

# Diverses formes de stabilisation

Alain Cournier & Stéphane Devismes

# Plan du cours

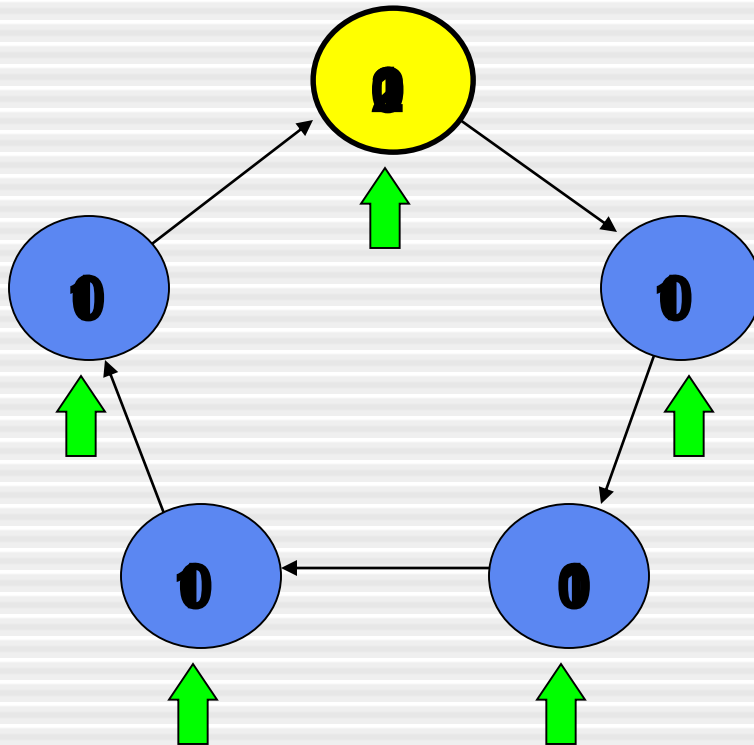
1. Autostabilisation déterministe
2. Avantages
3. Inconvénients
4. Formes faibles
5. Autostabilisation probabiliste
6. Stabilisation faible

# Autostabilisation déterministe

- [Dijkstra, 1974]:
  - « Un protocole  $P$  est *autostabilisant* si, Pour **toute** configuration initiale, **toute** exécution de  $P$  atteindra en un temps fini une configuration à partir de laquelle son comportement sera correct »

# Autostabilisation [Dijkstra, 1974]

## Exemple: Circulation de jeton (Anneau)



Constante :  $K > N$  (ici 6)

