Le consensus
Un rsultat d'impossibilité
dans les systèmes répartis asynchrones

Frédéric Tronel

Irisa - Projet ADP

Séminaire SRS, 20 Juin 2000

Modèle Asynchrone

- Ensemble de processus reliés deux à deux par des canaux fiables;
- Possibilité de processus défaillants;
- Communications par passage de messages;
- Pas d'horloge globale, ni de mémoire partagée;
- Pas de borne sur les temps de transmission, ni sur les temps de calcul.

$\longrightarrow$ Modèle général et simple.

Qu'est que le problème du Consensus ?

- Problème d'accord entre processus;
- Accord sur une valeur choisie parmi les propositions des processus.
- La valeur finalement décidée est commune à tous.

Consensus Uniforme

Définition formelle

Quatre propriétés :

- Terminaison $\longrightarrow$ corrects décident finalement;
- Intégrité $\longrightarrow$ au plus une décision;
- Validité $\longrightarrow$ décision est une valeur initiale;
- Accord $\longrightarrow$ décision unanime des processus.

Modélisation d'un protocole

- Chaque site est modélisé par un automate.
- Chaque automate peut envoyer (recevoir) des messages.
- Pour chaque automate, on distingue quatre états :
  - ⋆ Deux états initiaux I_0 et I_1.
  - ⋆ Deux états finaux F_0 et F_1.
- $F_{i \in \{0,1\}}$ sont des états puits.

Modélisation d'un protocole

- Les automates s'envoient des messages (p_i,m) :
  - $\star$ p_i identité du destinataire.
  - $\star$ $m \in M$ message lui-même.
  - $\star$ M univers prédéfini des messages.

- Existence d'un multiensemble $\mathcal{M}$ de messages définis par deux primitives :
  - $\star$ **send**(p_i,m) : $\mathcal{M} = \mathcal{M} \cup (p_i,m)$
  - $\star$ **receive**(p_i) : $(p_i,m) \in \mathcal{M} \vee \emptyset$

- Un nombre d'appels infinis à receive(p_i) conduit à la réception d'un message (s'il en existe un).

Modélisation d'un protocole

- Une *configuration* C est un couple $(s,\mathcal{M})$
  - $\star$ $s(i)$ état de l'automate associé au site p_i.
  - $\star$ $\mathcal{M}$ multiensemble de messages.

Modélisation d'un protocole

- Soient $C_1 = (s_1,\mathcal{M}_1)$, $C_2 = (s_2,\mathcal{M}_2)$ deux configurations.
- Un *pas* $e = (p_i,m)$ est applicable à C_1 ($C_2 = e(C_1)$) si :
  - ⋆ p_i reçoit un message m_1 ($m_1 \in \mathcal{M}_1 \cup \emptyset$).
  - ⋆ $\forall j \neq i \quad s_1(j) = s_2(j)$.
  - ⋆ p_i passe de l'état $s_1(i)$ à $s_2(i)$
  - ⋆ p_i émet un message $m_2 \in M \cup \emptyset$.
  - ⋆ $\mathcal{M}_2 = \mathcal{M}_1 \backslash m_1 \cup m_2$.

Modélisation d'un protocole

- Une séquence σ est une suite de pas $(e_k)_{k \in I}$ (possiblement infinie).

- σ est applicable à C et conduit à C' si :

$$\exists C_{k \in I} \quad \begin{cases} C &= C_0 \\ C_i &= e_i(C_{i-1}) \\ C' &= C_r \end{cases}$$

