Plan

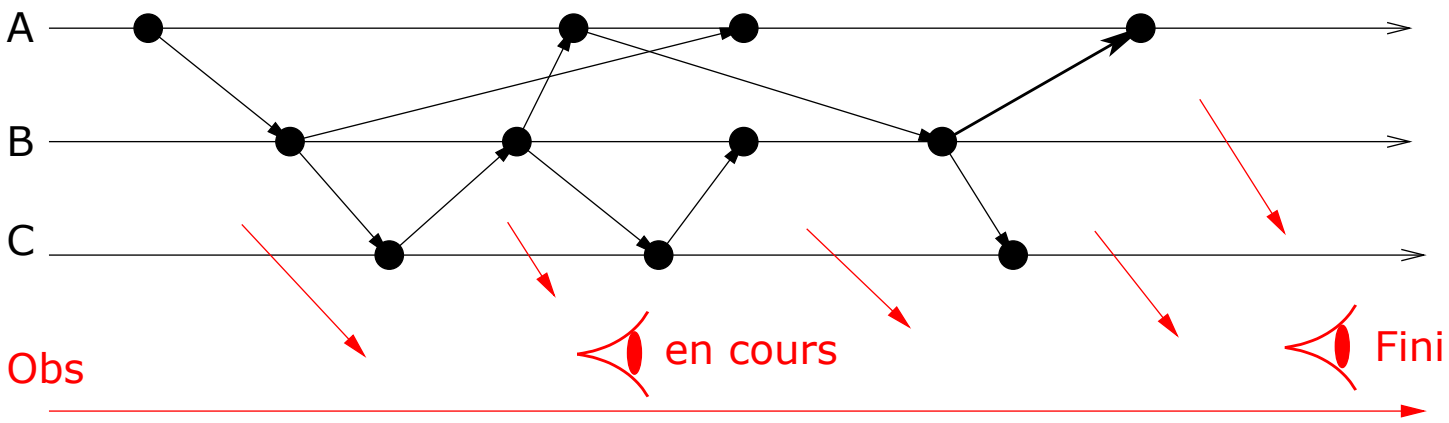
Terminaison : détecter une propriété stable ♪♪♪

Spécification

- Propriété stable à détecter :

Tous les processus sont passifs ET pas de message en transit.

- Sûreté : Pas de fausse détection :
  $Term \Rightarrow (\forall\ i\ ::\ P_i.passif \wedge EnTransit = \emptyset)$
- Vivacité : La terminaison finit par être détectée :
  $(\forall\ i\ ::\ P_i.passif \wedge EnTransit = \emptyset) \rightsquigarrow Term$



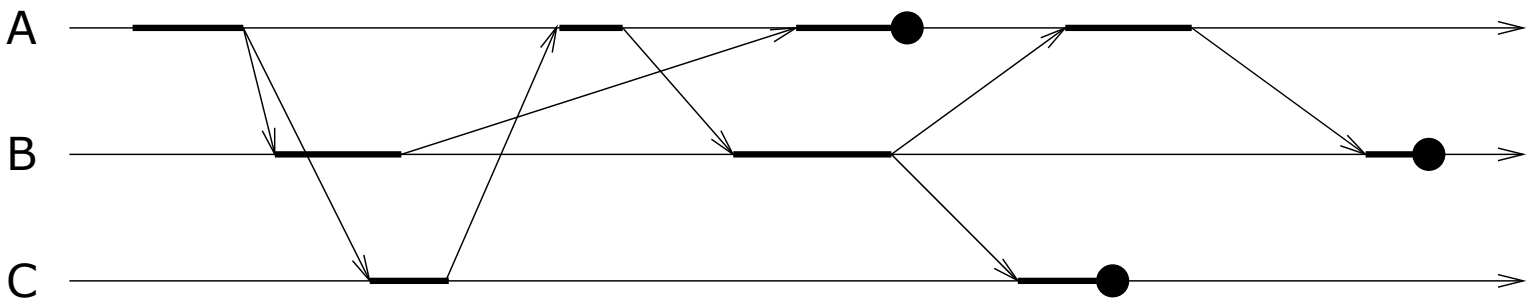
[Précis 4.2 pp.66–69]