Variable :  $S \in [0..K-1]$

Pour la racine

Jeton  $\equiv (S = S_{Pred})$

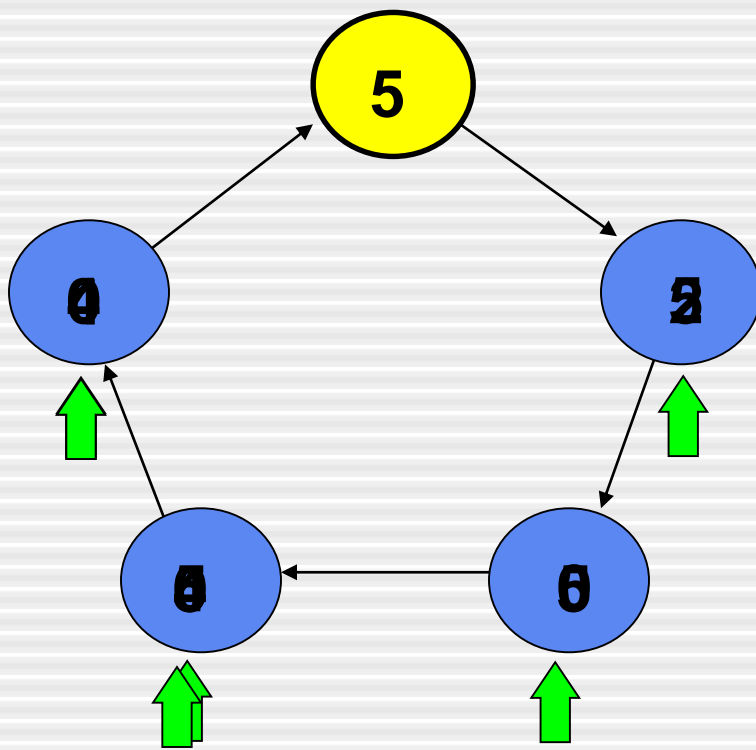
Si Jeton alors  $S \leftarrow S+1$

Pour les autres

Jeton  $\equiv (S \neq S_{Pred})$

Si Jeton alors  $S \leftarrow S_{Pred}$

# configuration quelconque

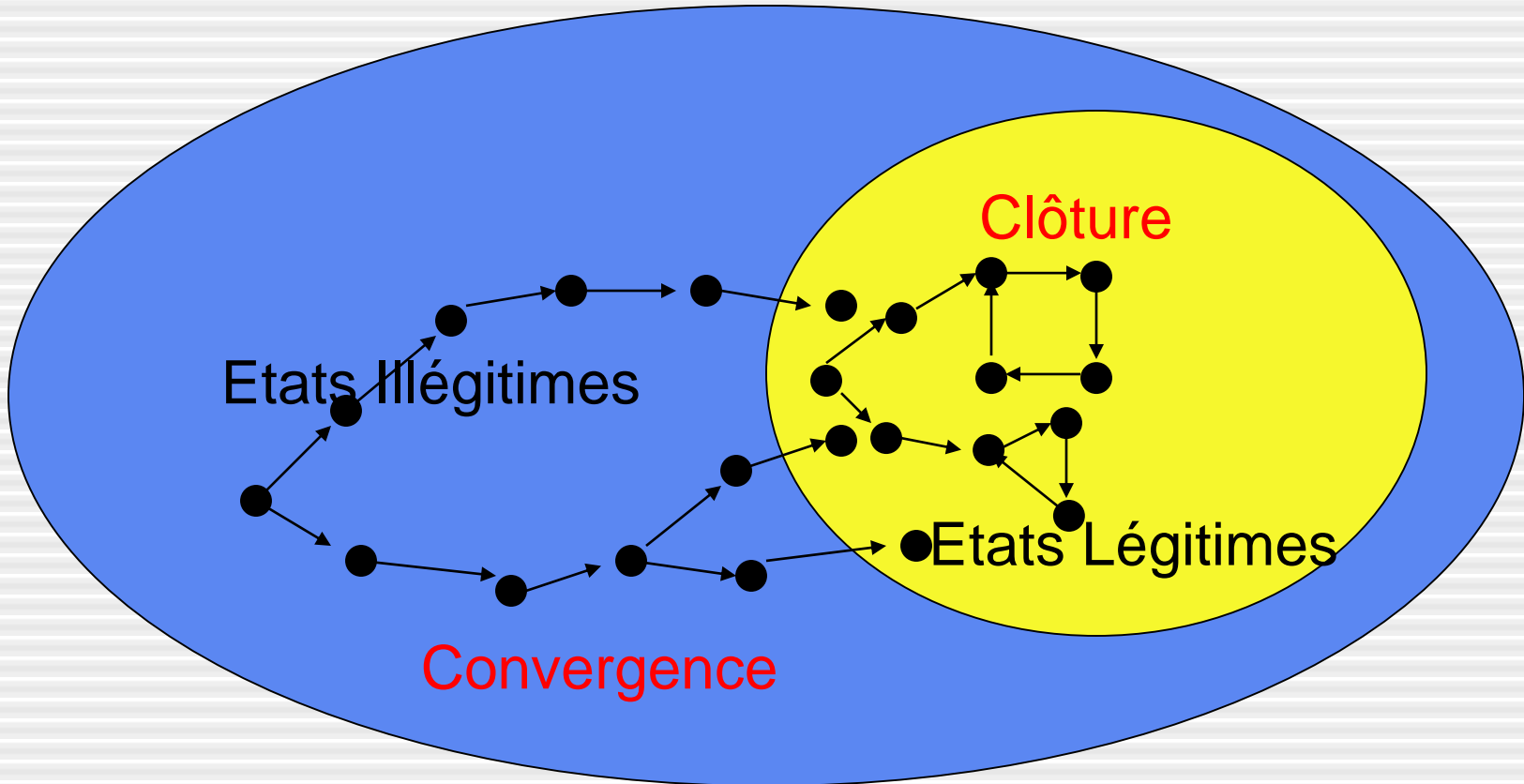


Constante :  $K > N$  (ici 6)  
Variable :  $S \in [0..K-1]$

Pour la racine  
Jeton  $\equiv (S = S_{\text{Pred}})$   
Si Jeton alors  $S \leftarrow S+1$

Pour les autres  
Jeton  $\equiv (S \neq S_{\text{Pred}})$   
Si Jeton alors  $S \leftarrow S_{\text{Pred}}$

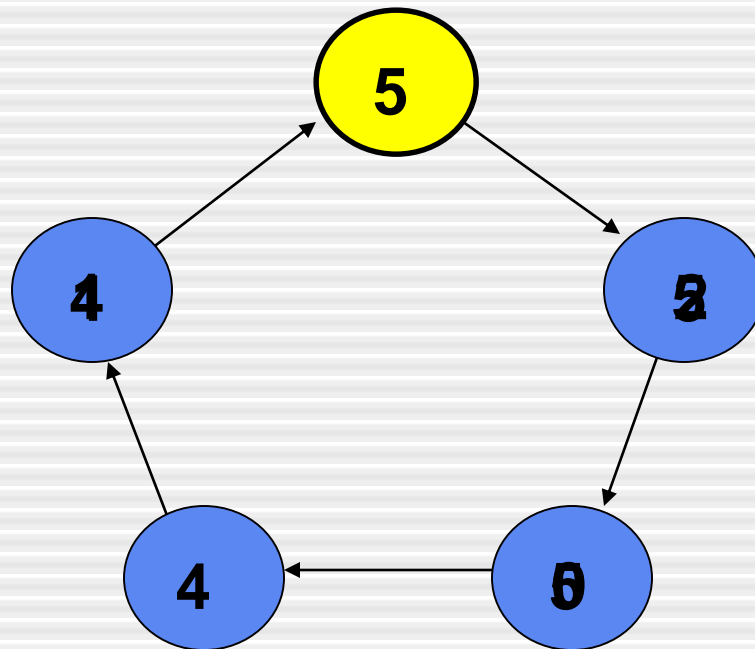
# Définition: Clôture & Convergence



States of the System

# Utilité

La perturbation dure un temps fini, e.g.,  
après une attaque du système...



# Avantages

## 1. Tolérance aux fautes transitoires :

- Aucune hypothèse sur la nature et l'ampleur de la faute
- Retour à la normale par une méthode unique

## 2. Pas d'initialisation :

- Passage à l'échelle

## 3. Dynamicité :

- Auto organisation : réseaux de senseurs et réseaux adhoc



# Inconvénients

## 1. Sureté garantie à terme

- E.g., *Exclusion mutuelle* : un processus peut entrer en SC un grand nombre de fois (non bornée) sans satisfaire la sureté

## 2. Surcoût

- Les protocoles autostabilisants peuvent être gourmand en ressources

## 3. Résultats d'impossibilité

- Problèmes fondamentaux sans solution autostabilisante

# Inconvénients

- Composer avec ces inconvénients :
  - **Propriétés faibles** (Première partie)
  - **Propriétés fortes** (Seconde partie)

# Formes faibles de l'autotabilisation

- Pseudo-Stabilisation
- $K$ -Stabilisation
- Stabilisation probabiliste
- Stabilisation faible
  
- **But** : Prévenir les résultats d'impossibilité
  - E.g., *Coloriage, Election, Circulation de jeton*

