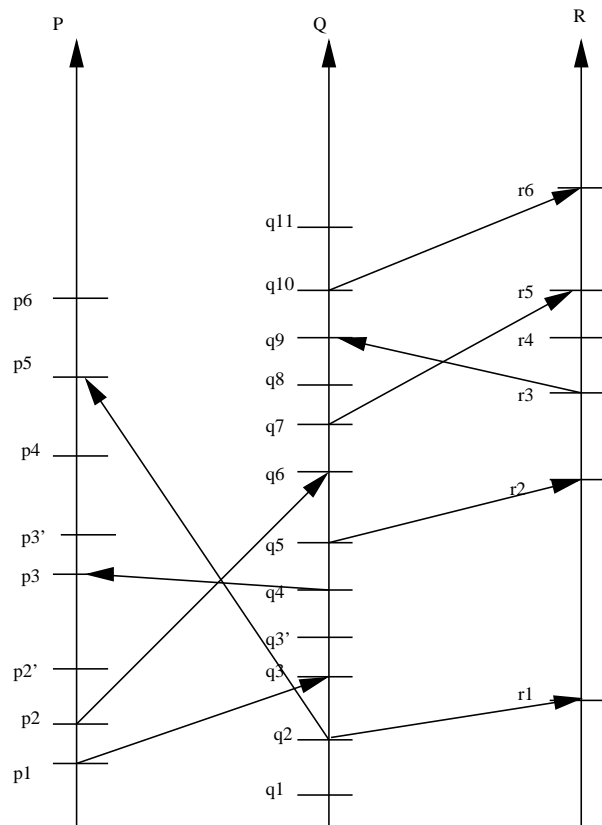


Exercices Master Informatique, IFI, Parcours CSSR
Course 2
2009-2010

F. Baude

Exercice 1 about Integer Clocks

Soit le diagramme de causalité suivant:



1. Que représentent les événements p_5 , q_1 , p_2 et r_4 ?
2. Au vu du diagramme, les voies de communication sont-elles FIFO ?
3. Peut-on avoir:
 - une émission de message en q_5 et sa réception en p_2' ? (en supposant donc que les événements correspondant à l'échange de ce message remplacent les événements actuellement dessinés sur le diagramme).

- une émission de message en $p3'$ et sa réception en $q3'$ (même remarque que pour le point précédent) ?
4. En considérant les horloges de Lamport initialisées à 0, quelles sont les valeurs des horloges en P après $p6$, en Q après $q11$ et en R après $r6$?
 5. Quelle est la durée maximum de l'algorithme décrit par ce diagramme (on néglige la durée des événements locaux et on considère que la durée d'une transmission est bornée par τ_{max}). De même, donner la durée minimum de l'algorithme (ou dit autrement, donner la profondeur du diagramme d'événements).
 6. Quels sont les événements indépendants (au sens de la relation de causalité) de $q6$?
 7. Montrer que si 2 événements a et b sont tels que $heure(a) < heure(b)$, on n'a pas nécessairement que a précède b au sens de la relation de causalité.

Exercice 2 about Vector Clocks

On considère le diagramme temporel (dessiné en figure 1) associé à un algorithme distribué faisant intervenir 3 processus $P1, P2, P3$ et les événements a_1, b_1, c_1, d_1 sur $P1$, a_2, b_2, c_2, d_2 sur $P2$, a_3, b_3, c_3, d_3 sur $P3$.