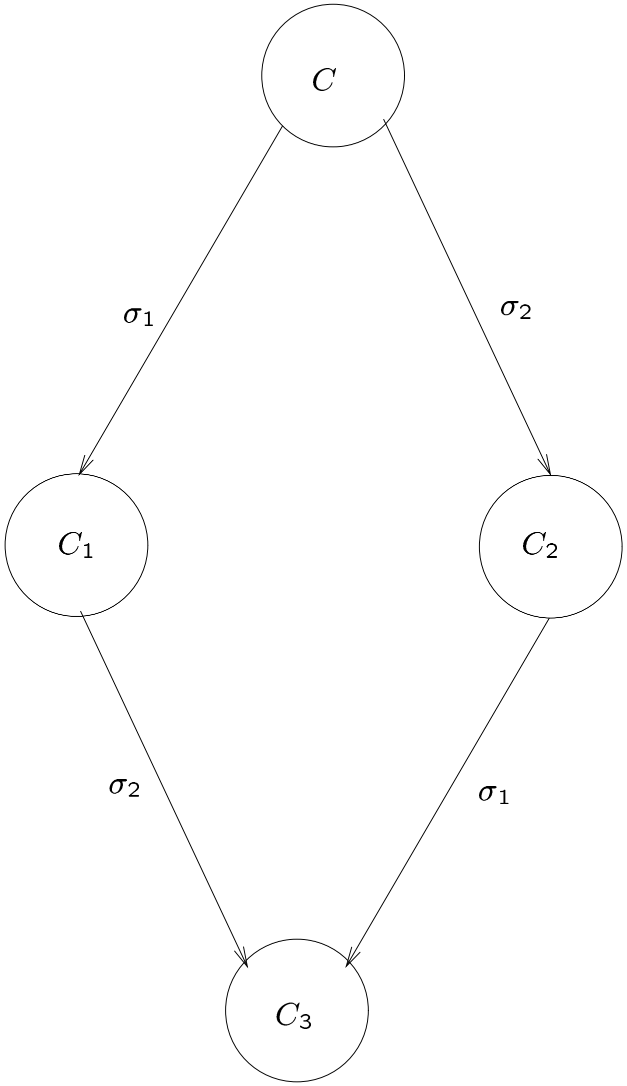
- $C' = \sigma(C)$ et C' est dite *atteignable* depuis C.

- C est dite *accessible* si elle est atteignable depuis une configuration initiale.

Lemme 1

- C une configuration.
- Séquences σ_1 (resp. σ_2) conduisant à C_1 (resp. C_2) depuis C.
- Si les ensembles de processus prenant des pas dans σ_1 et σ_2 sont disjoints
- Alors il existe une configuration C_3, telle que σ_1 (resp. σ_2) est applicable à C_2 (resp. C_1) et mène à C_3.

Lemme 1



Modélisation d'un protocole

- $C = (s,\mathcal{M})$ possède une *valeur de décision* $v \in \{0,1\}$ si $\exists j$ tel que $\quad s(j) = F_v$.

- un protocole P est partiellement correct si :
  - ⋆ Aucune configuration accessible possède plus d'une valeur de décision.
  - ⋆ $\forall v \in \{0,1\}$, il existe une configuration accessible possédant comme valeur de décision v.

Modélisation d'un protocole

- σ une séquence infinie.
- Un processus p_i est *correct* s'il prend une infinité de pas, sinon il est *défaillant*.
- σ est *admissible* si :
  - ⋆ σ comporte au plus un processus défaillant.
  - ⋆ Tous les messages émis vers des processus corrects sont reçus.

Modélisation d'un protocole

- Un protocole P est *correct* si :
  - ⋆ Il est partiellement correct.
  - ⋆ Chaque exécution admissible possède une valeur de décision.

Le résultat

Aucun protocole de consensus n'est correct.