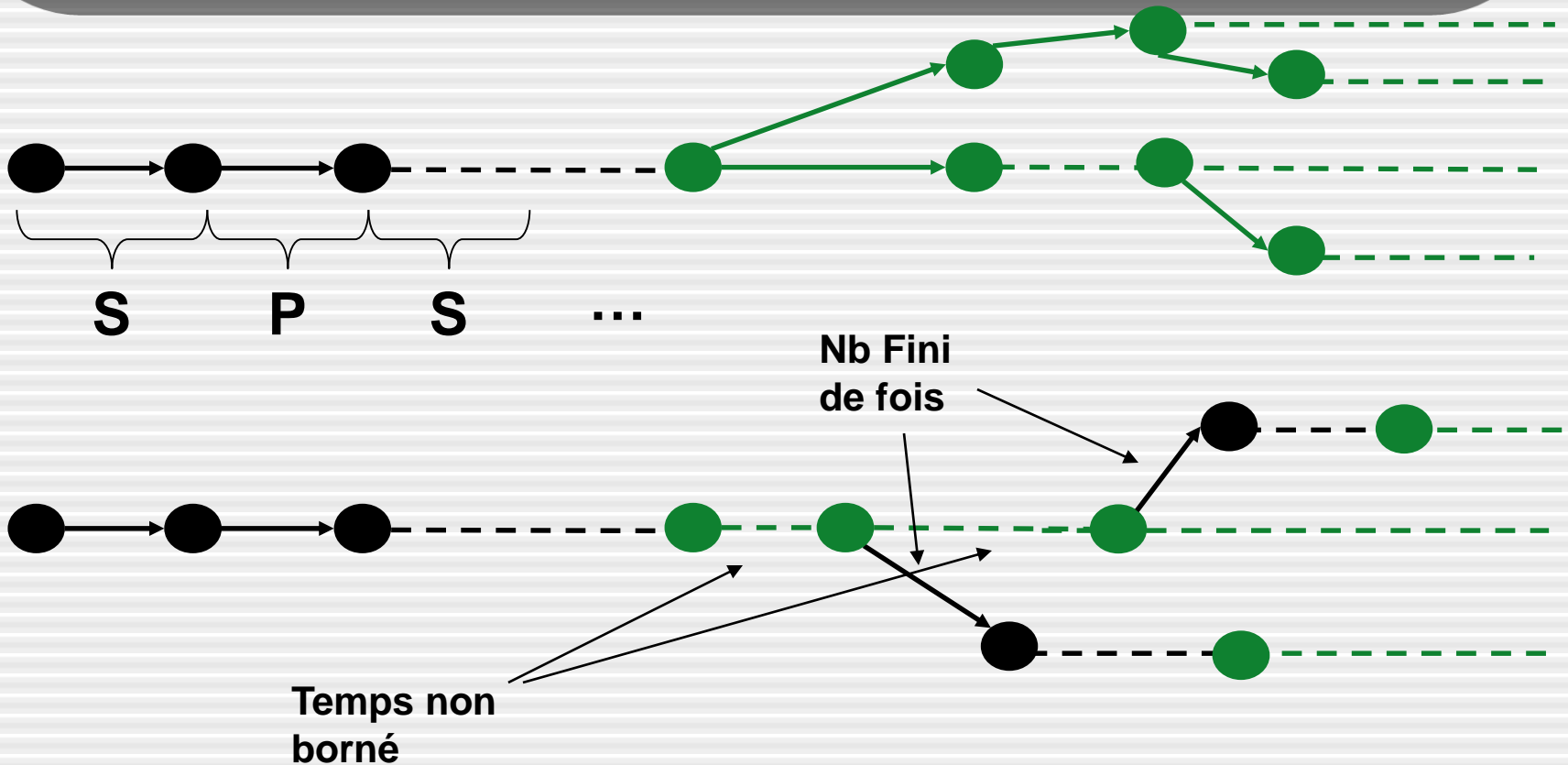


# Pseudo-stabilisation

# Pseudostabilization [Burns *et al*, WSS'89]

- « Toute exécution converge vers *un suffixe correct* »
- Autostabilization:
  - « Toute exécution converge vers une configuration à partir de laquelle tout *suffixe est correct* »
- **Convergence identique & clôture +faible**

# Pseudo- vs. Self-



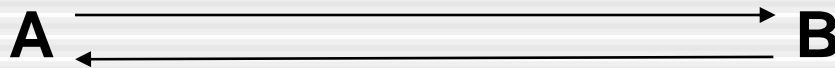
**Conséquence :** le temps de convergence vers un suffixe correct est non borné dans le cas général

# Echanges de messages

- Entre deux nœuds A et B
  - S : Etat de départ
  - E : A envoie le message m
  - R : B reçoit le message
  - T : B envoie l'accusé réception ar
  - S : A reçoit ar, retour état de départ

# Echanges de messages

- Entre deux nœuds A et B
  - S : A reçoit ar, retour état de départ





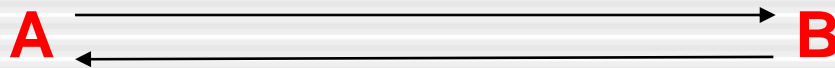
# Echanges de messages

- Entre deux nœuds A et B
  - E : A envoie le message m



# Echanges de messages

- Entre deux nœuds A et B
  - R : B reçoit le message



# Echanges de messages

- Entre deux nœuds A et B
  - T : B envoie l'accusé réception ar



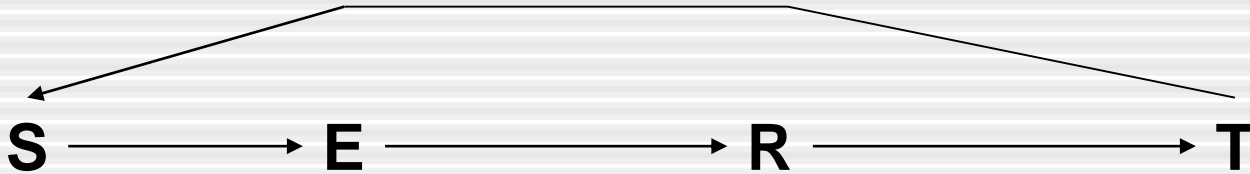
# Echanges de messages

- Entre deux nœuds A et B
  - S : Etat de départ



# Echanges de messages

- Diagramme de base



# Echanges de messages

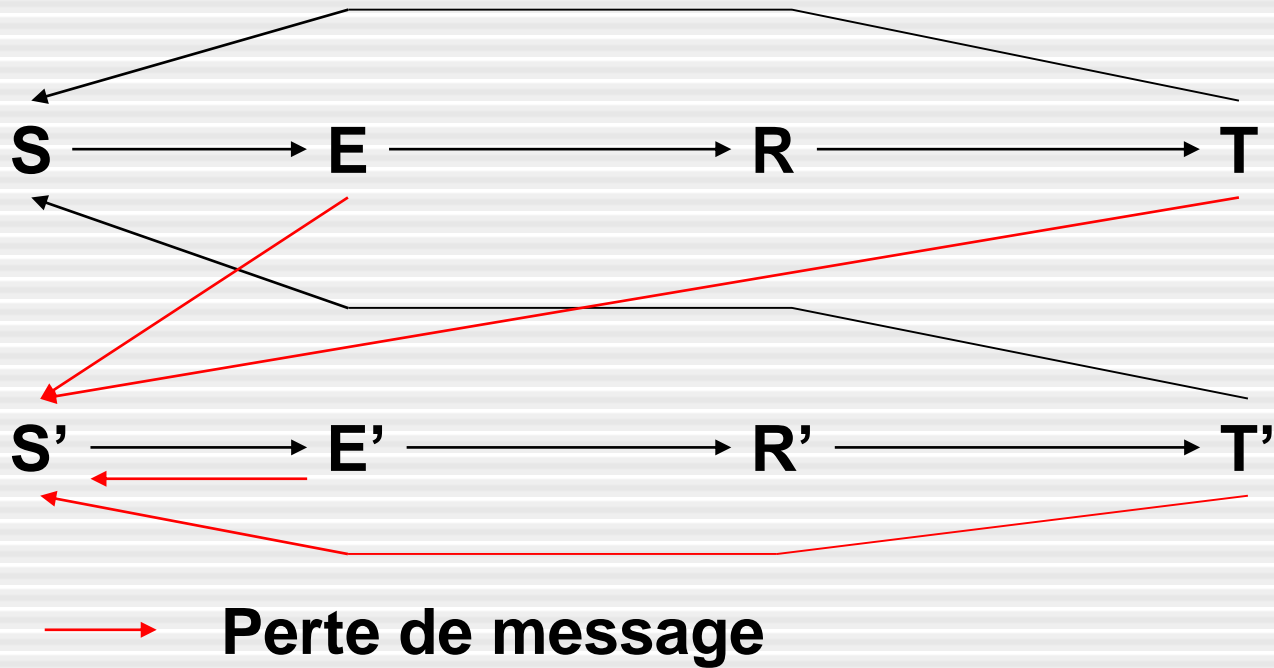
- Si perte d'un message
  - Protocole de rattrapage