Calcul diffusant

Définition d'un calcul diffusant

- Un processus initial émet un ou plusieurs messages
- Puis, tous les processus adoptent le même comportement :

```
loop {   /* un pas de calcul  */
         recevoir(m) ;
         traiter m ;
         envoyer 0 à N − 1 messages ;
}
```



$\Rightarrow$ Les phases actives peuvent être vues comme atomiques

Terminaison sur un anneau (Misra, 1983) ♪♪♪

Principe

Les sites sont organisés en anneau (communication FIFO depuis le précédent / vers le suivant mais à destinataire arbitraire).
Parcourir l'anneau et vérifier que tous les sites sont passifs.

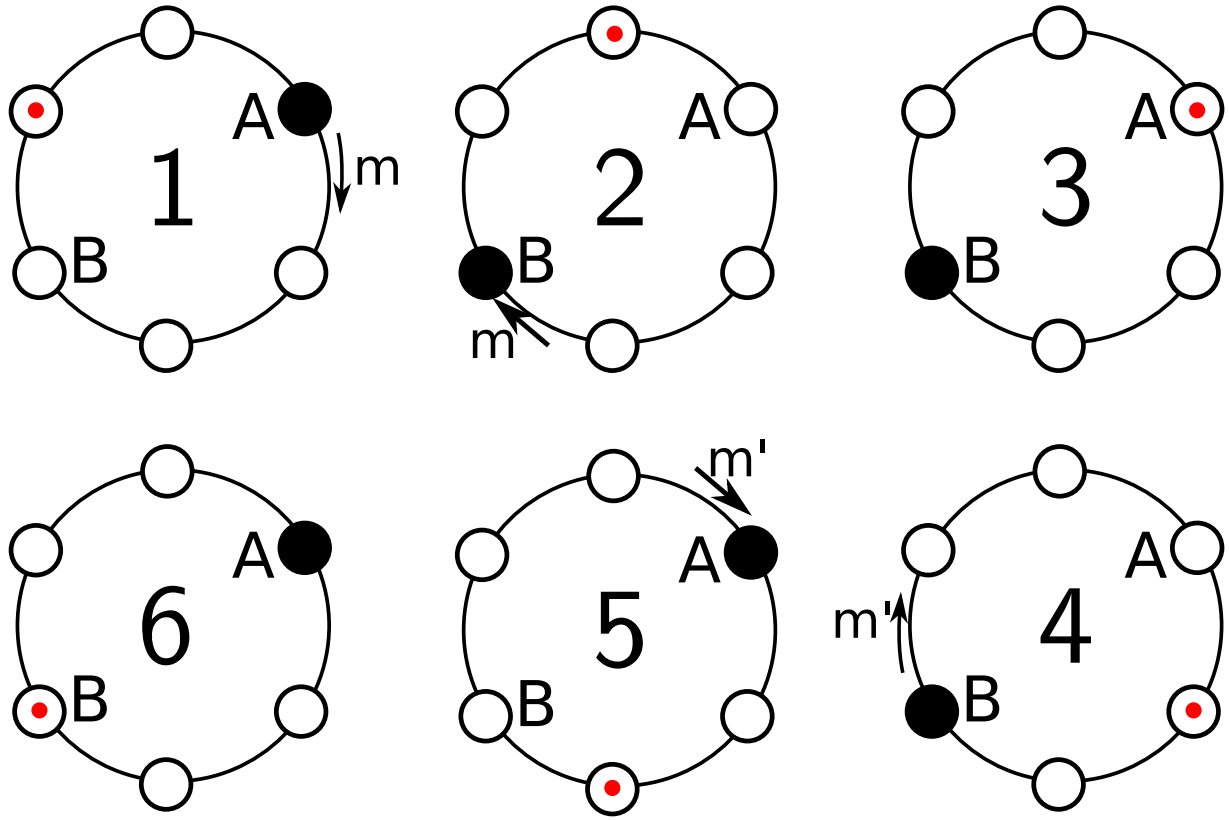
Difficulté

Un message émis avant le passage du visiteur sur le site émetteur peut être reçu après le passage du visiteur sur le site récepteur et réactiver un site trouvé passif

$\Rightarrow$ faire deux tours en vérifiant qu'aucun site n'a changé d'état entre temps

1. *Detecting Termination of Distributed Computation Using Markers*, Jayadev Misra. 2nd ACM Symposium on Principles of Distributed Computing, 1983.

