

DETERMINATION D'UN ETAT GLOBAL COHERENT

1. Introduction

L'observation d'un état global permet

- de détecter des propriétés caractérisant l'exécution (terminaison, interblocage, vérification d'assertions, etc.)
- d'effectuer des mesures de performances
- de capter des états pouvant servir de points de reprise en cas de défaillance

Mais il est Difficile d'effectuer une observation globale dans un contexte réparti à cause de :

- pas d'horloge commune
- temps de transfert des messages non borné (fini si pas de perte)
- impossible d'effectuer une observation simultanée par exemple : des photographies des vols d'oiseaux

L'objectif de cette partie est la détermination d'un état global cohérent qui peut être observé.

Cet état est constitué des états locaux des sites et des canaux de communication.

2. Etat global

2.1. Définitions

Chaque processus et chaque canal possède à tout moment un **état local** :

- ✓ l'état local el_i d'un processus P_i résulte de son état initial et de la séquence d'événements dont ce processus a été le siège.
- ✓ l'état ec_{ij} d'un canal c_{ij} est l'ensemble des messages en transit sur ce canal, c'est à dire qui ont été émis par le processus P_i et n'ont pas encore été reçus par le processus P_j .

Chaque **événement** met en jeu un processus et éventuellement un canal. On distingue :

- **Événement interne** sur P_i : provoque la transition de el_i à el_i' , états avant et après l'événement

- **Émission de m** par P_i sur c_{ij} (cet événement est noté $émission_i(m)$) qui provoque : la transition de el_i à el_i' et l'affectation $ec_{ij} := ec_{ij} \cup \{m\}$
- **Réception de m** par P_i sur c_{ji} (cet événement est noté $réception_i(m)$) qui provoque : la transition de el_i à el_i' et l'affectation $ec_{ji} := ec_{ji} \setminus \{m\}$.

Chacun de ces événements est supposé *atomique*

Formellement, un **état global cohérent** est tel que :

i) el_i est un état local du processus P_i

ii) Les conditions **C1** et **C2** suivantes sont vérifiées:

- **C1** : si l'événement $émission_i(m)$ est capté dans el_i , alors l'événement $réception_j(m)$ est soit capté dans el_j , soit le message m appartient à ec_{ij}
- **C2** : si l'événement $émission_i(m)$ n'est pas capté dans el_i , l'événement $réception_j(m)$ n'est pas non plus capté dans el_j

Un état global cohérent est aussi appelé “**coupe cohérente**” (« *consistent cut* »).

Exemple :

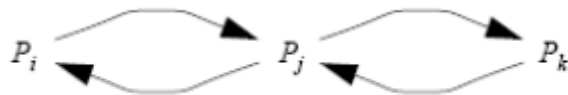


Figure 1. Un graphe de communication.

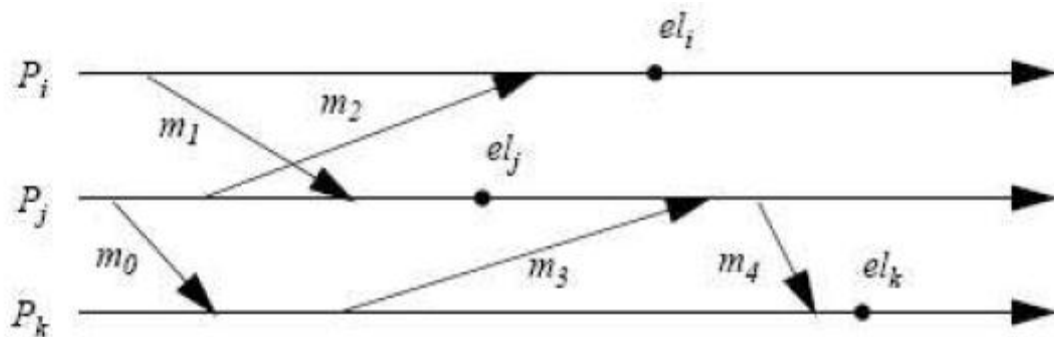


Figure 2. Une exécution répartie.