# Echanges de messages

- Entre deux nœuds A et B
  - S' : Etat de départ du rattrapage
  - E' : A nouvel envoi du message **m**
  - R' : B reçoit le message
  - T' : B envoie l'accusé réception **ar**
  - S : A reçoit **ar**, retour état de départ

# Echanges de messages

- Diagramme global



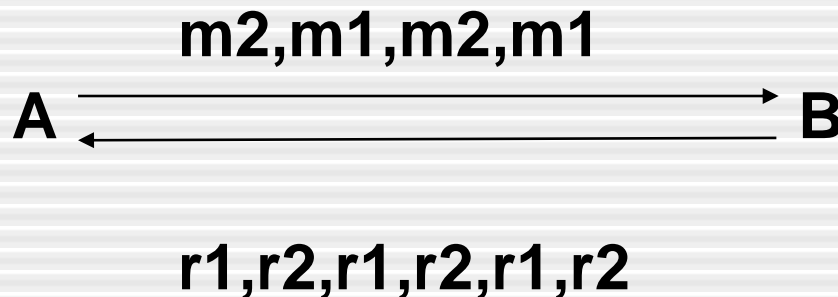


# Echanges de messages

- Pertes équitables et régulières
- Pas autostabilisant
- Pseudostabilisant

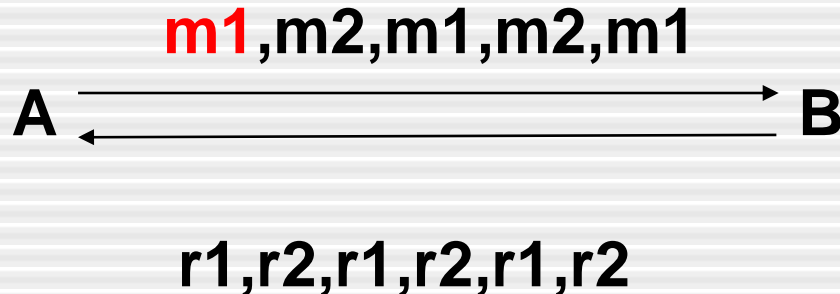
# Echanges de messages

- Pertes équitables et régulières
- Pas autostabilisant
- Etat initial



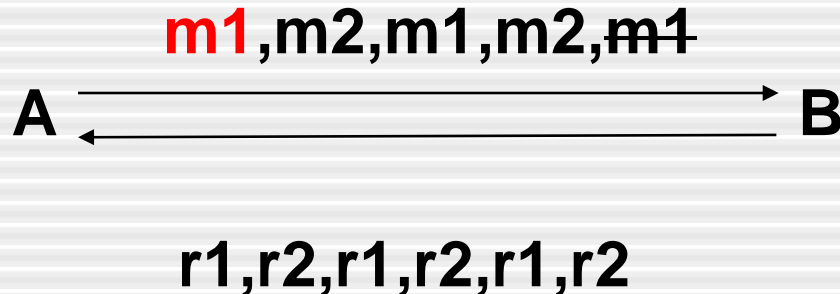
# Echanges de messages

- Pertes équitables et régulières
- Pas autostabilisant
- A envoie un message (en rouge)



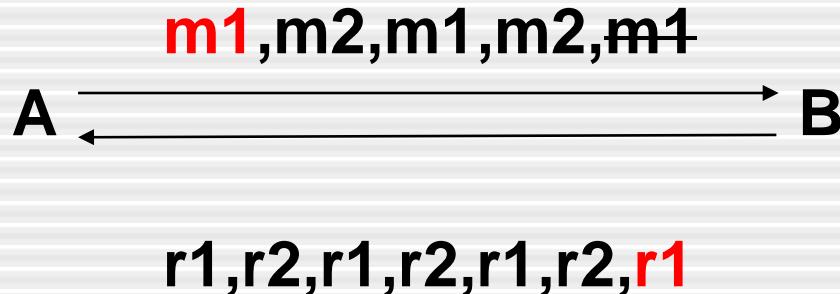
# Echanges de messages

- Pertes équitables et régulières
- Pas autostabilisant
- B reçoit un message (barré)



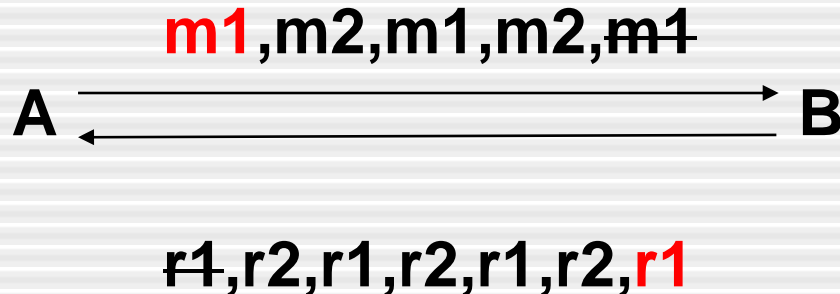
# Echanges de messages

- Pertes équitables et régulières
- Pas autostabilisant
- B envoie un accusé réception (rouge)



# Echanges de messages

- Pertes équitables et régulières
- Pas autostabilisant
- A reçoit un accusé réception (barré)



