IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

TD 1 – Modélisation de la causalité et du temps dans les systèmes répartis

Échange de messages et aspects temporels

FIAB

En Elixir, nous avons vu qu'une application est composée d'un ensemble de processus qui s'exécute sur un ensemble de nœuds. Chaque nœud a sa propre mémoire et sa propre vitesse de calcul. L'ensemble des nœuds doit se connaître pour permettre des échanges. L'ensemble de tous les processus sont indépendants et communiquent avec les autres par envoi de messages. La communication suit donc par défaut une approche asynchrone (sur laquelle, on peut éventuellement reconstruire des échange en requête réponse).

Nous allons construire un modèle logique d'une telle application pour en analyser certains aspects. Comme tout modèle, il est construit par abstraction. Nous allons donc ignorer certains détails.

Exercice 1 (*Préliminaires*)

▷ **Question 1.1 :**
Si notre but est d'analyser les communications et l'évolution de chacun des processus, quels sont, selon vous, les éléments dont nous avons besoin et ceux que l'on peut ignorer ?

▷ **Question 1.2 :**
Dans quelle situation, les éléments ignorés ci-dessus devraient être pris en compte ?

Exercice 2 (*Événements et relations causales*)

On définit la relation suivante dite de *causalité* : «*un événement* e_i^x *précède un événement* e_j^y». Elle s'écrit $e_i^x \xrightarrow{ev} e_j^y$.

Un événement e_i^x précède un événement e_j^y si et seulement si l'une des trois conditions suivantes est vraie :

1. les événements se déroulent au sein d'un même processus
2. les deux événements concernent le même message m

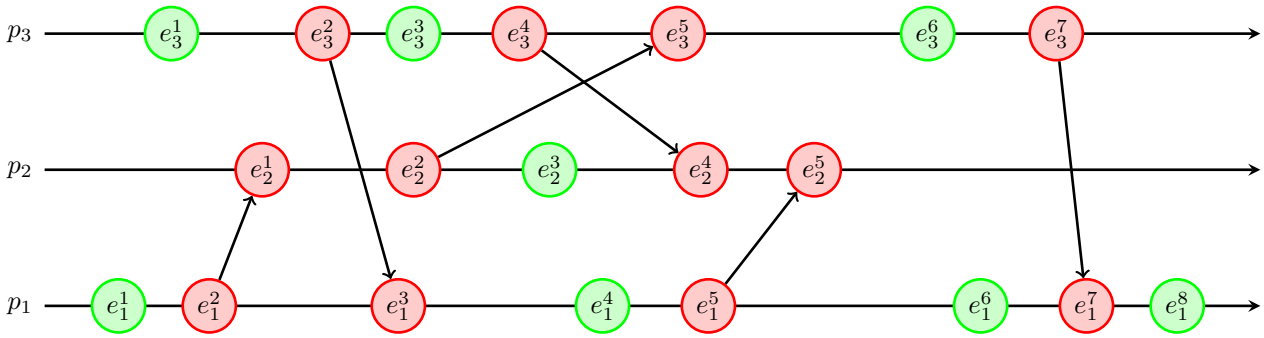


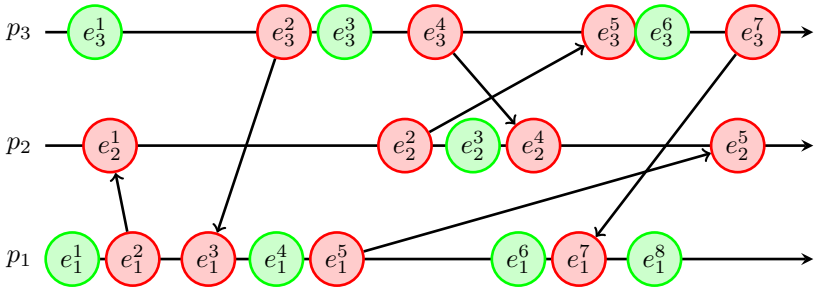
FIGURE 1 – Un modèle avec 3 processus qui s'échangent des messages

3. une transition par un événement indépendant existe

▷ **Question 2.1 :**
Définir formellement ces trois conditions (attention, c'est trivial).

La figure 1 contient un exemple de trace d'une exécution d'un système distribué constitué de trois processus p_1, p_2 et p_3. Les événements internes sont représentés en vert et les événements de communication sont en rouge. Un événement rouge qui est la source d'une flèche représente un envoi de message tandis qu'un destination représente une reception.

▷ **Question 2.2 :**
Quelle différence faites-vous entre ce diagramme et la réprésentation ci-dessous ? Pourquoi ?



Un *chemin causal* est défini comme étant une suite d'événements consécutifs au sens de la relation $\xrightarrow{ev}$. Par extension, le *passé causal* (respectivement *futur causal*) d'un événement e est l'ensemble des événements situés avant (respectivement après) e dans l'ensemble des chemins causaux.

▷ **Question 2.3 :**
Dans le diagramme de la figure 1, quel est le passé causal de e_2^2 ? Quel est son futur causal ? Quelle est la définition formelle du passé (resp. futur) causal d'un événement ?