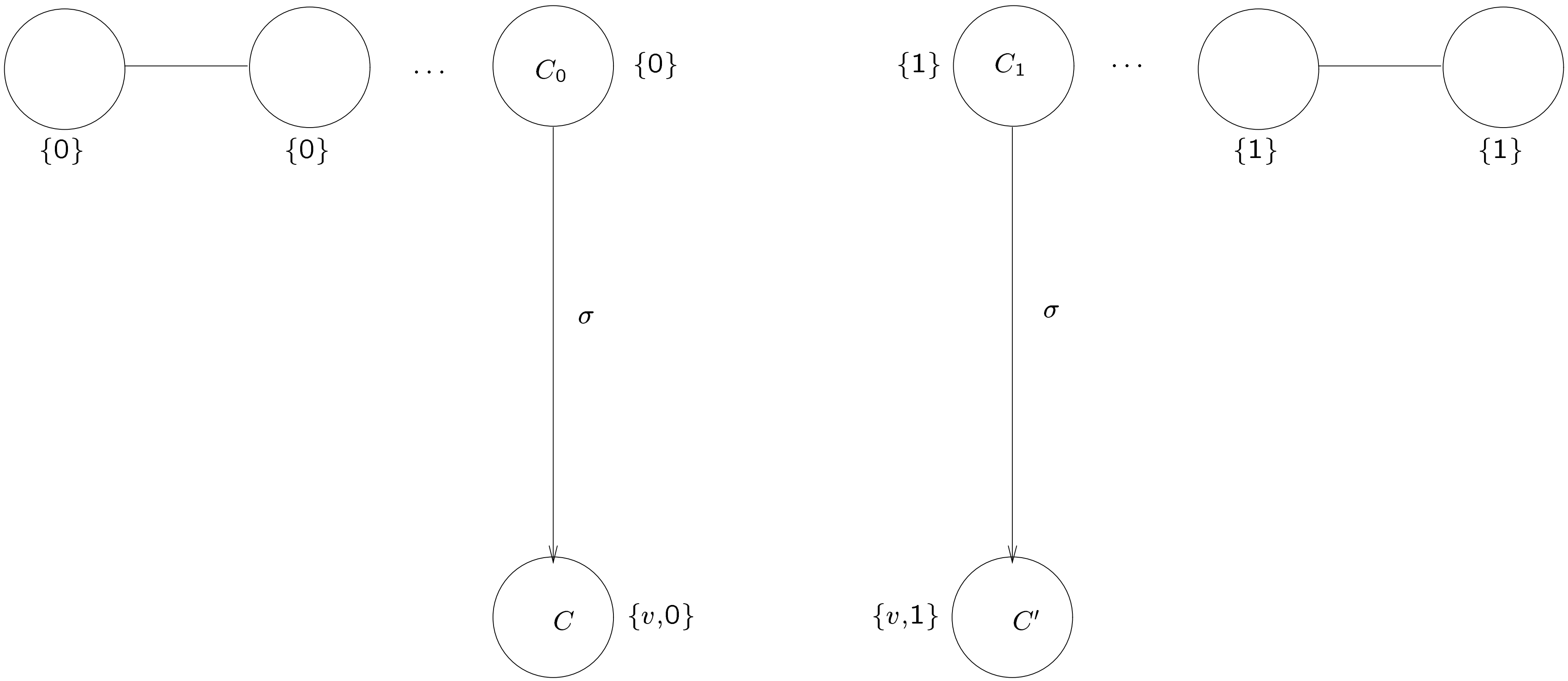
Modélisation d'un protocole

- $C = (s,\mathcal{M})$ une configuration.
- V l'ensemble des valeurs de décision atteignable depuis C.
- C est bivalente si $\mid V \mid = 2$. Sinon C est monovalente.
- Si $V = v$, C est v-valent.

Lemme 2

P possède une configuration initiale bivalente.

Démonstration du lemme 2



Lemme 3

- C une configuration bivalente de P.
- $e = (p_i, m)$ un pas applicable à C.
- $\mathcal{C} = \{E \mid \exists \sigma$ tel que $E = \sigma(C)$ et $p_i \notin \sigma\}$
- $\mathcal{D} = e(\mathcal{C})$
- Alors $\mathcal{D}$ contient une configuration bivalente.