1.2 Définition équivalente d'une coupure cohérente

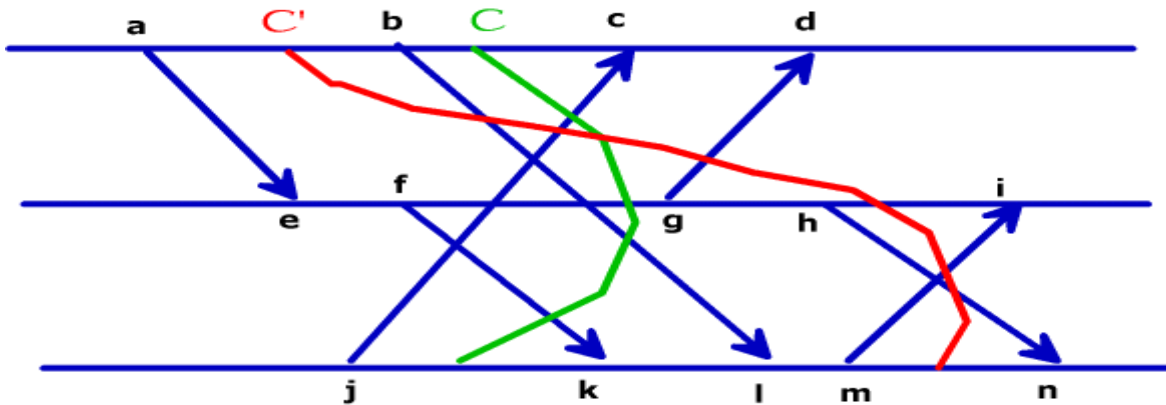
Une **coupure C** de l'histoire globale d'un système est un n-uplet dont chaque composante est un préfixe de l'histoire locale du site correspondant :

$$C = \langle c_1, \dots, c_j, \dots, c_n \rangle \text{ où } c_j = \langle e_{j1}, e_{j2}, \dots, e_{jm} \rangle$$

Une **coupure C** est dite **cohérente** (ou consistante) si elle est fermée vis à vis du passé de ses éléments : tout élément dans le passé d'un événement de la coupure appartient lui-même à la coupure.

Exercice

Dans l'exemple suivant, les coupures C et C' sont-elles cohérentes ??



1.3. Caractérisation des coupures cohérentes par l'estampillage vectoriel de Lamport

On associe à une coupure C la suite des événements $\langle e_1, \dots, e_j, \dots, e_n \rangle$ (e_j est l'événement le plus récent du site S_j appartenant à la coupure). Associons à la coupure C l'estampille vectorielle $EV(C)$ définie par : $EV(C) = \langle EV(e_1) [1], \dots, EV(e_j) [j], \dots, EV(e_n) [n] \rangle$.

La **coupure est cohérente** si et seulement si :

$$EV(C) = \sup (EV(e_1), \dots, EV(e_j), \dots, EV(e_n)), \text{ avec : } \sup (EV(e_1), \dots, EV(e_j), \dots, EV(e_n)) [k] = (\sup (EV(e_1) [k], \dots, \sup (EV(e_j) [k], \dots, \sup (EV(e_n) [k]$$

Exercice : appliquer ce résultat aux coupures C et C' de l'exemple précédent.

3. Hypothèses sur les canaux de communication

3.1. Définition des types de messages

Soit c un canal, m et $m1$ deux messages empruntant ce canal.

On dit que m double $m1$ si et seulement si :

$$\text{émission}(m1) \rightarrow \text{émission}(m) \text{ et réception}(m) \rightarrow \text{réception}(m1)$$

Quatre types possibles de messages peuvent être définis selon les contraintes de doublement :

i) Un message m est de type **marqueur** s'il ne peut ni doubler ni être doublé par aucun message transitant sur le même canal :

$$\square m1, \text{émission}(m1) \rightarrow \text{émission}(m) \Rightarrow \text{réception}(m1) \rightarrow \text{réception}(m)$$

$$\text{et } \square m2, \text{émission}(m) \rightarrow \text{émission}(m2) \Rightarrow \text{réception}(m) \rightarrow \text{réception}(m2)$$

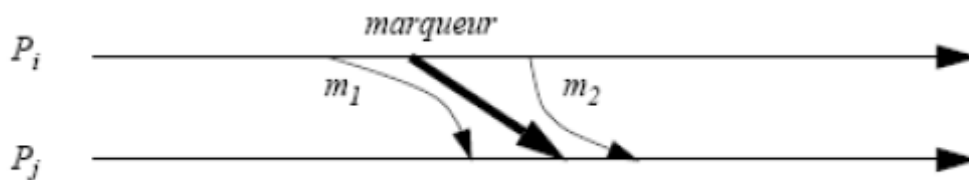


Figure 3. Livraison d'un marqueur.

ii) Un message m est de type **ct_passé** (contraint par son passé) s'il ne peut doubler aucun message : $\square m1, \text{émission}(m1) \rightarrow \text{émission}(m) \Rightarrow \text{réception}(m1) \rightarrow \text{réception}(m)$

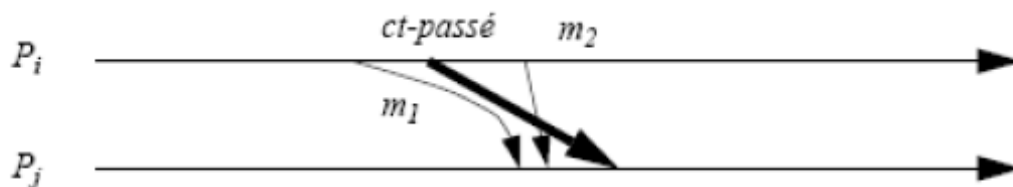


Figure 4. Livraison d'un message contraint par son passé.
(tous les messages émis avant lui sont reçus avant lui)

iii) Un message m est de type **ct_futur** (contraint par son futur) s'il ne peut être doublé par aucun message : $\square m2, \text{émission}(m) \rightarrow \text{émission}(m2) \Rightarrow \text{réception}(m) \rightarrow \text{réception}(m2)$.