Terminaison – Misra



- tous trouvés passifs, mais calcul pas terminé !
- sites A et B trouvés passifs, mais ils ont été actifs entre temps

$\Rightarrow$ faire deux tours en vérifiant qu'aucun site n'a changé d'état entre temps

Terminaison sur un anneau

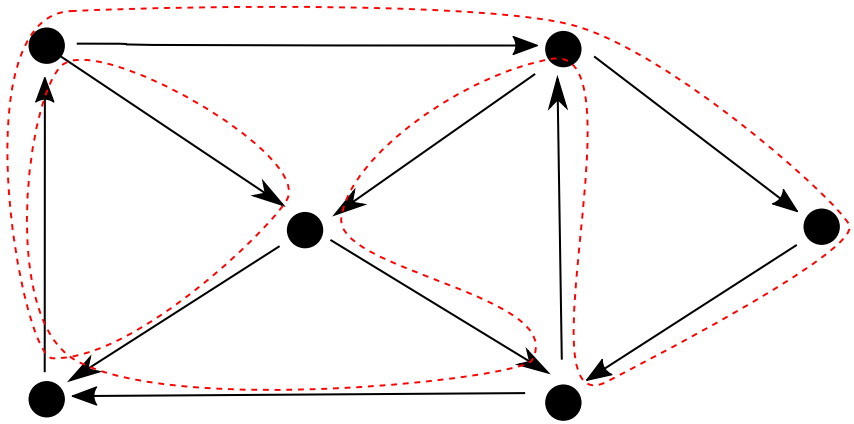
```
Process P(i : 0..N-1) {
 variables : couleur ∈ {blanc,noir}
             état ∈ {actif,passif}
             jeton ∈ {true,false}
 ⋮
  on reception message_applicatif :    // action
       couleur ← noir;
       état ← actif;
  on reception jeton(val) :            // réception jeton
       jeton ← true; nb ← val;
       if (nb = N ∧ couleur = blanc) then TERMINAISON DÉTECTÉE
  on jeton ∧ état = passif :           // envoi jeton
       if (couleur = blanc) then send jeton(nb + 1) to P_{i⊕1}
                            else send jeton(1) to P_{i⊕1}
       couleur ← blanc;
       jeton ← false;
  else :                               // attente
       état ← passif;
```

Terminaison avec un anneau logique

♪♪♪

Graphe de communication arbitraire

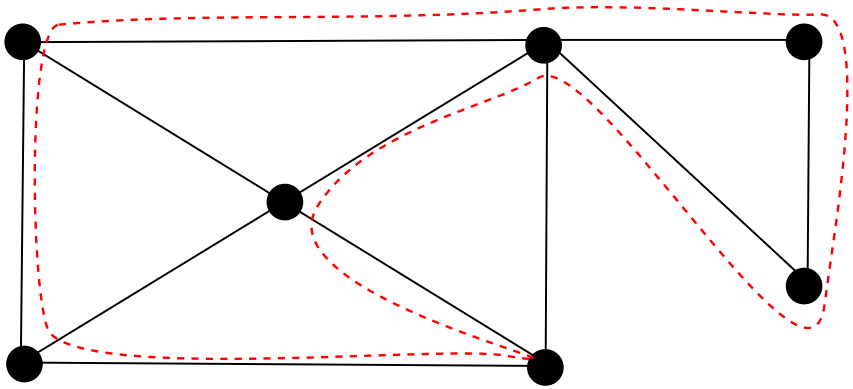
- Circuit logique contenant tous les arcs (éventuellement plusieurs fois)
- Communication FIFO : le jeton ne peut pas dépasser un message antérieur sur le même arc
- Terminaison comme précédemment, avec N = longueur du circuit