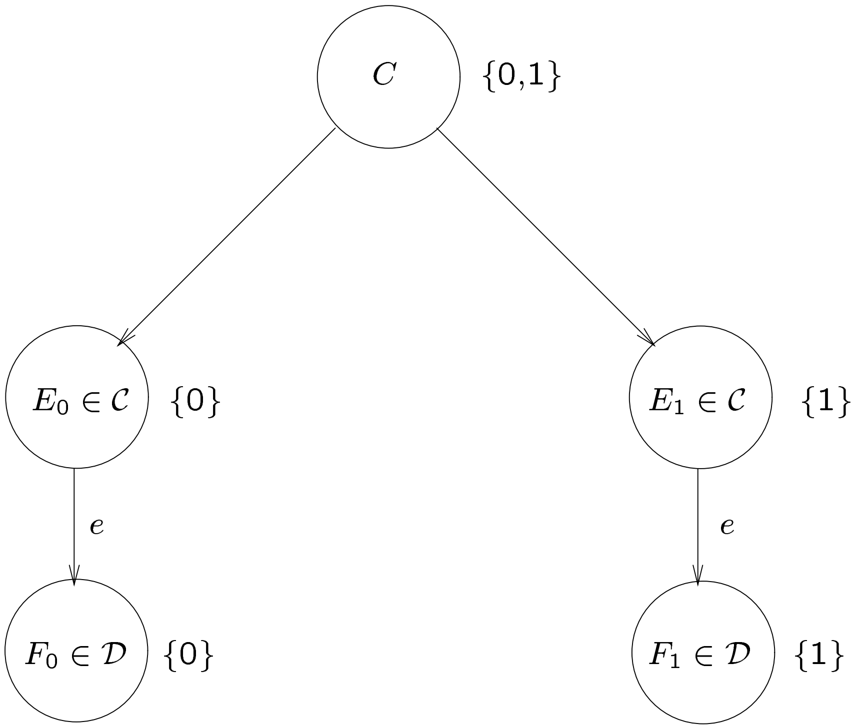
Démonstration du lemme 3

- On suppose par contradiction que $\mathcal{D}$ ne contient que des configurations monovalentes.
- Alors $\mathcal{D}$ contient des configurations 0-valentes et 1-valentes.
- Donc il existe C_0 et C_1 dans $\mathcal{C}$, telles que :
  - $\star$ $D_i = e(C_i) \in \mathcal{D}$ est i-valente.
  - $\star$ $C_1 = e'(C_0)$ avec $e' = (p', m')$.