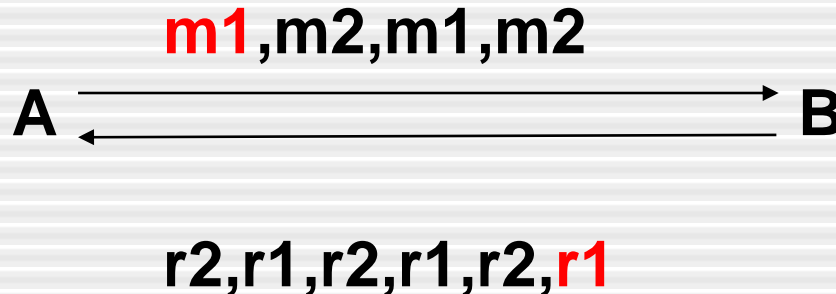
# Echanges de messages

- Pertes équitables et régulières
- Pas autostabilisant
- Fin de cycle



# Echanges de messages

- A a envoyé son message m1
- A a reçu un AR pour m1
- On recommence un cycle





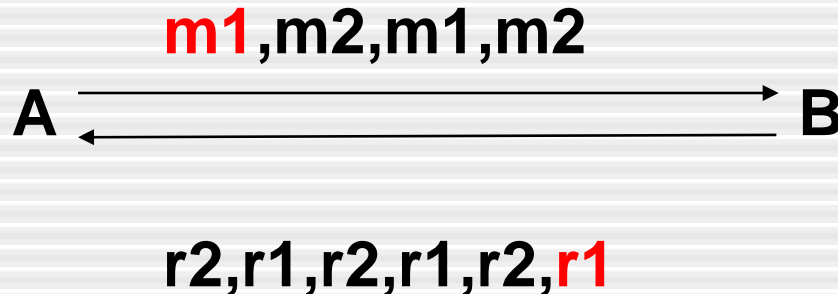
# Echanges de messages

- Pas de stabilisation (nettoyage des canaux) en l'absence de perte de messages.



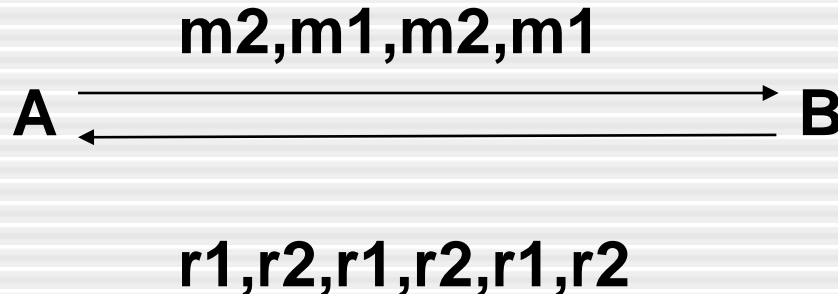
# Echanges de messages

- Donc pas autostabilisant.



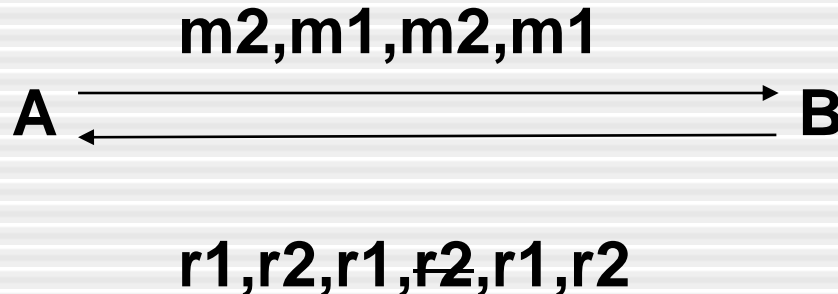
# Echanges de messages

- Pertes équitables et régulières
- Pseudostabilisation
- Etat initial



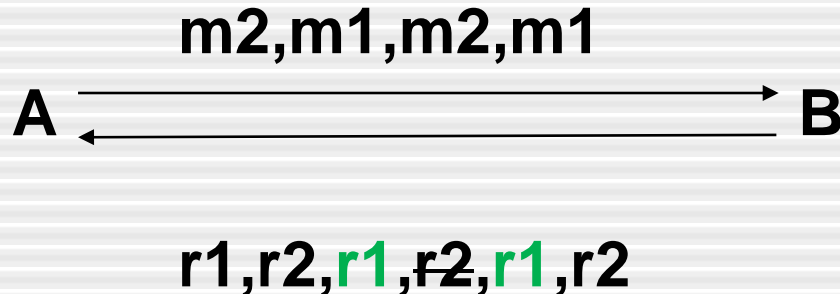
# Echanges de messages

- Pertes équitables et régulières
- Pseudostabilisation
- Perte d'un message (barré)



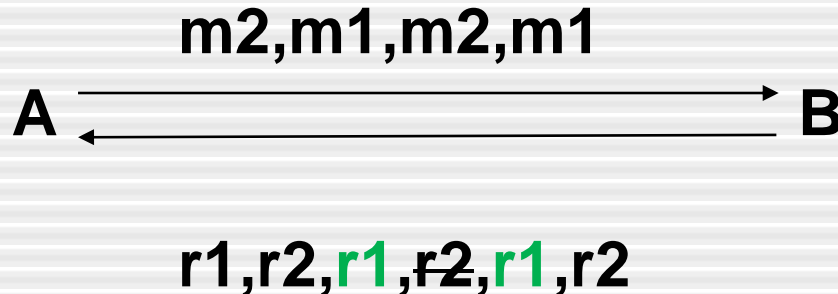
# Echanges de messages

- Pertes équitables et régulières
- Pseudostabilisation
- Un message est répété (ici r1 en vert)



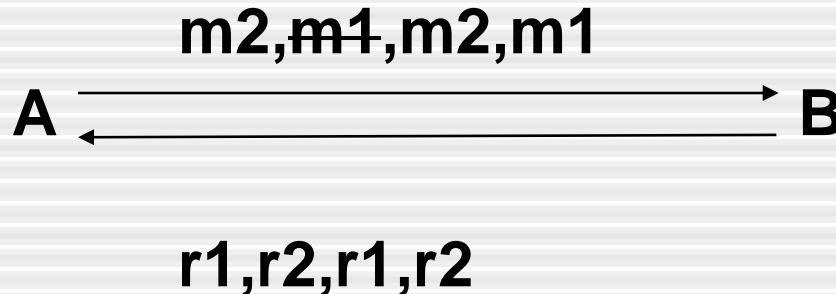
# Echanges de messages

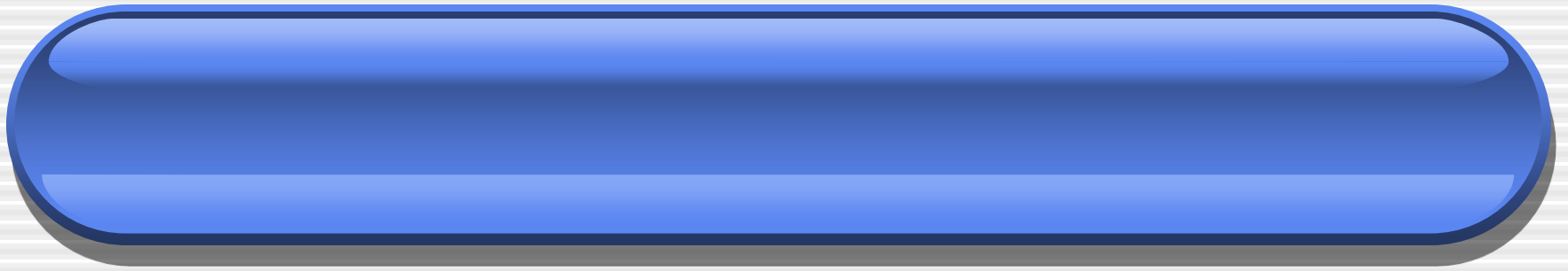
- La perte d'un message entraine la disparition de deux messages dans les canaux.