Terminaison avec un anneau logique

Graphe de communication arbitraire – avec la causalité

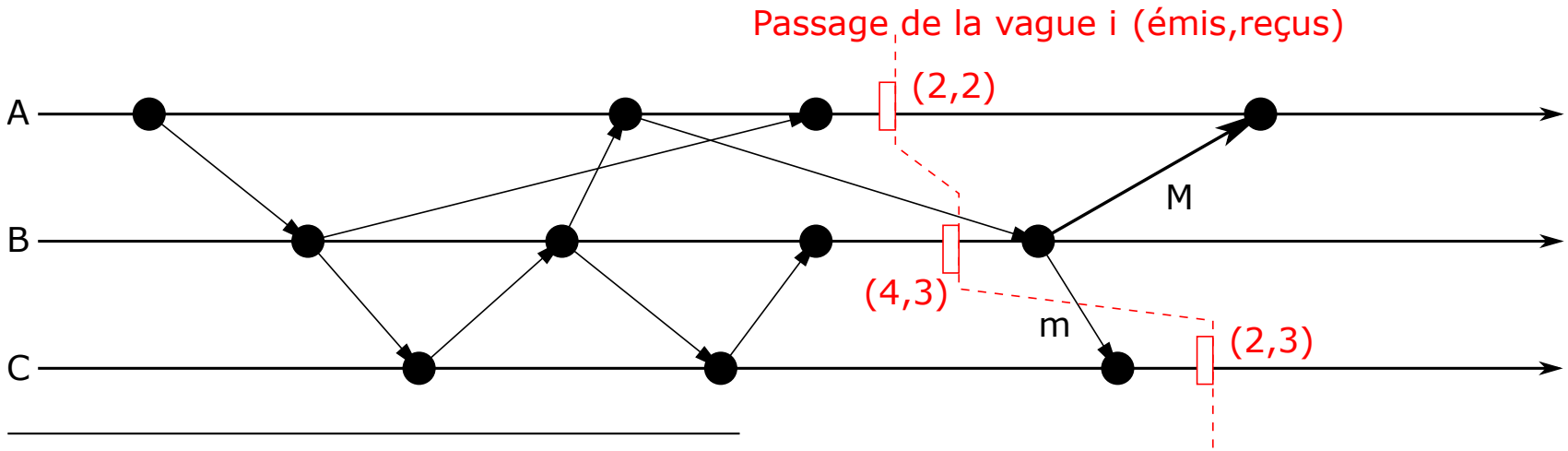
- Circuit logique contenant tous les sites (mais pas nécessairement tous les arcs)
- Communication causale : le jeton en transit depuis un site s ne peut pas arriver sur s' avant les messages émis avant sa visite sur s et envoyés directement de s vers s'
- Terminaison comme précédemment, avec $N =$ longueur du circuit