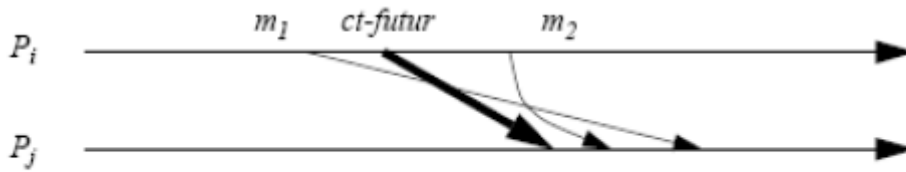


Figure 5. Livraison d'un message contraint par son futur.
(tous les messages émis après lui seront reçus après lui)

iv) Un message m de type *ordinaire* n'impose pas de conditions de réception :

- il ne peut pas doubler les messages *marqueur* et *ct_futur*
- il ne peut pas être doublé par les messages *marqueur* et *ct_passé*

Les types de messages définissent une gamme de comportement du canal :

- le moins contraint (tous les messages sont ordinaires)
- le plus contraint (tous les messages sont de type marqueur) => canal FIFO

Pour un canal donné, les trois propriétés A1, A2 et A3 suivantes sont équivalentes :

- A1 : tous les messages sont de type *ct_passé*
- A2 : tous les messages sont de type *ct_futur*
- A3 : tous les messages sont de type *marqueur*

3.2. Hypothèses globales sur les canaux de communication

La propriété d'ordre causal porte sur l'ensemble des canaux.

□ P_i, P_j, P_k , □ m émis sur C_{ij} , □ m_1 émis sur C_{kj} :

$$\text{émission}_i(m) \rightarrow \text{émission}_k(m_1) \Rightarrow \text{réception}_j(m) \rightarrow \text{réception}_j(m_1)$$

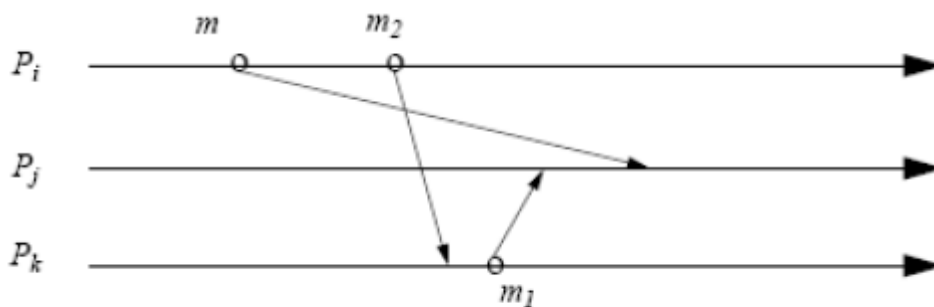


Figure 6. Réception non causale.

Remarque : si la propriété d'ordre causal est garantie alors chaque canal a un comportement FIFO (réciproque fausse).

3.3. Superposition

A chaque processus P_i est associé un processus observateur CTL_i . Ce processus peut lire l'état de P_i . La réception d'un message par P_i est réalisée par CTL_i , qui le délivre ensuite à P_i . P_i confie à CTL_i les messages qu'il veut transmettre vers tout autre processus P_j . Les processus CTL_i sont chargés de capter un état global cohérent de l'application et ils coopèrent entre eux à l'aide de messages de contrôle.

Un tel schéma d'observation des processus P_i par des contrôleurs CTL_i est appelé *superposition*.

4. Algorithme de Chandy et Lamport

4.1. Principe du fonctionnement :

Chaque processus P_j enregistre son état local e_j et l'état de ses canaux entrants en observant les deux règles suivantes :

R1 : un processus P_j qui capte son état local envoie sur tous ses canaux de sortie un message de contrôle (appelé **mk**) pour signifier à ses voisins qu'il a enregistré son état local.

R2 : à la réception d'un message **mk** sur un de ses canaux d'entrée, par

exemple c_{ij} , un processus P_j peut se trouver dans l'une des deux situations suivantes :

- P_j n'a pas encore capté son état local : il doit alors enregistrer cet état (R1) \Rightarrow l'état ec_{ij} du canal c_{ij} est vide car P_j a reçu tous les messages émis avant le message **mk**.
- P_j a déjà enregistré son état local (soit de sa propre initiative, soit suite à une réception antérieure d'un message **mk** sur un autre canal d'entrée) \Rightarrow l'état du canal c_{ij} est constitué des messages reçus sur ce canal après que P_j ait enregistré son état local et avant qu'il ne reçoive **mk**.

