

SDR

Estampilles

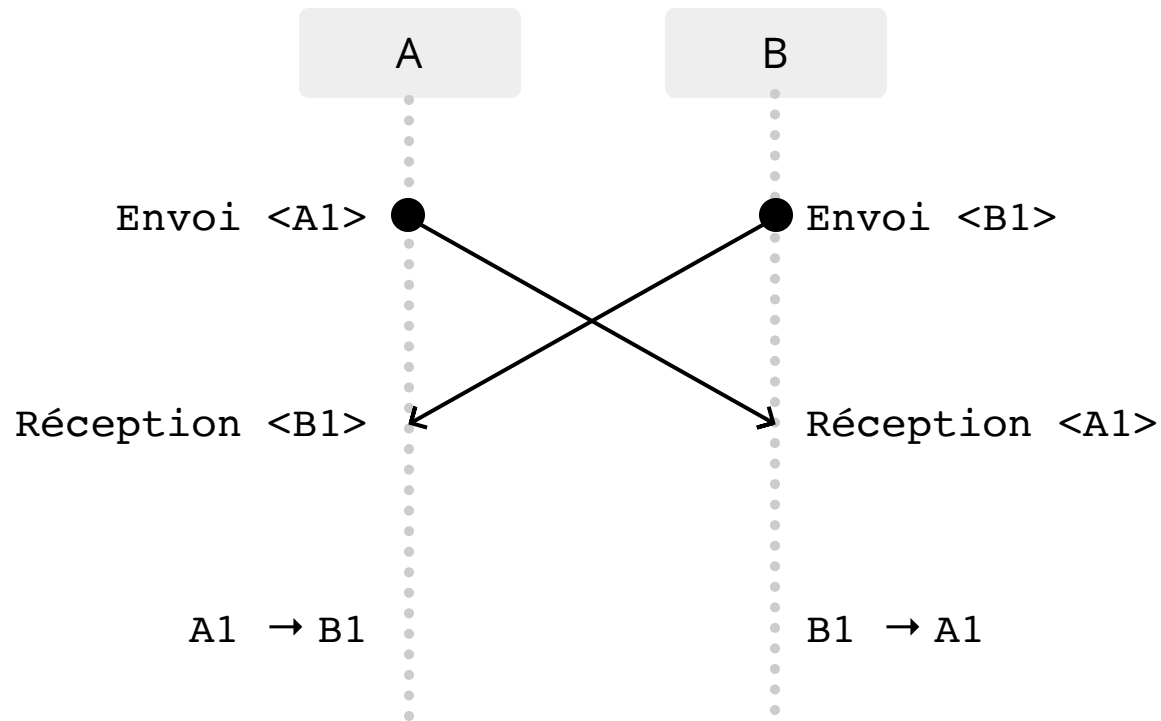
Horloges logiques

Olivier Lemer

↓ **Le problème des horloges**

Le problème des horloges

L'ordre des évènements n'apparaît pas le même partout.



La synchronisation comme solution ?

Créer une horloge commune ?

La synchronisation comme solution ?

Créer une horloge commune ?

Problème : Horloges à quartz dérivent différemment selon la machine.

La synchronisation comme solution ?

Créer une horloge commune ?

Problème : Horloges à quartz dérivent différemment selon la machine.

Solutions : protocoles de mise à l'heure.

- Network Time Protocol : longue distance, ms precision.
- Precision Time Protocol : courte distance, μ s precision.

La synchronisation comme solution ?

Créer une horloge commune ?

Problème : Horloges à quartz dérivent différemment selon la machine.

Solutions : protocoles de mise à l'heure.

- Network Time Protocol : longue distance, ms precision.
- Precision Time Protocol : courte distance, μ s precision.

Pas de solution permettant une horloge globale parfaite : toujours soumis aux délais et variabilités du réseau.

↓ **Nos besoins**

Résolution par requirements

Quels sont les besoins ?

Résolution par requirements

Quels sont les besoins ?

On veut connaître l'heure exacte de chaque évènement.

Résolution par requirements

Quels sont les besoins ?

On veut connaître l'heure ~~exacte~~ de chaque évènement.
relative

Résolution par requirements

Quels sont les besoins ?

On veut connaître ~~l'heure exacte de chaque évènement.~~

relative

l'ordre des évènements.

Résolution par requirements

Quels sont les besoins ?

On veut connaître ~~l'heure exacte de chaque évènement.~~

relative

l'**ordre** des évènements.

?

Résolution par requirements

Quels sont les besoins ?

On veut connaître ~~l'heure exacte de chaque évènement.~~

relative

l'**ordre** des évènements.

?

Qu'est-ce qu'un évènement ?

Résolution par requirements

Quels sont les besoins ?

On veut connaître ~~l'heure exacte de chaque évènement.~~

relative

l'**ordre** des évènements.

?

Qu'est-ce qu'un évènement ?

- Une action sur une machine (*exécution d'une ou plusieurs instructions*)

Résolution par requirements

Quels sont les besoins ?

On veut connaître ~~l'heure exacte de chaque évènement.~~

relative

l'**ordre** des évènements.

?

Qu'est-ce qu'un évènement ?

- Une action sur une machine (*exécution d'une ou plusieurs instructions*)
- Emission d'un message

Résolution par requirements

Quels sont les besoins ?

On veut connaître ~~l'heure exacte de chaque évènement.~~

relative

l'**ordre** des évènements.

?

Qu'est-ce qu'un évènement ?

- Une action sur une machine (*exécution d'une ou plusieurs instructions*)
- Emission d'un message
- Réception d'un message

Résolution par requirements

Quels sont les besoins ?

On veut connaître ~~l'heure exacte de chaque évènement.~~

relative

l'ordre des évènements.

?

?

Qu'est-ce qu'un évènement ?

- Une action sur une machine (*exécution d'une ou plusieurs instructions*)
- Emission d'un message
- Réception d'un message

Qu'est-ce qu'un ordre d'évènements ? *Quand est-ce que $E1 \rightarrow E2$?*

Résolution par requirements

Quels sont les besoins ?

On veut connaître ~~l'heure exacte de chaque évènement.~~

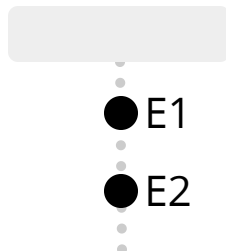
relative
l'ordre des évènements.
? ?

Qu'est-ce qu'un évènement ?

- Une action sur une machine (*exécution d'une ou plusieurs instructions*)
- Emission d'un message
- Réception d'un message

Qu'est-ce qu'un ordre d'évènements ? *Quand est-ce que $E1 \rightarrow E2$?*

- Si E1 et E2 sont sur la même machine
et E1 arrive avant E2.



Résolution par requirements

Quels sont les besoins ?

On veut connaître ~~l'heure exacte de chaque évènement.~~

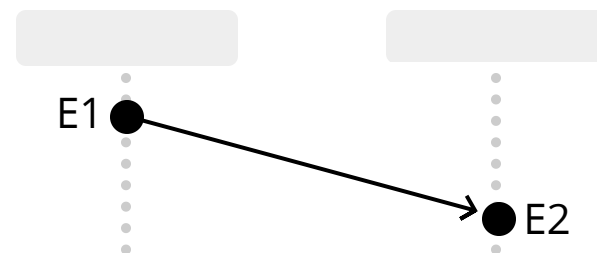
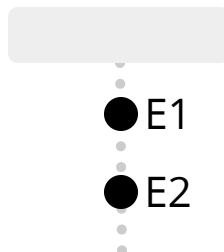
relative
l'ordre des évènements.
? ?

Qu'est-ce qu'un évènement ?

- Une action sur une machine (*exécution d'une ou plusieurs instructions*)
- Emission d'un message
- Réception d'un message

Qu'est-ce qu'un ordre d'évènements ? *Quand est-ce que $E1 \rightarrow E2$?*

- Si E1 et E2 sont sur la même machine et E1 arrive avant E2.
- Si E1 est l'émission d'un message et E2 sa réception.