# Echanges de messages

- La perte régulière de messages entraîne le nettoyage des canaux
- Donc stabilisation (pseudo)



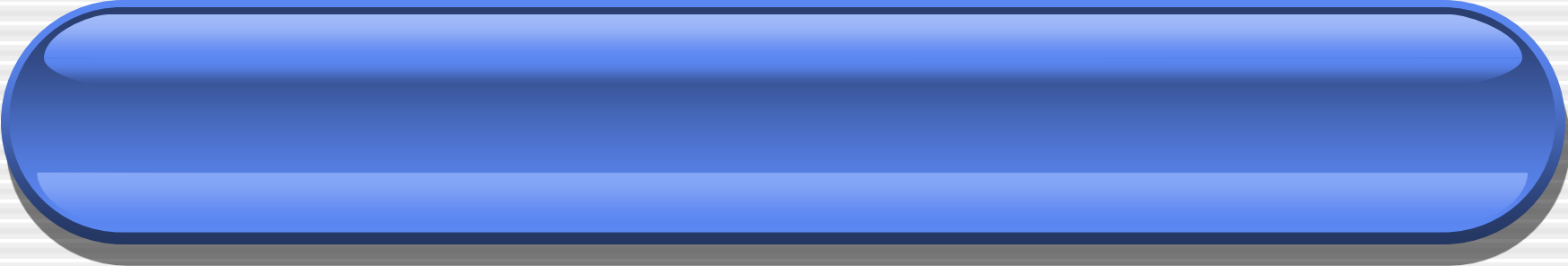


# K-Stabilisation



# *K*-Stabilisation [Beauquier *et al*, PODC'98]

- Autostabilisation:
  - Commencer dans un état quelconque
- *K*-Stabilisation:
  - Certains états peuvent être initiaux : Tous les états pouvant être obtenus avec au plus *K* fautes



# Autostabilisation probabiliste

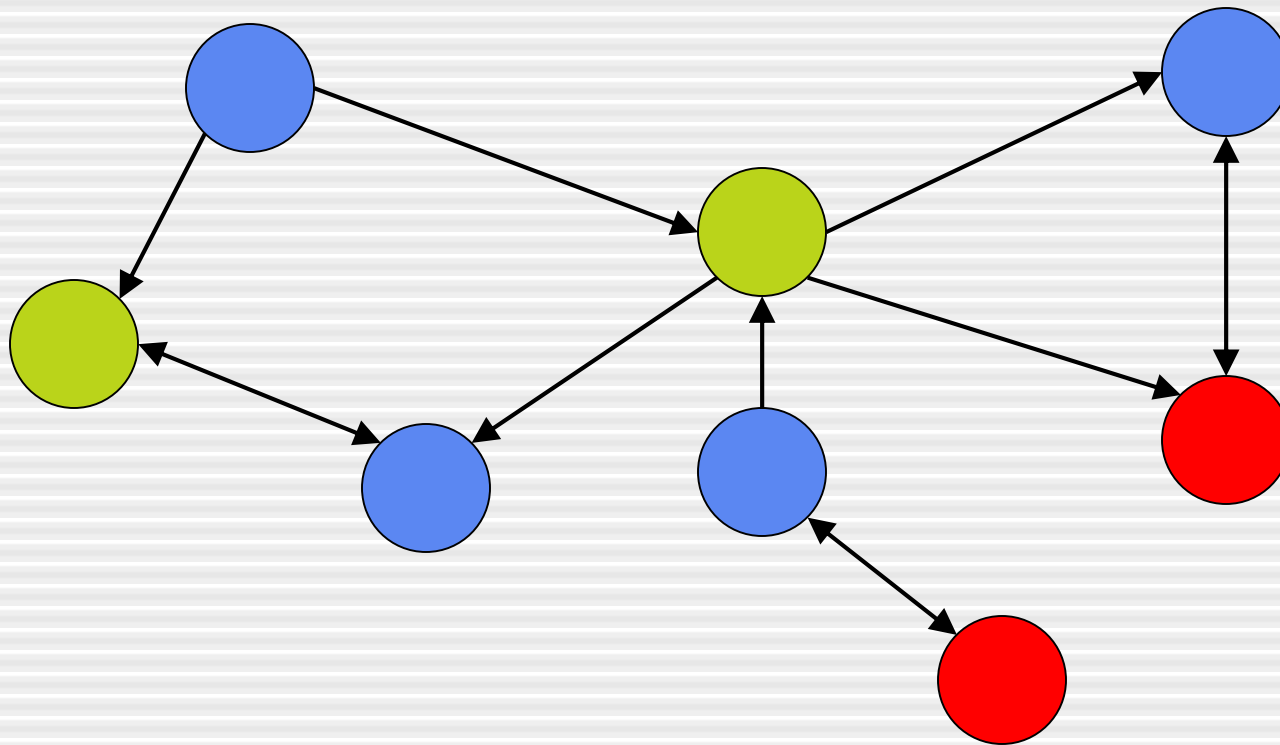
# Autostabilisation probabiliste [Israeli and Jalfon, PODC'90]

- Même règle de clôture
- On allège la convergence :
  - Toute exécution converge *avec une proba de 1*
  - L'espérance du temps de convergence est *fini*



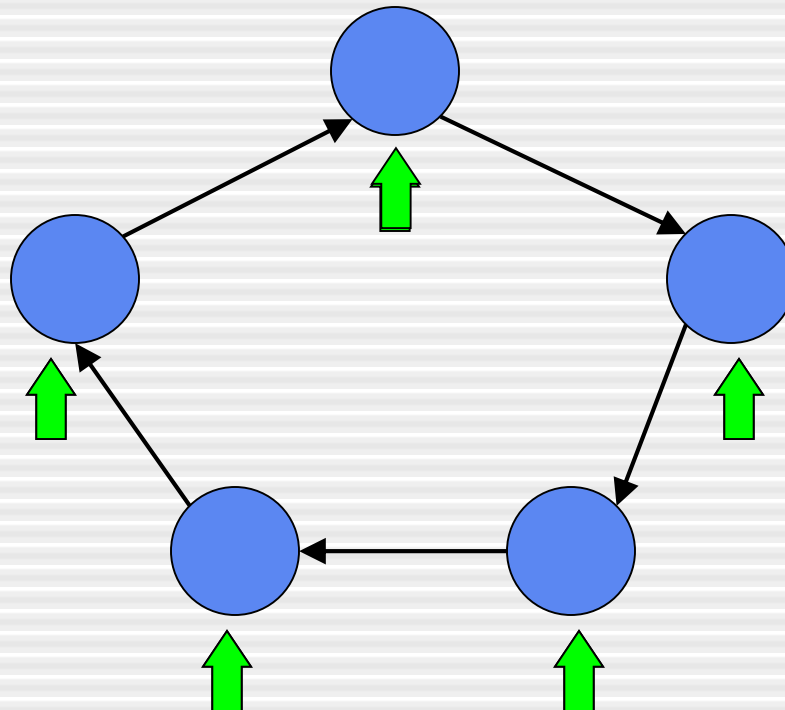
# Colorier un réseau Unidirectionel (General) [RT08]