Algorithme des quatre compteurs

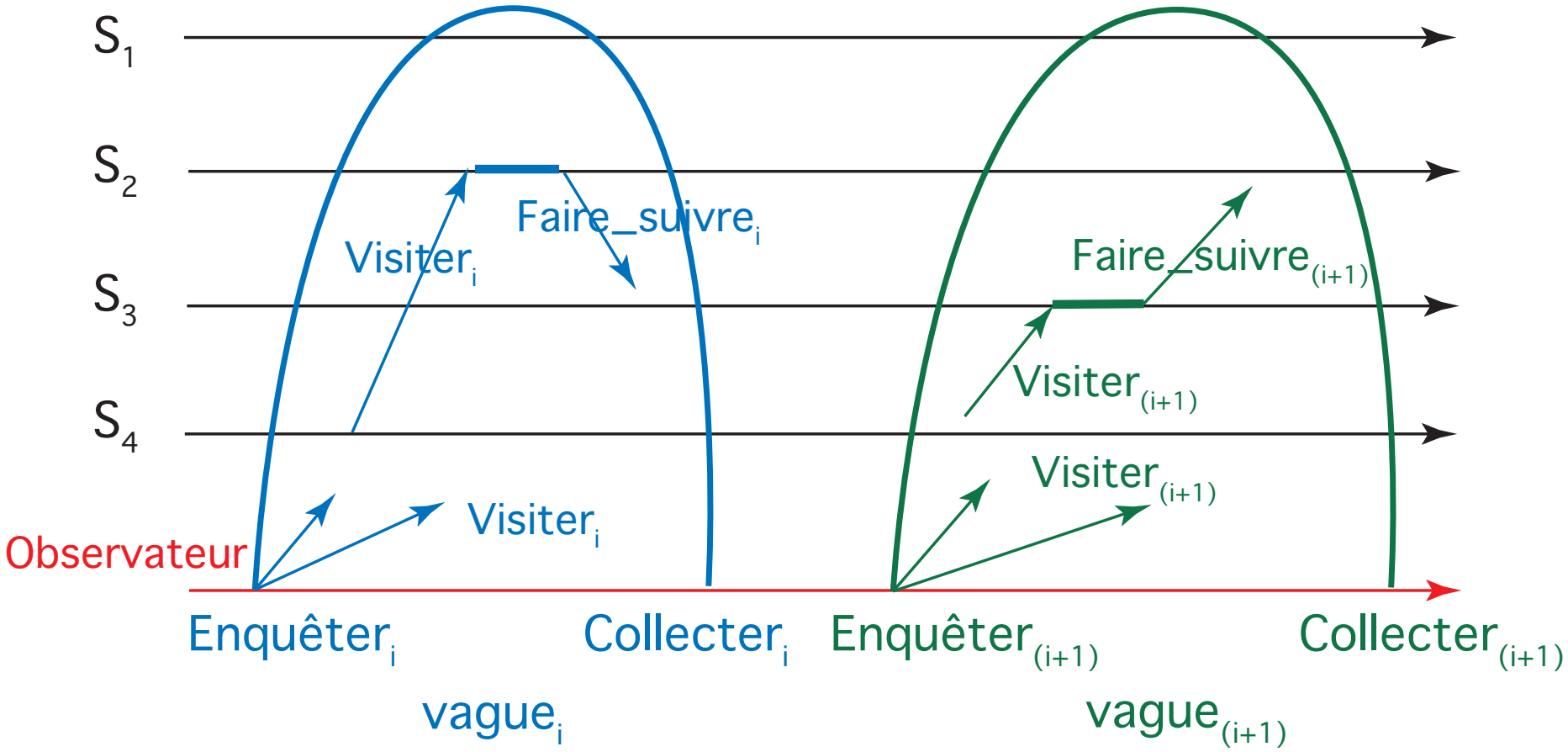
Principe

- Terminaison $\stackrel{\Delta}{=} Emis(t) = Reçus(t)$ mais impossible à évaluer
- Approche : Compteurs locaux des messages émis et reçus
- Mécanisme de vague pour collecter les valeurs des compteurs
- La vague i collecte $R_i \stackrel{\Delta}{=} \sum r_i$ et $E_i \stackrel{\Delta}{=} \sum e_i$



1. *Algorithms for Distributed Termination Detection*, Friedemann Mattern. Distributed Computing, 1987.