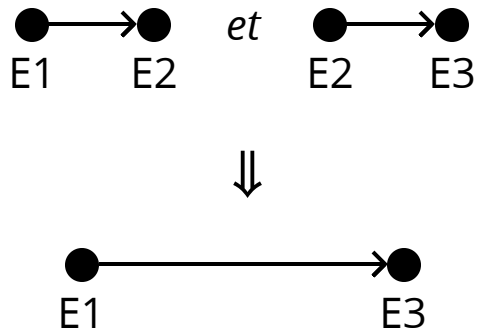
Une relation d'ordre

Propriétés

Une relation d'ordre

Propriétés

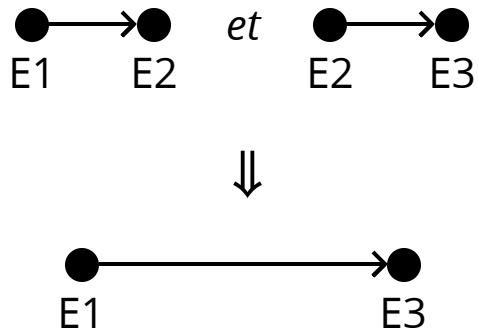
Transitif



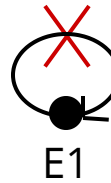
Une relation d'ordre

Propriétés

Transitif



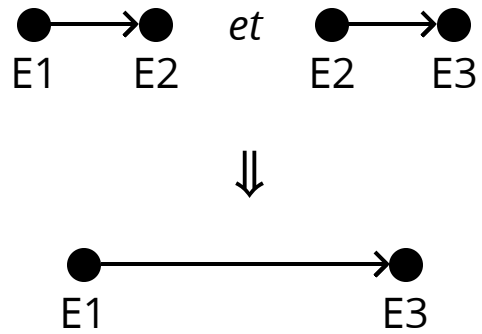
Anti-réflexif



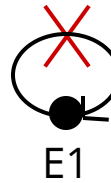
Une relation d'ordre

Propriétés

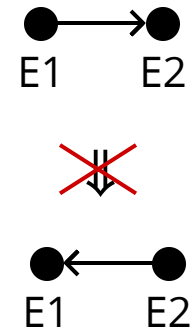
Transitif



Anti-réflexif



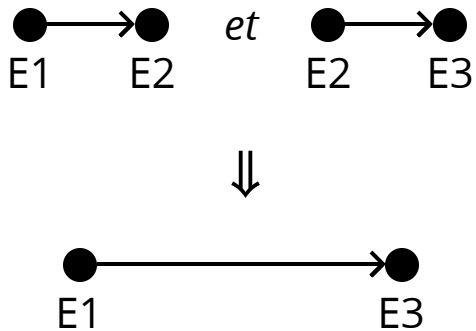
Anti-symétrique



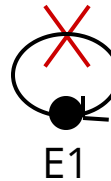
Une relation d'ordre

Propriétés

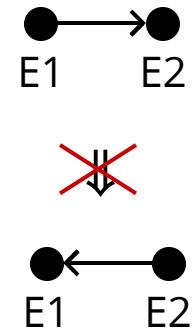
Transitif



Anti-réflexif



Anti-symétrique



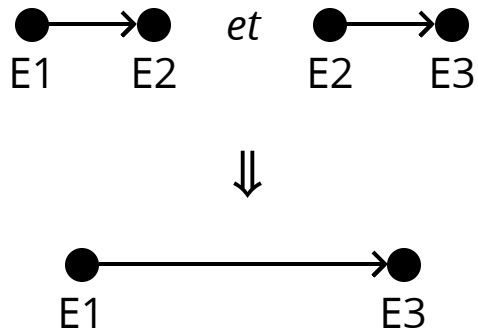
Ordre Partiel ou Total ?

(Partiel s'il existe deux évènements $E1$ et $E2$ tels que ni $E1 \rightarrow E2$, ni $E2 \rightarrow E1$.)

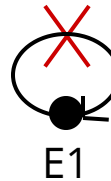
Une relation d'ordre

Propriétés

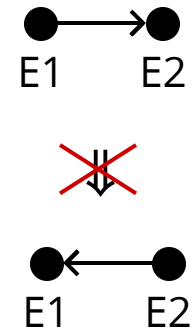
Transitif



Anti-réflexif



Anti-symétrique



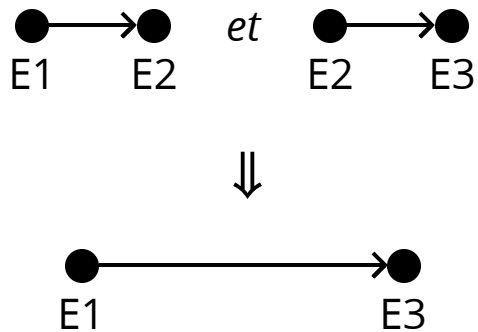
Ordre Partiel

(i.e. il existe deux évènements $E1$ et $E2$ tels que ni $E1 \rightarrow E2$, ni $E2 \rightarrow E1$.)

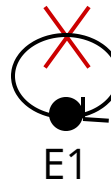
Une relation d'ordre

Propriétés

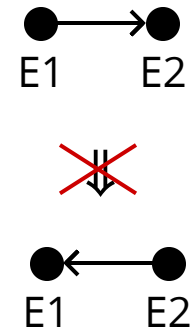
Transitif



Anti-réflexif

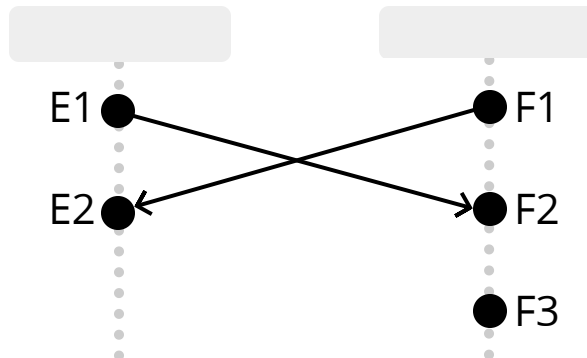


Anti-symétrique



Ordre Partiel

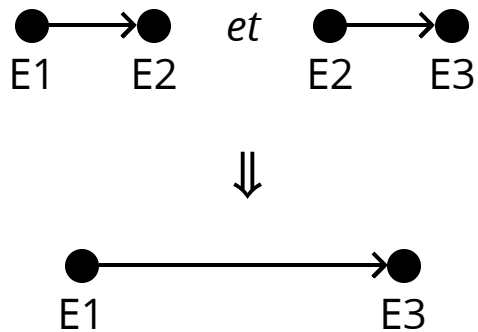
(i.e. il existe deux évènements $E1$ et $E2$ tels que ni $E1 \rightarrow E2$, ni $E2 \rightarrow E1$.)



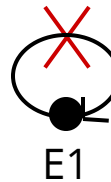
Une relation d'ordre

Propriétés

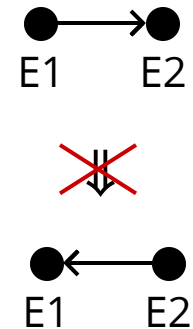
Transitif



Anti-réflexif

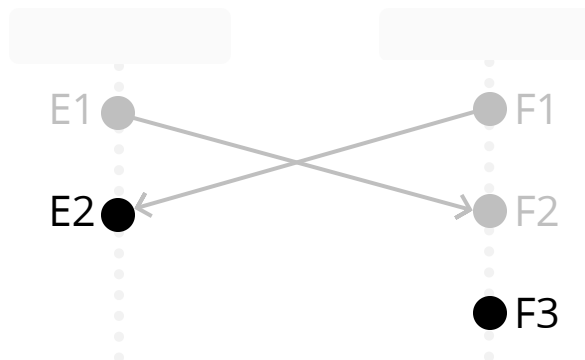


Anti-symétrique



Ordre Partiel

(i.e. il existe deux évènements $E1$ et $E2$ tels que ni $E1 \rightarrow E2$, ni $E2 \rightarrow E1$.)