S. Bernard, S. Devismes, M. Gradinariu, et S. Tixeuil



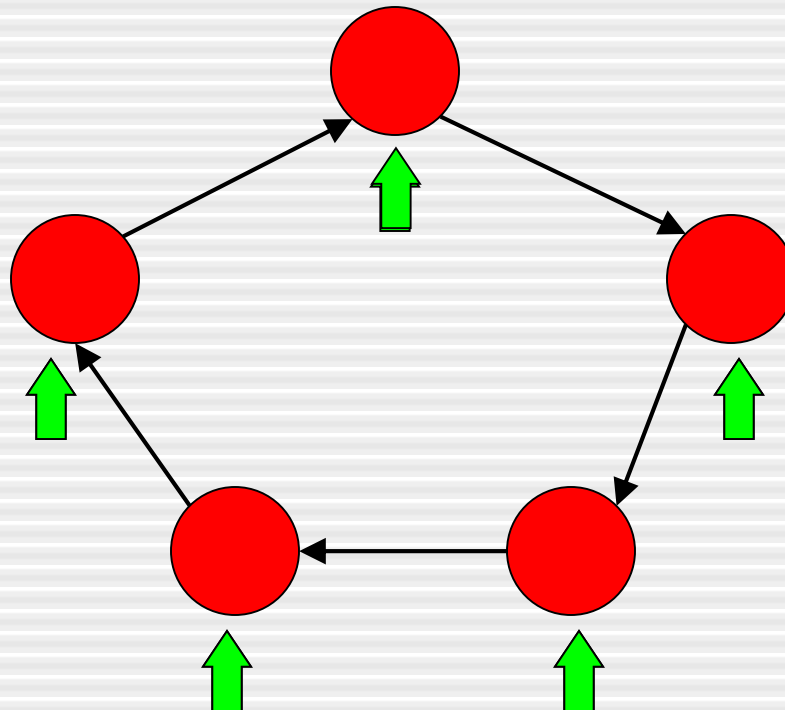
# Résultat d'impossibilité

- Pas de solutions autostabilisant déterministe avec un démon distribué

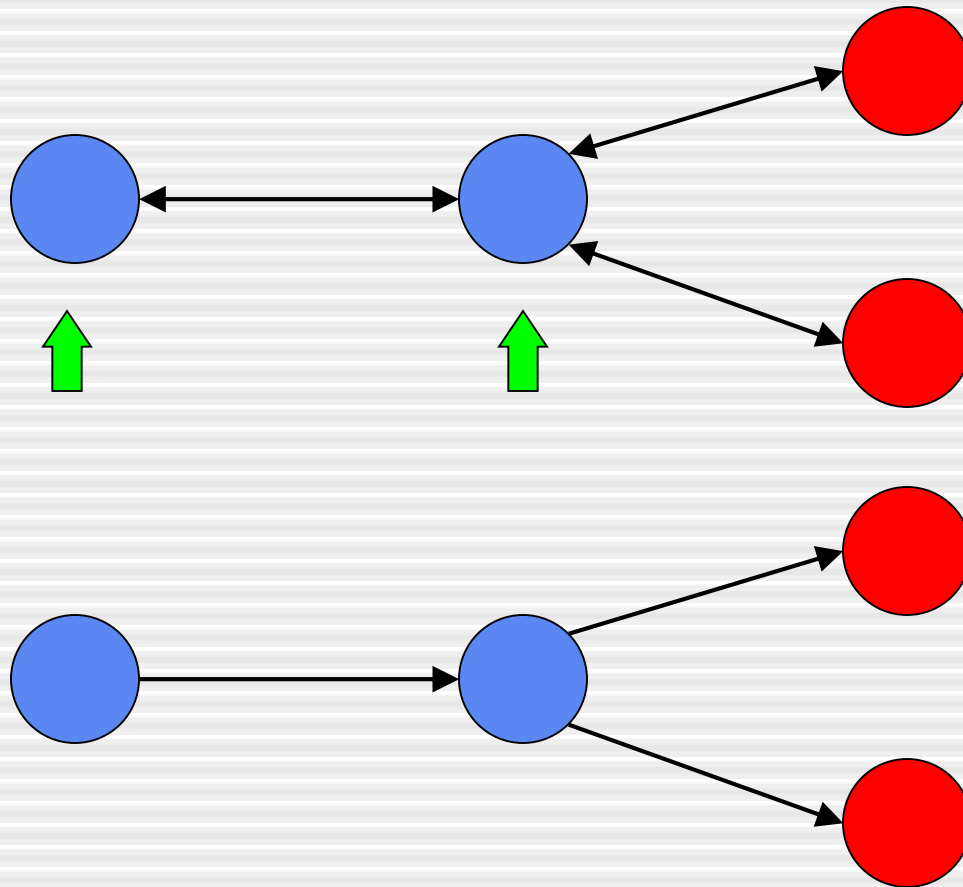


# Résultat d'impossibilité

- Pas de solutions autostabilisant déterministe avec un démon distribué



# Remarque : Plus Difficile que le cas bidirectionnel



Dans le cas unidirectionnel, le nombre de « conflits » peut augmenter !