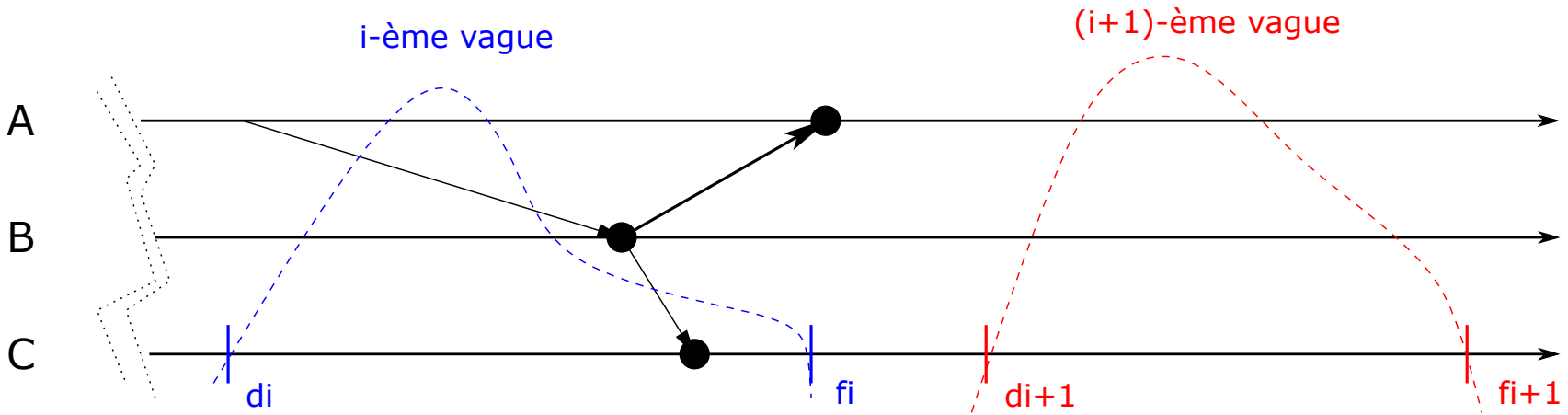
Vague : itération répartie



Algorithme des quatre compteurs

Détection de la terminaison

- Nécessite deux vagues successives
- Terminaison si : $R_i = E_{i+1}$ (car $\Rightarrow \exists t < d_{i+1} : E(t) = R(t)$)
- Détection avec un retard d'au plus la durée de la dernière vague

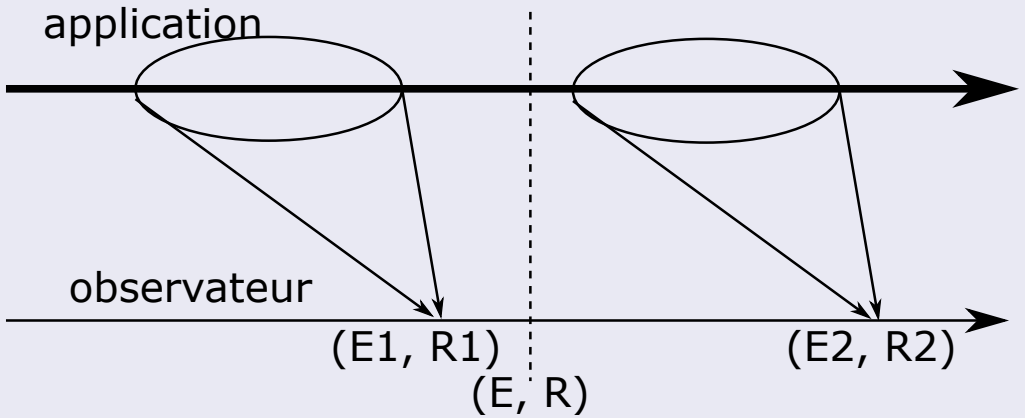


Algorithme des quatre compteurs – preuve

Vivacité : si le calcul est terminé, alors la terminaison est détectée

Calcul terminé $\Rightarrow$ il existe une date à partir de laquelle les compteurs de messages émis/reçus par site ne changent plus $\Rightarrow$ deux vagues successives trouveront $E_1 = R_1 = E_2 = R_2$.

Sûreté : si la terminaison est annoncée, alors le calcul est terminé



(1)	$E \geq R$	*calcul réparti*
(2)	$E_1 \leq E \leq E_2$	*compteurs croissants*
(3)	$R_1 \leq R \leq R_2$	*idem*
(4)	$E_2 = R_1$	*détection annoncée*
(5)	$E \leq R$	*d'après 2,3,4*
	$E = R$	*d'après 1,5*

(E, R) valeurs réelles (et inconnues) des compteurs entre deux vagues successives.

Algorithme des crédits (Mattern)

Un peu oublié mais pourtant simple et performant...

Principe

- Le processus initial possède un crédit de 1
- Le crédit courant est partagé entre les messages émis
- Un processus rend son crédit s'il n'envoie pas de message
- Terminaison lorsque la somme collectée égale 1