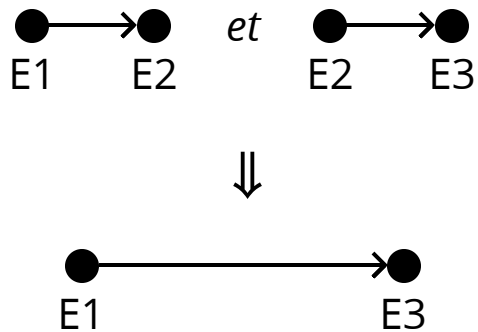
$E2 \rightarrow F3 ?$

$F3 \rightarrow E2 ?$

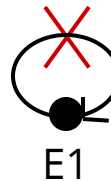
Une relation d'ordre

Propriétés

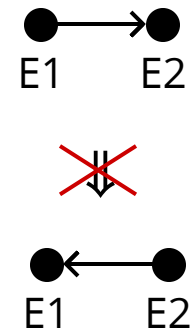
Transitif



Anti-réflexif

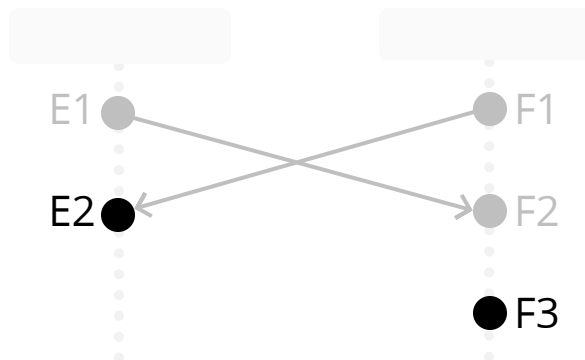


Anti-symétrique



Ordre Partiel

(i.e. il existe deux évènements $E1$ et $E2$ tels que ni $E1 \rightarrow E2$, ni $E2 \rightarrow E1$.)



$E2 \rightarrow F3$?

$F3 \rightarrow E2$?

Les deux sont possibles.

↓ Horloge logique

Une horloge *logique*

Représenter l'ordre partiel des évènements.

Une horloge *logique*

Représenter l'ordre partiel des évènements.

On veut :

Une horloge *logique*

Représenter l'ordre partiel des évènements.

On veut : une fonction H telle que :

$$E1 \rightarrow E2$$



$$H(E1) < H(E2)$$

Une horloge *logique*

Représenter l'ordre partiel des évènements.

On veut : une fonction H telle que :

$$\begin{array}{c} E1 \rightarrow E2 \\ \Downarrow \\ H(E1) < H(E2) \end{array}$$

Idée 1 : horloge logique = horloge locale ?

(i.e. $H(E)$ = heure sur la machine sur laquelle l'évènement a eu lieu)

Une horloge *logique*

Représenter l'ordre partiel des évènements.

On veut : une fonction H telle que :

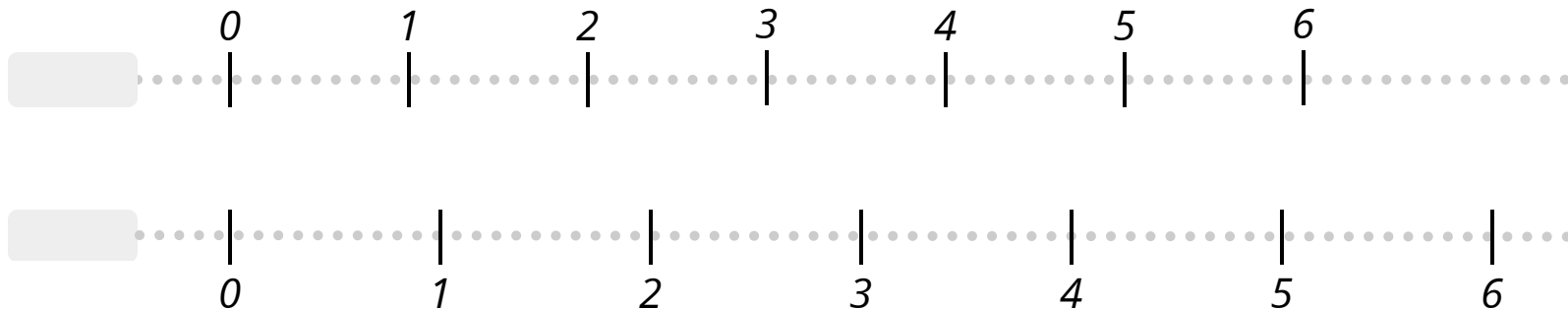
$$E1 \rightarrow E2$$

$\Downarrow$

$$H(E1) < H(E2)$$

Idée 1 : horloge logique = horloge locale ?

(i.e. $H(E)$ = heure sur la machine sur laquelle l'évènement a eu lieu)



Une horloge *logique*

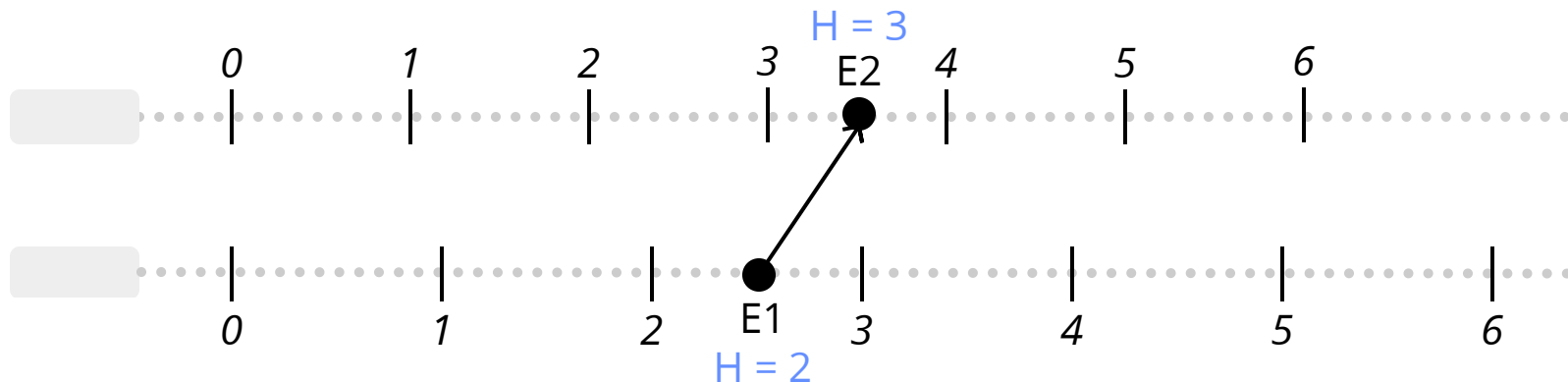
Représenter l'ordre partiel des évènements.

On veut : une fonction H telle que :

$$\begin{array}{c} E1 \rightarrow E2 \\ \Downarrow \\ H(E1) < H(E2) \end{array}$$

Idée 1 : horloge logique = horloge locale ?

(i.e. $H(E)$ = heure sur la machine sur laquelle l'évènement a eu lieu)



Une horloge *logique*

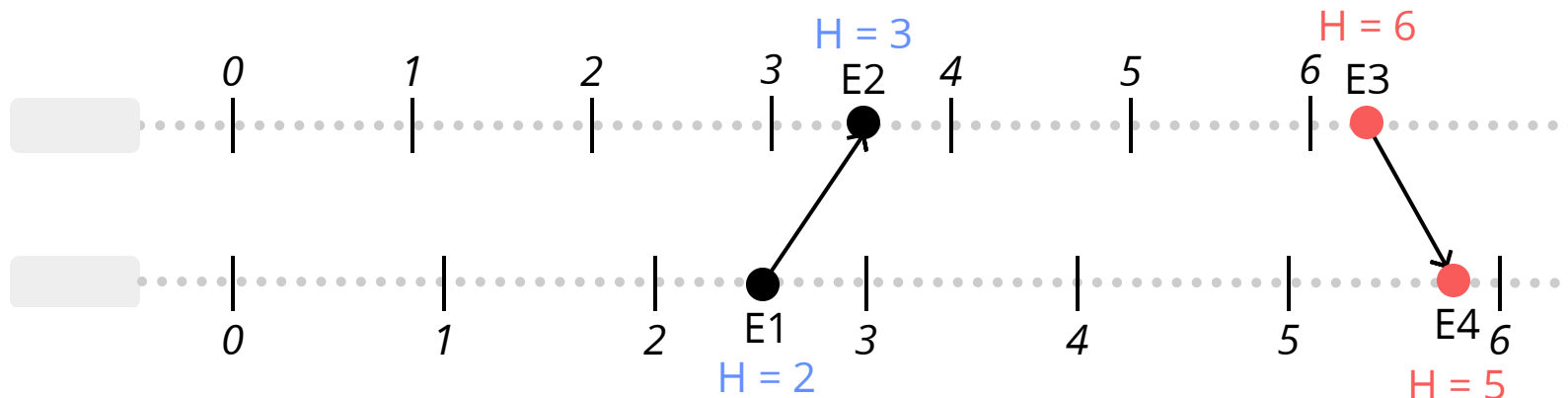
Représenter l'ordre partiel des évènements.