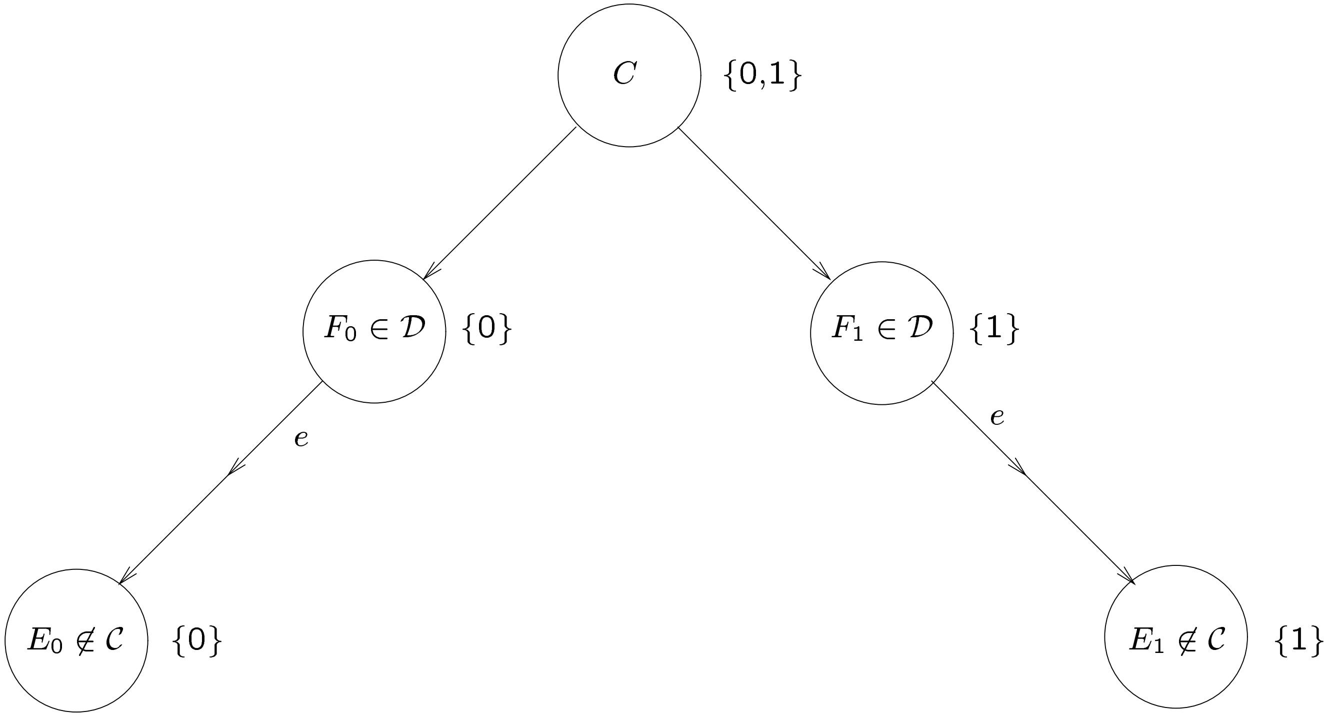
Démonstration du lemme 3

- Si $p \neq p'$ alors $D_1 = e'(D_0)$ (Lemme 1), ce qui est impossible.
- Si $p = p'$, on considère σ une séquence dans laquelle p ne prend pas de pas applicable à C_0. $A = \sigma(C_0)$. On suppose de plus que A possède une valeur de décision.
  - ⋆ σ est applicable à D_i et conduit à $E_i = \sigma(D_i)$, et E_i est i-valente.
  - ⋆ Par le lemme 1, $E_0 = e(A)$ et $E_1 = e(e'(A))$, donc A est bivalente, ce qui est impossible car A est mono-valente.
- Donc $\mathcal{D}$ contient une configuration bivalente.