Soit F un ensemble de n événements donnés, un événement par processus. Une *coupe* est l'ensemble des événements antérieurs aux événements de F dans chaque processus. On appelle *frontière* cet

ensemble F. Par exemple dans le diagramme précédent, une coupe C_1 est définie par sa frontière $\{e_1^3, e_2^1, e_3^4\}$ et la coupe C_2 est définie par $\{e_1^3, e_2^1, e_3^5\}$ [1].

▷ **Question 2.4 :**
Donnez la définition formelle de la notion de coupe. Quelle différence fondamentale voyez-vous entre les coupes C_1 et C_2 ? Définissez formellement une coupe *cohérente*.

Avez-vous une formulation qui permettrait à un enfant de savoir si une coupe est cohérente sur un diagramme de temps comme celui représenté ci-dessus ?

Exercice 3 (*Cohérence d'états et exécutions*)

Nous désignons maintenant l'état dans lequel se trouve le processus i immédiatement après l'événement e_i^x par σ_i^x. Naturellement, le processus est dans cet état jusqu'à l'événement e_i^{x+1}.

On peut définir une relation causale entre les états par la transition $\xrightarrow{state}$. Ainsi, nous avons :

$$e_i^{x+1} \xrightarrow{ev} e_j^y \iff \sigma_i^x \xrightarrow{state} \sigma_j^y$$

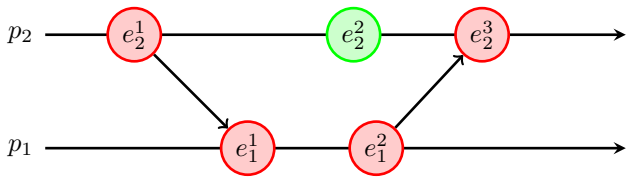
On parle d'état local pour désigner l'état d'un processus. Un *état global* du système est un ensemble d'états locaux, un par processus du système. Par exemple $\Sigma = \{\sigma_1^{x_1}, \ldots, \sigma_n^{x_n}\}$.

▷ **Question 3.1 :**
Qu'est-ce qu'un état global cohérent ? Qu'est-ce que cela signifie concrètement dans un système distribué ?

Une *trace* est un ensemble ordonné d'événements. On parle également d'*exécution*.

▷ **Question 3.2 :**
Qu'est-ce qu'une exécution cohérente ?

Partons d'un diagramme relativement simple.



▷ **Question 3.3 :**
Déterminez l'ensemble des exécutions cohérentes du système et tentez de le représenter.

Exercice 4 (*Propriétés*)

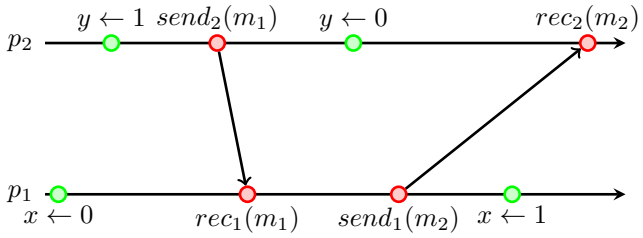
On dit qu'un état global Σ_2 est *atteignable* à partir d'un état global Σ_1 s'il existe une exécution cohérente qui permet de passer de Σ_1 à Σ_2. Un état global peut vérifier une propriété donnée, ou pas (on parle également de prédicat).

1. Un événement à la frontière d'une coupe est considéré comme étant inclus dans la coupe.

▷ **Question 4.1 :**
Qu'est-ce qu'une propriété globale stable ? Donnez quelques exemples de propriétés globales stables.

Définissez la *vivacité* et la *sûreté* d'un système distribué.

Voici un diagramme.



▷ **Question 4.2 :**
Dans ce système, quelle est la caractéristique de la propriété : $x = 0 \ \wedge \ y = 0$.

Exercice 5 (*Horloges*)

▷ **Question 5.1 :**
Dans le modèle de la figure 1, qu'est-ce qui pourrait jouer le rôle d'horloge ?

Une horloge de Lamport est un compteur local à un processus qui progresse à chaque événement interne et est transmis lors des envois de message. À la réception, le processus récepteur met à jour son horloge par une opération *max*. Ce compteur local est donc stockée dans l'état du processus et :

- il vaut 0 initialement,
- un événement local provoque l'incrément de ce compteur,
- un envoi provoque également l'incrémentation et le résultat est ajouté à l'envoi,
- lors d'une reception le compteur est remplacé par le maximum entre le compteur local et celui reçu dans le message puis incrémenté.

Dans sa définition formelle, l'horloge de Lamport associe à chaque événement la valeur du compteur après l'événement.

▷ **Question 5.2 :**
Construire sur le modèle la mise en œuvre de l'horloge de Lamport.

Une horloge vectorielle est un vecteur qui encode pour chaque processus sa connaisance de l'horloge de tous les processus. Les messages sont l'occasion du partage de cette connaissance. À la réception, un *max* est fait sur les vecteurs.

▷ **Question 5.3 :**
Construire sur le modèle la mise en œuvre d'une horloge vectorielle.

▷ **Question 5.4 :**
Que pensez-vous de ces horloges vis-à-vis de la relation de causalité définie plus haut ?