On veut : une fonction H telle que :

$$\begin{aligned} E1 &\rightarrow E2 \\ \Downarrow \\ H(E1) &< H(E2) \end{aligned}$$

Idée 1 : horloge logique = horloge locale ?

(i.e. $H(E)$ = heure sur la machine sur laquelle l'évènement a eu lieu)



Une horloge *logique*

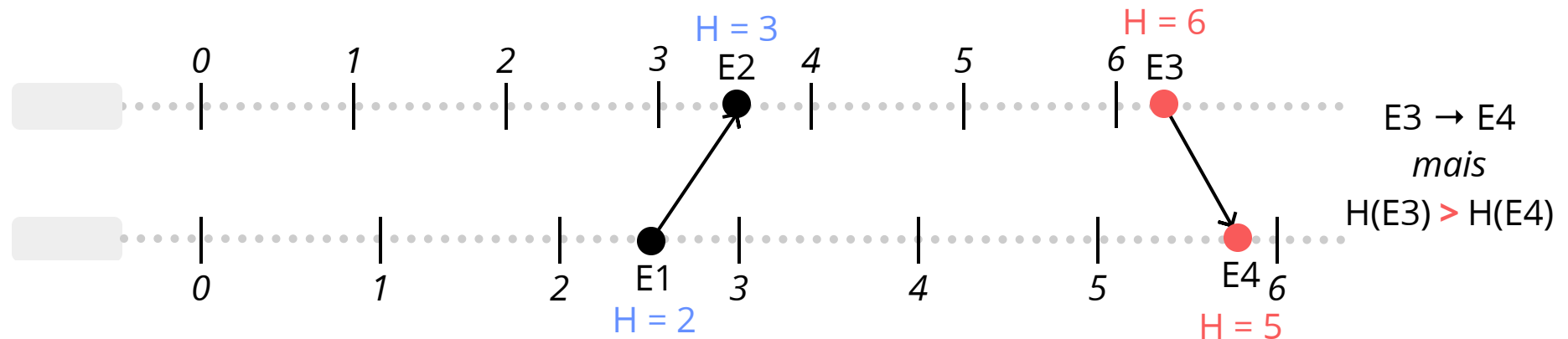
Représenter l'ordre partiel des évènements.

On veut : une fonction H telle que :

$$\begin{aligned} E1 &\rightarrow E2 \\ \Downarrow \\ H(E1) &< H(E2) \end{aligned}$$

Idée 1 : horloge logique = horloge locale ?

(i.e. $H(E)$ = heure sur la machine sur laquelle l'évènement a eu lieu)



L'horloge logique *de Lamport*

L'idée : les timestamps (*ou estampilles*)



L'horloge logique *de Lamport*

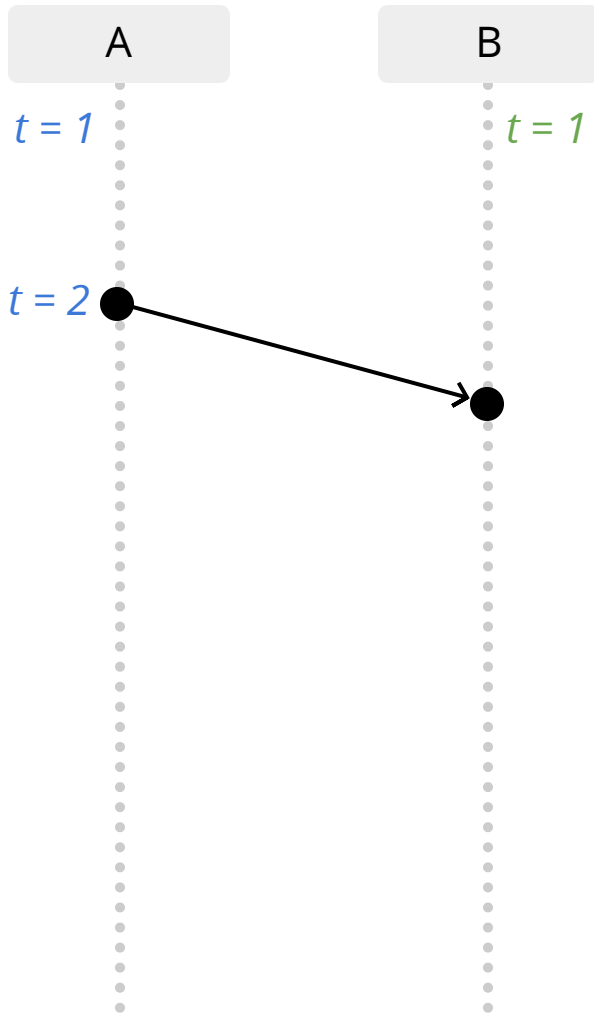
L'idée : les timestamps (*ou estampilles*)



- Chaque site maintient un numéro.

L'horloge logique *de Lamport*

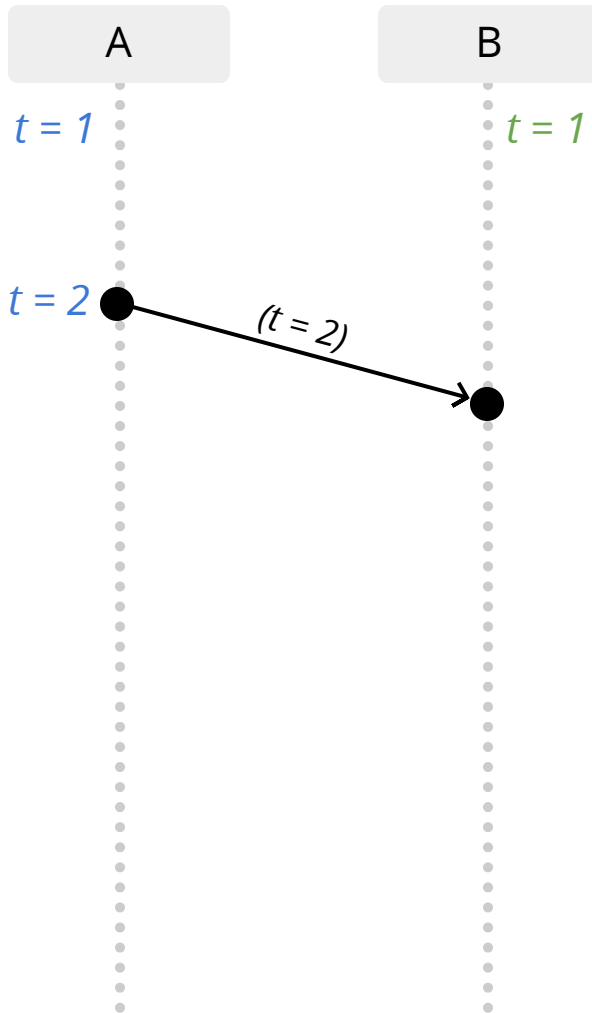
L'idée : les timestamps (*ou estampilles*)



- Chaque site maintient un numéro.
- À chaque événement *local*, le timestamp est incrémenté.