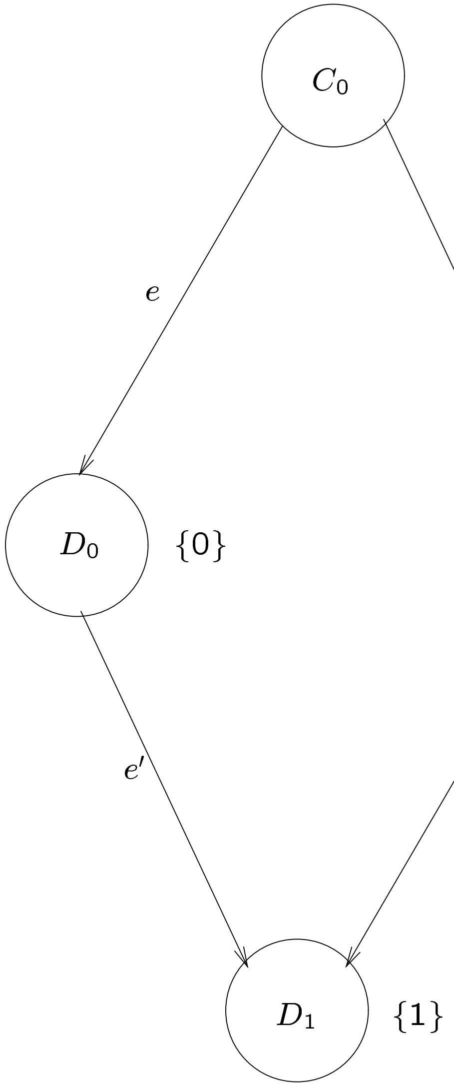
Démonstration du lemme 3



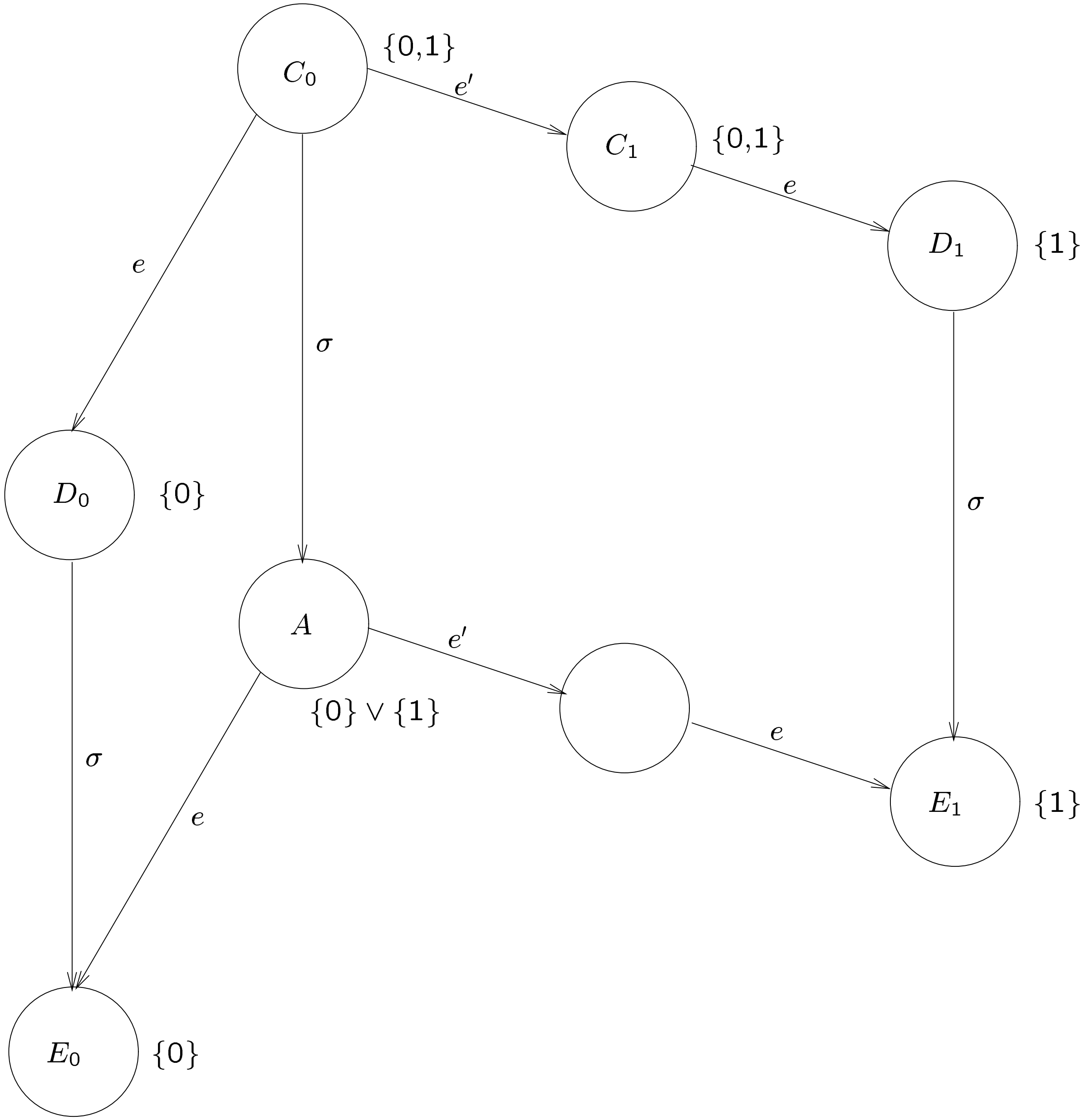
Démonstration du lemme 3



Démonstration du lemme 3



Démonstration du lemme 3



Enfin fini ...

- On peut construire une séquence admissible telle que toutes ces configurations sont bivalentes. Donc l'algorithme ne décide pas. Il n'est pas correct.

- On commence par une configuration initiale bivalente $C_0 = (s_0, \emptyset)$ (Lemme 2). Soit p un processus quelconque. Il existe une séquence σ qui se termine par $(p, \emptyset)$, et qui conduit à $C = (s, \mathcal{M})$ bivalente.

- On réitère avec C, un autre processus p, et $m \in \mathcal{M}$.