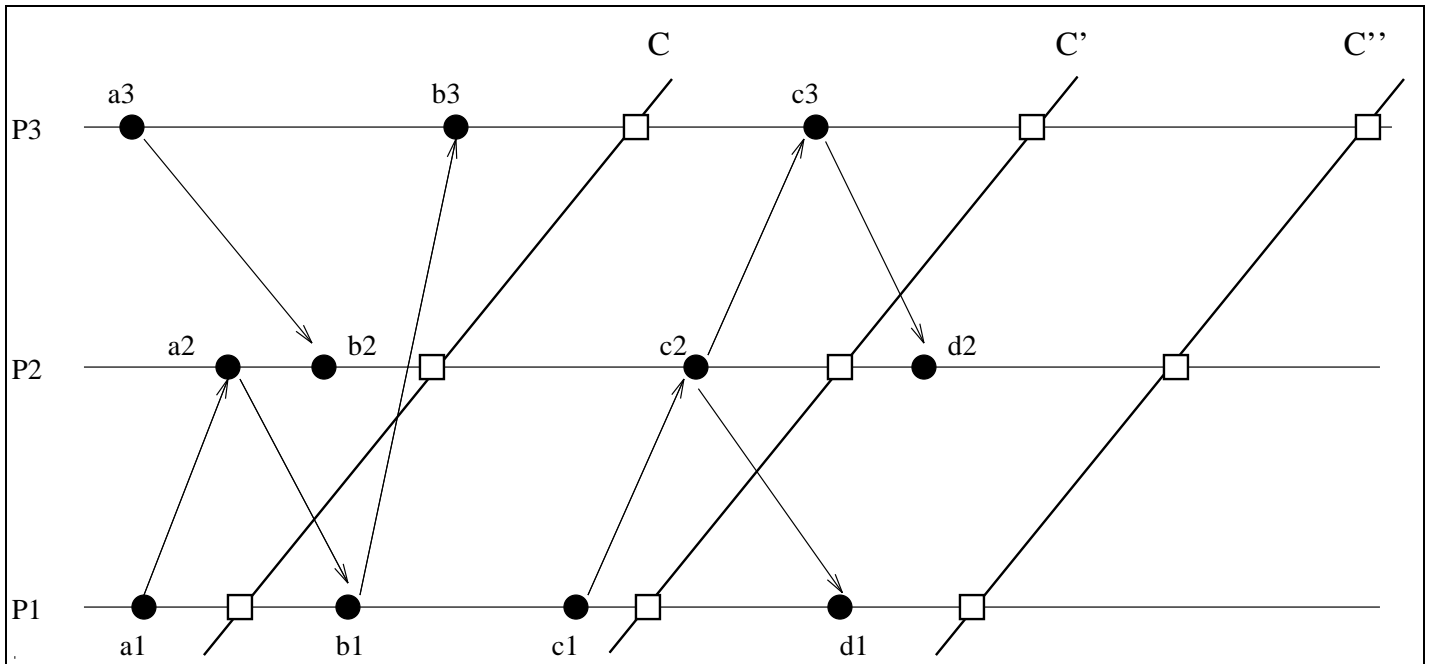


Figure 1: Diagramme temporel

1

Est-ce que la façon dont est présenté ce diagramme temporel utilise l'idée du modèle atomique (voir article en anglais de Lamport) ?

2

Calculer les estampilles de Lamport (entiers) pour chacun de ces événements.

3

Calculer les “horloges vectorielles” (vecteurs à 3 composantes) pour chacun de ces événements.

4

Commenter les valeurs des vecteurs estampillant les événements c1 et c3.

5

Dire comment les événements b3 et c2 estampillés par les horloges de Lamport se comparent (sont-ils comparables selon la relation de causalité notée \rightarrow) ?

Puis, en considérant leur estampille vectorielle, dire comment ces événements b3 et c2 se comparent.

Recommencer exactement la même question en considérant à présent les événements b3 et d2.

6

Décrire en termes d'événements les coupes C , C' , C'' . Sont-elles cohérentes ?

7

Soit r_i le nombre de messages reçus par P_i et s_i le nombre de messages émis par P_i pour une coupe C . Calculer $\sum_i r_i = R$ et $\sum_i s_i = S$ pour les coupes C , C' , C'' .

8

Est-ce que les valeurs obtenues pour la coupe C'' permettent de conclure que l'algorithme est terminé ?

Exercice 3 (from Chapter 6 of Ghosh)

The limitation of timestamps is their unbounded size, since finite resources are inadequate to store or process them. The goal of this exercise is to explore if bounded-size timestamps can be used in specific solutions. Explore this possibility in the following scenario: Two processes (0,1) compete with each other to acquire a shared resource that can be used by one process at a time. The life of the processes are as follows:

Process i ($i = 0$ or 1)

do

 true \rightarrow

 Request for a resource;

 Acquire and use the resource;

```
        Release the resource
    od
```

To request the resource, a process sends a timestamped request to the other process, which grants the request only if (i) it is not interested in the resource at that moment, or, (ii) its own timestamp for resource request is larger than the timestamp of the incoming request. In all other situations, the grant is deferred. After receiving the grant, a process acquires the resource. Once a process acquires a resource, it guarantees to release the resource within a finite amount of time. Thereafter, in a finite time, the resource is released.

Can you solve the problem using timestamps of bounded size ? Explain your answer. (Hint: First find out what is the maximum difference between the timestamps of the two processes if the timestamps are unbounded. The timestamp of bounded size must resolve the order of requests without any ambiguity.)

Exercise 4 (adapted from one from Chapter 6 of Ghosh)

In a network of N processes ($N > 2$), all channels are FIFO, and of infinite capacity. Every process is required to **accept data from the other processes in strictly increasing order of timestamps**. You can assume (i) processes send data infinitely often, and (ii) no messages is lost in transit.

First, build a 3 processes example, that illustrates this requirement: Process P1 sends 2 messages, one to P2, one to P3. On reception, P3 sends a new message to P2. All messages are timestamped using Lamport's clocks.

Then, propose an implementation to make it possible that processes respect the requirement. (Hint: Consider using null messages through a channel to signal the absence of a message from a sender).

Exercise 5 (from Chapter 8, Ghosh)

Construct an example to show that Chandy-Lamport distributed snapshot algorithm does not work when the channels are not FIFO.

Exercise 6 (from Chapter 11, Coulouris and al. book)

Two processes P and Q are connected in a ring using two-channels, and they constantly rotate a message m . At any one time, there is only one copy of m in the system. Each process's state consists of the number of times it has received m , and P sends m first. At a certain point, P has the message and its state is 101. Immediately after sending m , P initiates the snapshot algorithm. Explain the operation of the algorithm in this case giving the global state reported by it.