L'horloge logique *de Lamport*

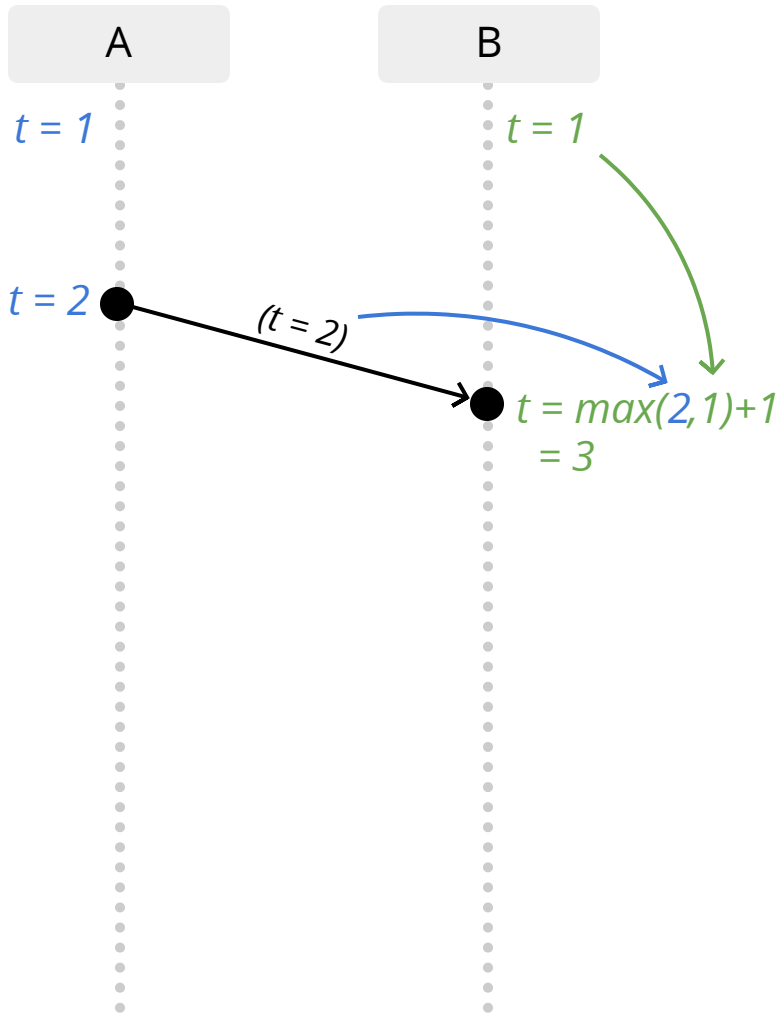
L'idée : les timestamps (*ou estampilles*)



- Chaque site maintient un numéro.
- À chaque événement *local*, le timestamp est incrémenté.
- Le timestamp est envoyé avec chaque nouveau message.

L'horloge logique *de Lamport*

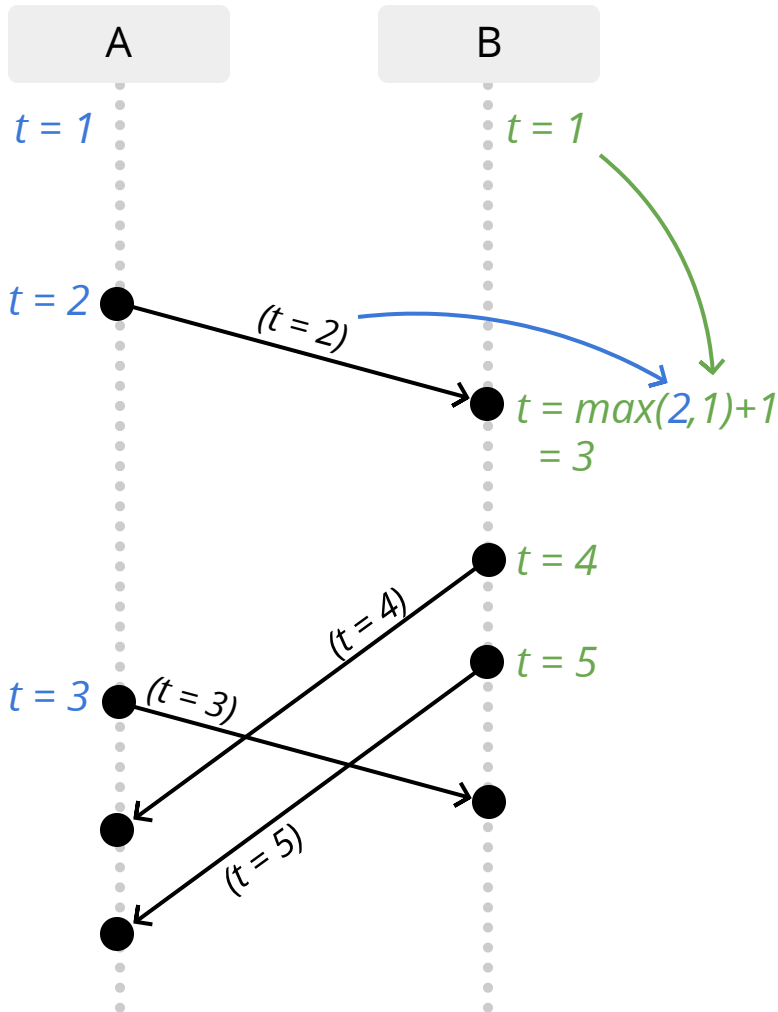
L'idée : les timestamps (*ou estampilles*)



- Chaque site maintient un numéro.
- À chaque événement *local*, le timestamp est incrémenté.
- Le timestamp est envoyé avec chaque nouveau message.
- À la réception, le timestamp le plus grand entre local et reçu est utilisé *et incrémenté*.

L'horloge logique *de Lamport*

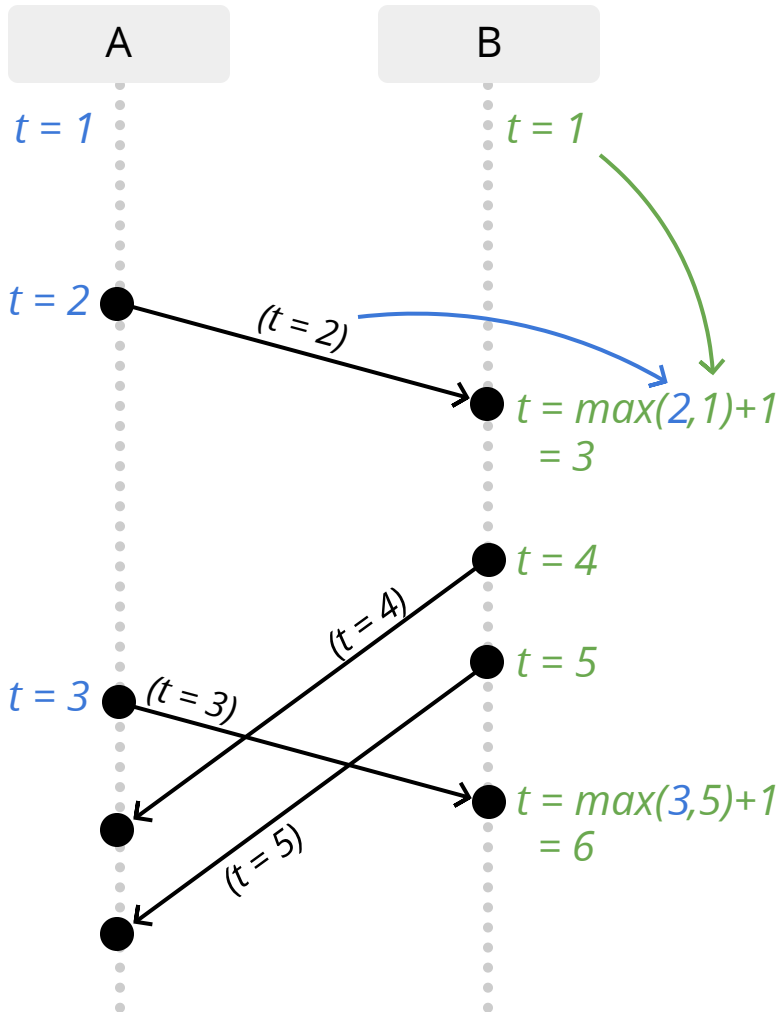
L'idée : les timestamps (*ou estampilles*)



- Chaque site maintient un numéro.
- À chaque événement *local*, le timestamp est incrémenté.
- Le timestamp est envoyé avec chaque nouveau message.
- À la réception, le timestamp le plus grand entre local et reçu est utilisé *et incrémenté*.

