque site. Chaque site possède une horloge locale H_s gérant une date logique sur laquelle les opérations d'incrément *Top* et de recalage *Recaler* peuvent être exécutées :

```
class Horloge {
    int cpt = 0 ;
    // Incrément-Lecture de l'horloge
    int Top() { cpt++; return cpt ; }
    // Recalage de l'horloge
    void Recaler( int d ) { if (cpt < d ) cpt = d ; }
}
```

L'opération $\text{Top}()$ incrémente sa valeur et renvoie la nouvelle valeur de l'horloge. L'opération $\text{Recaler}(d)$ affecte la valeur fournie en paramètre à l'horloge si et seulement si celle-ci était en retard.

Enfin, les actions de mise à jour de ces horloges lors des différents types d'événements sont les suivantes, où d_e (respectivement d_r) dénote la date de l'événement d'émission (respectivement la date de l'événement de réception) d'un message m :

Type d'événement sur un site s	Action mettant en jeu l'horloge du site s
Événement interne sur s	$H_s.\text{Top}()$
Émission sur s de m	$\text{int } d_e = H_s.\text{Top}()$; envoi de $\langle d_e, m \rangle$
Réception sur s de $\langle d_e, m \rangle$	$H_s.\text{Recaler}(d_e)$; $d_r = H_s.\text{Top}()$

FIG. 2 – Tableau des actions de mise-à-jour des horloges

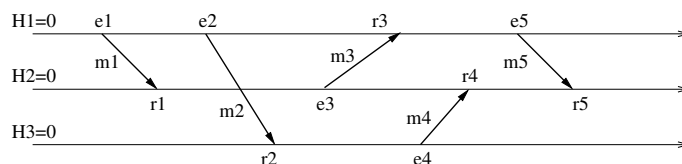


FIG. 3 – Chronogramme 2 : Datation

Questions

- Décorez le chronogramme de la figure 3 en précisant la date de chacun des événements affectés par le mécanisme d'horloge proposé. (Note : vous pouvez utiliser le chronogramme de la figure 5 en quatrième page comme support de la réponse.)
- Quelle propriété peut-on déduire des dates affectées à deux événements distincts e et e' lorsque $d_e = d_{e'}$?
- On appelle chemin causal $C(e_n)$ conduisant à un événement e_n , une suite d'événements $\{e_1, e_2, \dots, e_n\}$ vérifiant la propriété suivante : $e_1 \prec e_2 \prec \dots \prec e_n$.

La longueur d'un chemin causal, notée $|C(e_n)|$ est égale au nombre d'événements de la suite.¹ Montrez que la valeur de la date associée à un événement quelconque e est égale à la longueur du (ou des) plus long(s) chemin(s) causal(aux) conduisant à e . Autrement dit, l'invariant suivant est vérifié :

$$\forall e : \exists C(e) : d_e = |C(e)| \wedge \forall C'(e) : |C'(e)| \leq |C(e)|$$

Pour cela, montrez que si cette propriété est vraie avant l'événement, elle reste vraie après l'exécution de l'action correspondant au type d'événement considéré (voir tableau 2).

- Proposez une modification de l'opération de recalage qui permettrait d'obtenir l'invariant suivant : la date d'un événement est toujours supérieure au nombre total d'événements qui précèdent l'événement daté. Autrement dit, on aurait la propriété suivante :

$$\forall e : d_e \geq \text{Card}(G_e) \text{ avec } G_e = \{e' : e' \prec e\}$$

Note : Proposez une solution simple qui ne fasse toujours intervenir que la date "incidente" en paramètre et la valeur courante de l'objet horloge "réceptrice" .

¹Par convention, on admet qu'un chemin causal $C(e)$ réduit à l'événement lui-même ($C(e) = \{e\}$) est donc de longueur 1.

3 Prise de cliché global(8 points)

On considère un ensemble de N processus P_i observés par un processus *Collecteur*. On suppose que les communications sont fiables et point à point mais non FIFO. On cherche un algorithme permettant au *Collecteur* de construire un cliché global cohérent en utilisant le mécanisme de vague et un schéma de coloriage des processus et messages applicatifs.

Rappel Un cliché global comprend d'une part, les états locaux de chaque processus et d'autre part, les messages en transit associés à la coupure "traçée" par les états locaux collectés. Une coupure est cohérente si et seulement si les messages en transit traversent cette coupure du passé vers le futur (sur un chronogramme, l'événement d'émission est à gauche de la coupure et l'événement de réception correspondant est à droite de celle-ci). Un cliché est cohérent ssi il constitue une coupure cohérente.

Règles de coloriage On utilise seulement deux couleurs : blanc et noir.

- initialement, tous les processus et messages sont blancs ;
- un processus récepteur devient noir s'il reçoit un message noir ;
- un processus émet toujours des messages de sa propre couleur ;
- un fois noir, tout processus reste noir ;

Le processus *Collecteur* lance une vague de couleur noire. Un processus visité devient donc noir dès le passage de la vague (s'il ne l'était pas déjà). Lorsqu'un processus **blanc** reçoit la visite de la vague, il enregistre un cliché local de son état et devient noir.

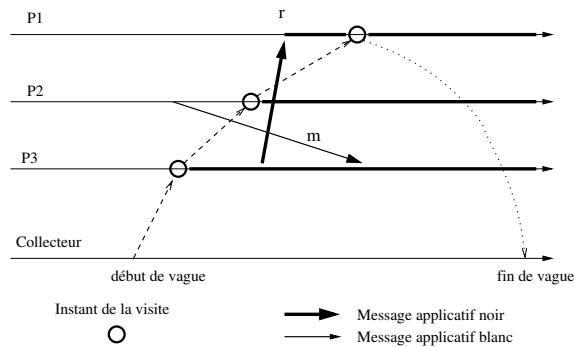


FIG. 4 – Prise de cliché par vague

Questions

7. Si un processus **noir** visité par la vague prend son cliché local au moment du passage de la vague (comme le fait un processus blanc), montrez que le cliché obtenu peut être incohérent (S'appuyer sur la figure 4). En conséquence, pour obtenir un cliché cohérent, quelle mesure doit prendre un processus blanc lorsqu'il devient noir par réception d'un message applicatif noir (événement r dans la figure 4) ?
8. Le *Collecteur* doit aussi obtenir les messages en transit tels que le message m dans la figure 4. Caractérisez la classe d'événements de réception qui permettent de détecter les messages en transit (couleur du message reçu, couleur du processus récepteur).
9. On suppose que, à chaque événement de réception précédent (détecteur d'un message en transit), le processus récepteur transmet le message reçu au collecteur. Proposez un algorithme pour que le *Collecteur* puisse savoir qu'il a reçu tous les messages en transit associés au cliché (problème particulier de terminaison).
10. Proposer une modification de l'algorithme pour pouvoir faire plusieurs clichés successifs à partir d'un même collecteur.

Question 3

Affecter à chaque événement e_i ou r_i sa date.

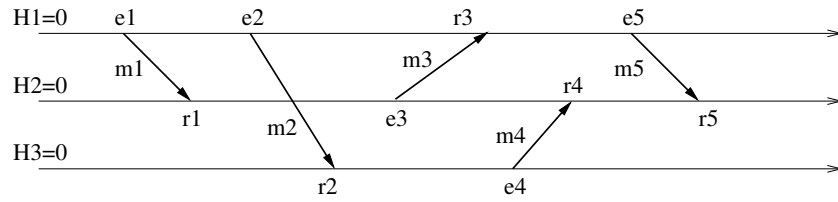


FIG. 5 – Chronogramme à décorer