L'horloge logique de Lamport

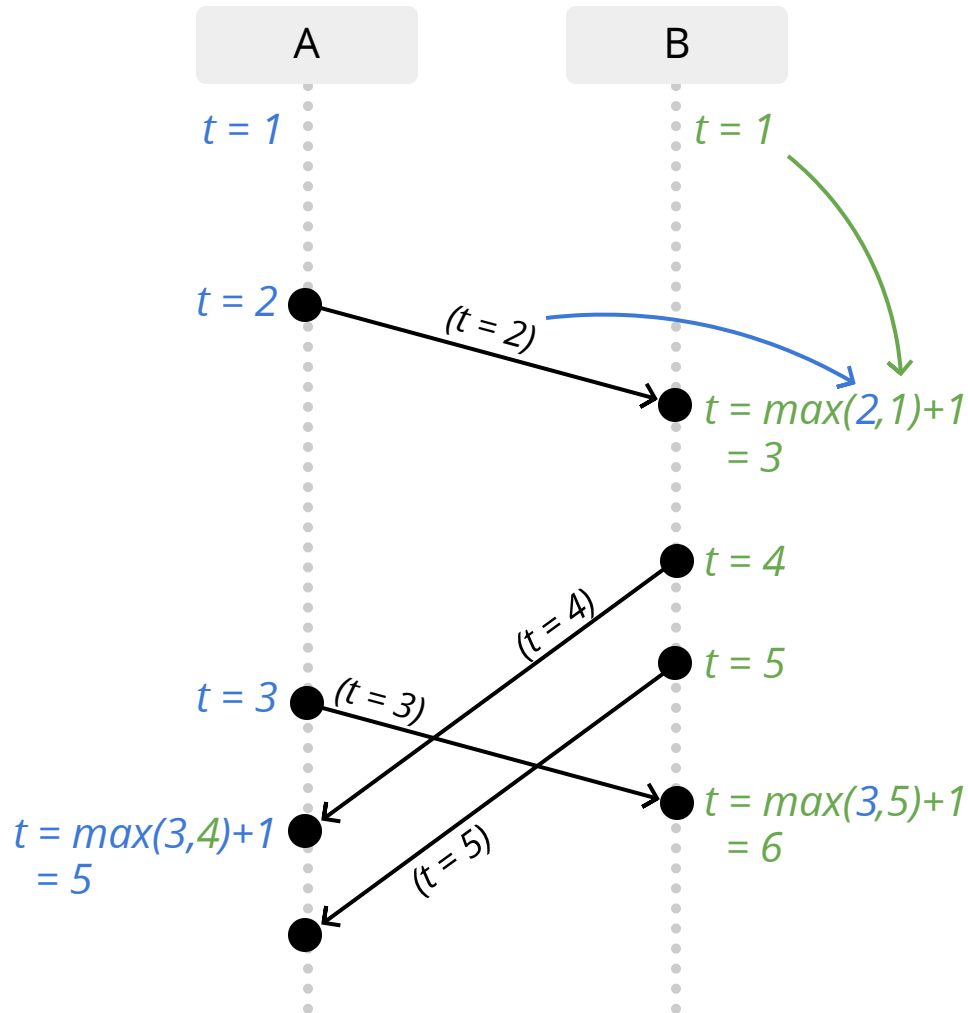
L'idée : les timestamps (ou estampilles)



- Chaque site maintient un numéro.
- À chaque événement *local*, le timestamp est incrémenté.
- Le timestamp est envoyé avec chaque nouveau message.
- À la réception, le timestamp le plus grand entre local et reçu est utilisé *et incrémenté*.

L'horloge logique de Lamport

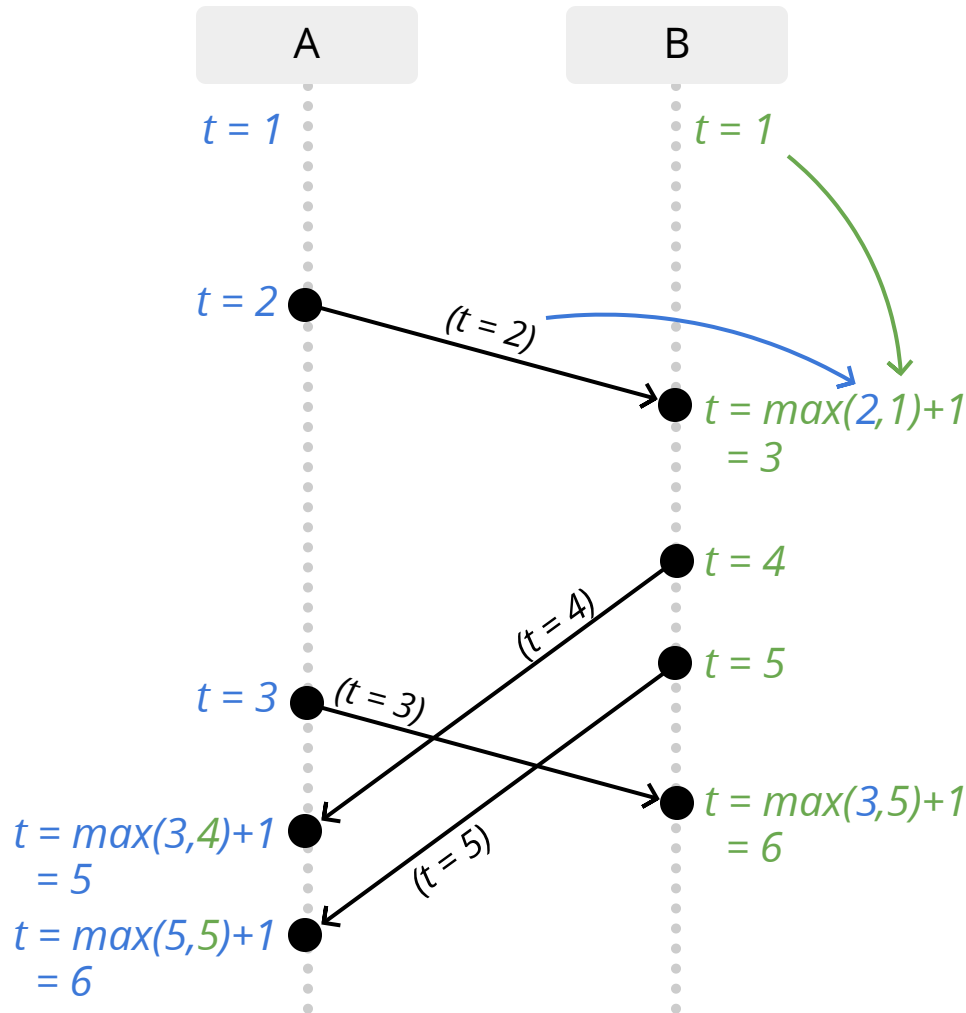
L'idée : les timestamps (ou estampilles)



- Chaque site maintient un numéro.
- À chaque événement *local*, le timestamp est incrémenté.
- Le timestamp est envoyé avec chaque nouveau message.
- À la réception, le timestamp le plus grand entre local et reçu est utilisé *et incrémenté*.

L'horloge logique de Lamport

L'idée : les timestamps (ou estampilles)



- Chaque site maintient un numéro.
- À chaque événement *local*, le timestamp est incrémenté.
- Le timestamp est envoyé avec chaque nouveau message.
- À la réception, le timestamp le plus grand entre local et reçu est utilisé *et incrémenté*.

L'horloge logique *de Lamport*

Vérification des besoins

L'horloge logique *de Lamport*

Vérification des besoins

On voulait : une fonction H telle que :

$$\begin{array}{c} E1 \rightarrow E2 \\ \Downarrow \\ H(E1) < H(E2) \end{array}$$

L'horloge logique *de Lamport*

Vérification des besoins

On voulait : une fonction H telle que :

$$\begin{array}{c} E1 \rightarrow E2 \\ \Downarrow \\ H(E1) < H(E2) \end{array}$$

On avait défini $E1 \rightarrow E2$:

- Si $E1$ et $E2$ sont sur la même machine et $E1$ arrive avant $E2$.
- Si $E1$ est l'émission d'un message et $E2$ sa réception.

L'horloge logique *de Lamport*

Vérification des besoins

On voulait : une fonction H telle que :

$$\begin{array}{c} E1 \rightarrow E2 \\ \Downarrow \\ H(E1) < H(E2) \end{array}$$

On avait défini $E1 \rightarrow E2$:

- Si $E1$ et $E2$ sont sur la même machine et $E1$ arrive avant $E2$.
- Si $E1$ est l'émission d'un message et $E2$ sa réception.

$$\begin{array}{c} \Downarrow \\ H(E1) < H(E2) \end{array}$$

?

$$\begin{array}{c} \Downarrow \\ H(E1) < H(E2) \end{array}$$

?

L'horloge logique *de Lamport*

Vérification des besoins

On voulait : une fonction H telle que :

$$\begin{array}{c} E1 \rightarrow E2 \\ \Downarrow \\ H(E1) < H(E2) \end{array}$$

On avait défini $E1 \rightarrow E2$:

- Si $E1$ et $E2$ sont sur la même machine et $E1$ arrive avant $E2$.
- Si $E1$ est l'émission d'un message et $E2$ sa réception.

$$\begin{array}{c} \Downarrow \\ H(E1) < H(E2) \end{array}$$



$$\begin{array}{c} \Downarrow \\ H(E1) < H(E2) \end{array}$$

?

L'horloge logique *de Lamport*

Vérification des besoins

On voulait : une fonction H telle que :

$$\begin{array}{c} E1 \rightarrow E2 \\ \Downarrow \\ H(E1) < H(E2) \end{array}$$

On avait défini $E1 \rightarrow E2$:

- Si $E1$ et $E2$ sont sur la même machine et $E1$ arrive avant $E2$.
- Si $E1$ est l'émission d'un message et $E2$ sa réception.

$$\begin{array}{c} \Downarrow \\ H(E1) < H(E2) \end{array}$$



$$\begin{array}{c} \Downarrow \\ H(E1) < H(E2) \end{array}$$



L'horloge logique *de Lamport*

Vérification des besoins - un ordre total ?

On voulait : une fonction H telle que :

$$\begin{array}{c} E1 \rightarrow E2 \\ \Downarrow \\ H(E1) < H(E2) \end{array}$$

Remarque :

$$\begin{array}{c} E1 \rightarrow E2 \\ \uparrow\uparrow \\ H(E1) < H(E2) \end{array}$$

L'horloge logique *de Lamport*

Vérification des besoins - un ordre total ?

On voulait : une fonction H telle que :

$$\begin{array}{c} E1 \rightarrow E2 \\ \Downarrow \\ H(E1) < H(E2) \end{array}$$

Remarque :

$$\begin{array}{c} E1 \rightarrow E2 \\ \nRightarrow \\ H(E1) < H(E2) \end{array}$$

*On projette un **ordre partiel** sur un **ordre total**.*

Autrement dit, si $E1$ et $E2$ n'ont pas de relation, H leur en donne une.

L'horloge logique *de Lamport*

Vérification des besoins - Un ordre total *strict* ?

L'horloge logique *de Lamport*

Vérification des besoins - Un ordre total *strict* ?

Remarque : On peut avoir $H(E1) = H(E2)$.

L'horloge logique *de Lamport*

Vérification des besoins - Un ordre total *strict* ?

Remarque : On peut avoir $H(E1) = H(E2)$.
*Ce n'est pas un ordre **strict**.*

L'horloge logique *de Lamport*

Vérification des besoins - Un ordre total *strict* ?

Remarque : On peut avoir $H(E1) = H(E2)$.
*Ce n'est pas un ordre **strict**.*

Complicé pour prendre des décisions, affecter des priorités, etc.

L'horloge logique *de Lamport*

Vérification des besoins - Un ordre total *strict* ?

Remarque : On peut avoir $H(E1) = H(E2)$.
*Ce n'est pas un ordre **strict**.*

Complicé pour prendre des décisions, affecter des priorités, etc.

Solution : le rendre strict

L'horloge logique *de Lamport*

Vérification des besoins - Un ordre total *strict* ?

Remarque : On peut avoir $H(E1) = H(E2)$.
*Ce n'est pas un ordre **strict**.*

Complicé pour prendre des décisions, affecter des priorités, etc.

Solution : le rendre strict , arbitrairement :
priorité à la machine à l'ID le plus petit.

L'horloge logique *de Lamport*

Vérification des besoins - Un ordre total *strict* ?

Remarque : On peut avoir $H(E1) = H(E2)$.
*Ce n'est pas un ordre **strict**.*

Complicé pour prendre des décisions, affecter des priorités, etc.

Solution : le rendre strict , arbitrairement :
priorité à la machine à l'ID le plus petit.

Si $E1$ arrive sur machine i ,
 $E2$ arrive sur machine j ,
 tels que $i < j$
 et $H(E1) = H(E2)$,
Alors $E1$ est arrivé avant $E2$.

L'horloge logique *de Lamport*

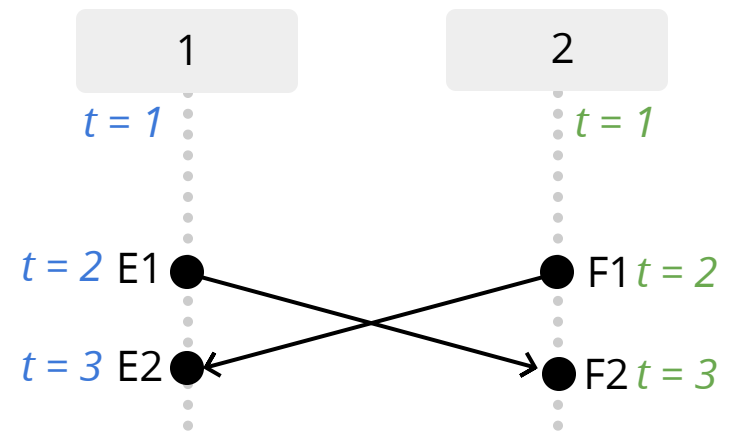
Vérification des besoins - Un ordre total *strict* ?

Remarque : On peut avoir $H(E1) = H(E2)$.
*Ce n'est pas un ordre **strict**.*

Complicé pour prendre des décisions, affecter des priorités, etc.

Solution : le rendre strict , arbitrairement :
priorité à la machine à l'ID le plus petit.

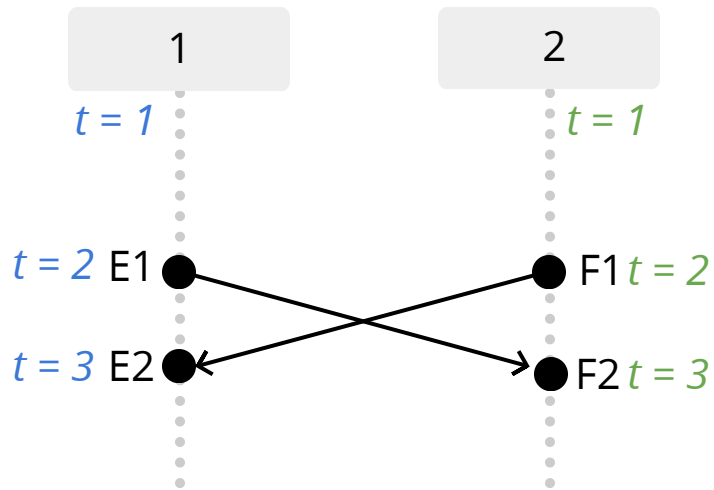
Si E1 arrive sur machine i ,
E2 arrive sur machine j ,
tels que $i < j$
et $H(E1) = H(E2)$,
Alors E1 est arrivé avant E2.



$H(E1) = H(F1)$, E1 est placé avant F1, car $1 < 2$.

L'horloge logique *de Lamport*

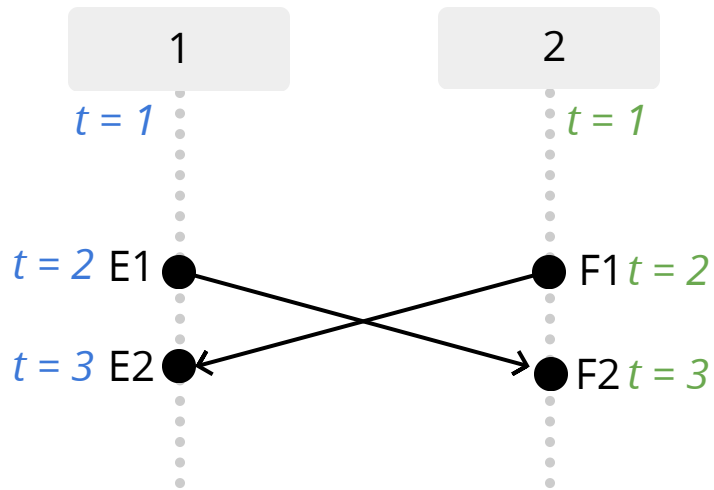
Ordre total rendu strict ; Exemple



Quel est l'ordre ?

L'horloge logique *de Lamport*

Ordre total rendu strict ; Exemple

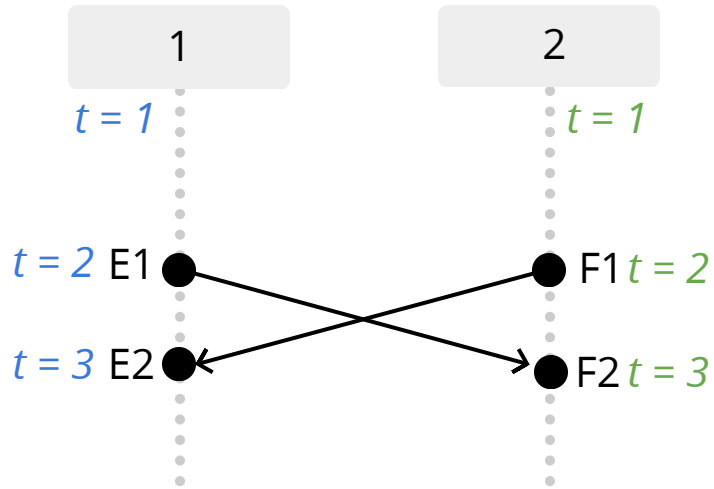


Quel est l'ordre ?

E1,

L'horloge logique *de Lamport*

Ordre total rendu strict ; Exemple

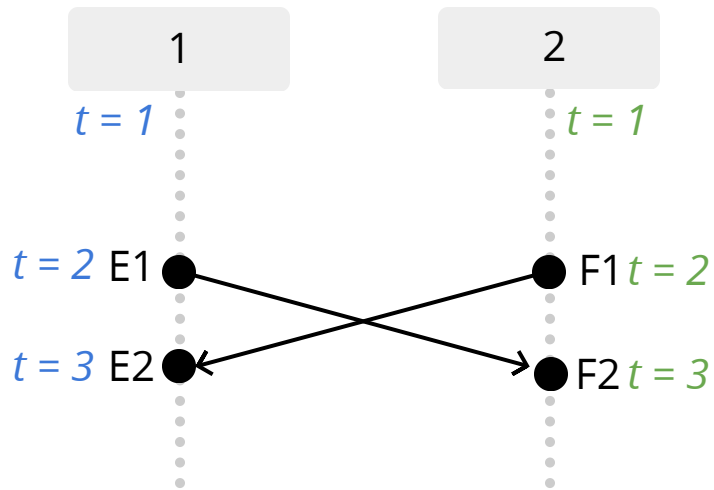


Quel est l'ordre ?

E1, F1,

L'horloge logique *de Lamport*

Ordre total rendu strict ; Exemple

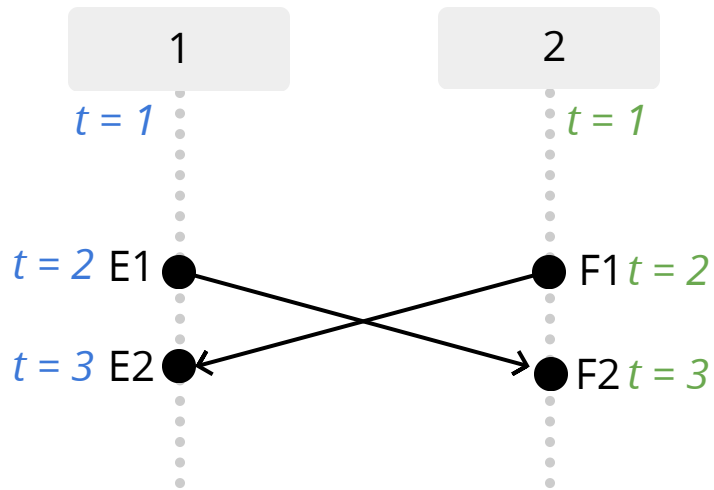


Quel est l'ordre ?

E1, F1, E2,

L'horloge logique *de Lamport*

Ordre total rendu strict ; Exemple

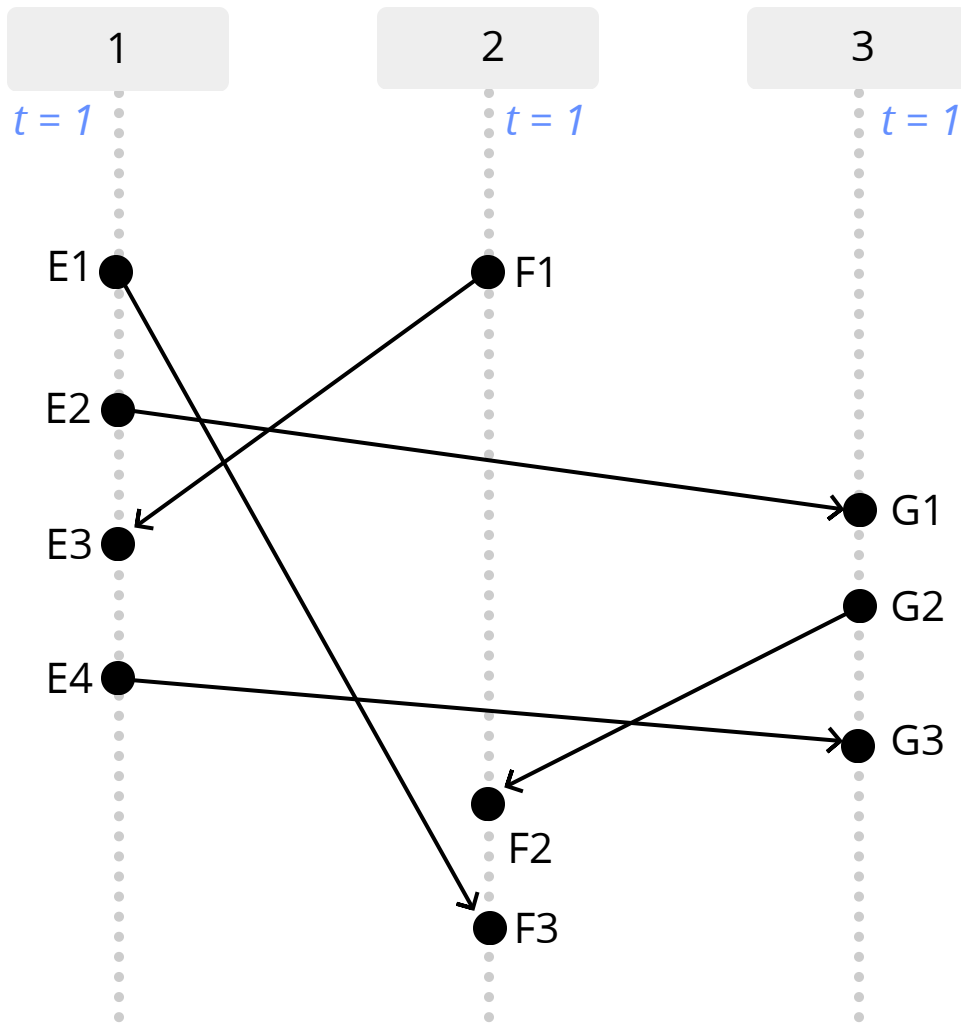


Quel est l'ordre ?

E1, F1, E2, F2.

↓ Exercices

Exercice 1 - ordre total strict



Quel est l'ordre des événements ?

Exercice 2 - file d'attente distribuée

Plusieurs distributeurs de tickets, communiquant suivant le protocole de Lamport, où une demande de ticket représente un événement.

Chaque distributeur peut délivrer un ticket, affichant

- la valeur actuelle du timestamp de Lamport*
- et le numéro du distributeur qui l'a délivré.*

Chaque client doit d'abord obtenir un ticket au distributeur le plus proche.

Celui ayant le ticket (timestamp, numéro) le plus petit peut aller au guichet.

Questions :

1. Tout personne est-elle sûre de passer un jour ?
2. L'ordre d'arrivée des personnes est-il respecté ?
3. Quels avantages par rapport à un système central délivrant des numéros aux distributeurs ?
4. Que se passe-t-il si un distributeur tombe en panne et redémarre en repartant à son dernier timestamp ?