# Algorithme

```
process i  
const  
    k : integer    Assumption:  $k > \Delta$   
    P.i : set of predecessors of i  
    C.i : set of colors of nodes in P.i  
parameter  
    p : node in P.i  
var  
    c.i : color of node i  
action  
     $p \in P.i, c.i = c.p \rightarrow$   
         $c.i := \text{random}(\{0, \dots, k - 1\} \setminus C.i)$ 
```

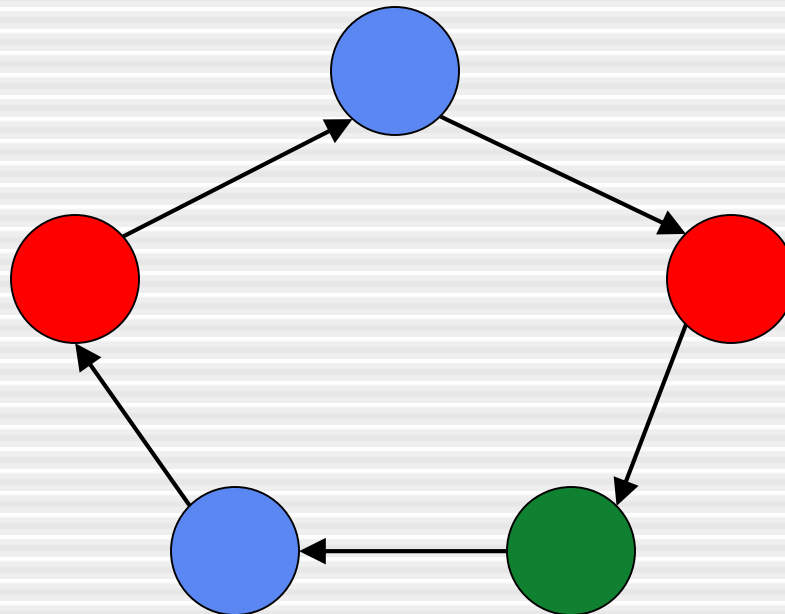
# Algorithme

## Prédécesseurs Propres

```
process i
const
  k : integer      Assumption:  $k > \Delta$ 
  P.i : set of predecessors of i
  C.i : set of colors of nodes in P.i
parameter
  p : node in P.i
var
  c.i : color of node i
action
   $p \in P.i, c.i = c.p \rightarrow$ 
     $c.i := \text{random}(\{0, \dots, k - 1\} \setminus C.i)$ 
```

# Preuve

- Clôture : Trivial
  - Quand un coloriage est réalisé, l'algorithme est silencieux

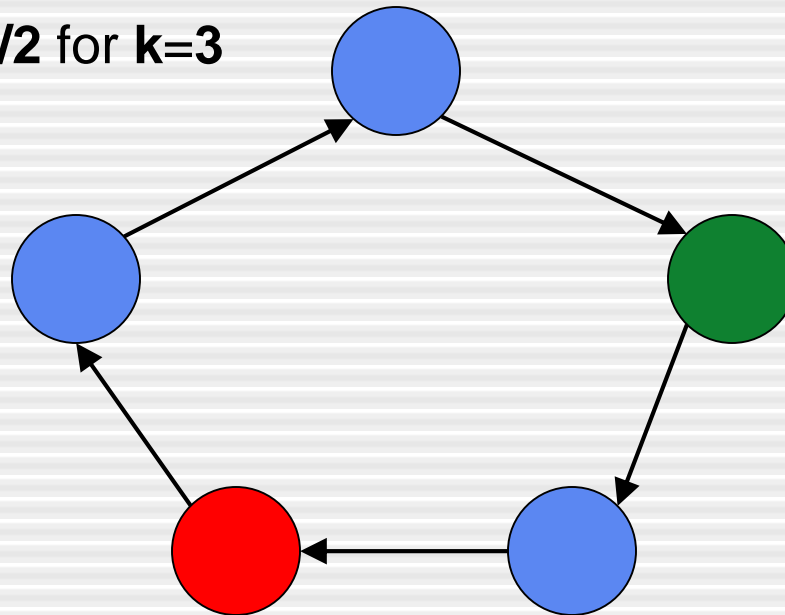


# Convergence (1/4)

**Lemma 1** Assume  $k > \Delta$ . Let  $(p, \gamma)$  be a conflict. The expected number of conflicts created by the execution of one step of  $p$  in order to resolve  $(p, \gamma)$  is at most:

$$\frac{\delta.p - \delta_{in}.p}{k - \delta_{in}.p} \quad (1)$$

- Exemple: Ring,  $1/2$  for  $k=3$



# Convergence (2/4)

**Lemma 2** *Let  $(p, \gamma)$  be a conflict. The expected number of conflicts created by the execution of one step of  $p$  in order to resolve  $(p, \gamma)$  is less than or equal to:*

$$M = \frac{\Delta - 1}{k - 1}, \quad k > \Delta \quad (2)$$

- Explications :

1.  $\delta.p$  is maximized by  $\Delta$

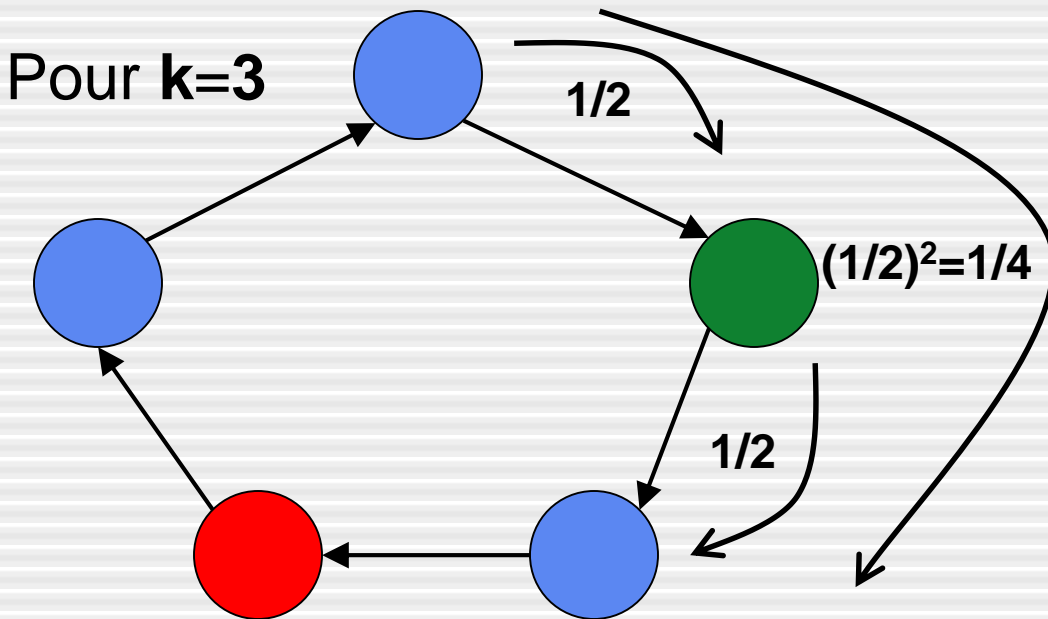
2. Using the derivative,  $M$  is maximal for  $\delta_{in}.p = 1$

Remarque: pour l'anneau,  $M = 1/2$  quand  $k = 3$

# Convergence (3/4)

**Lemma 3** *Let  $(p, \gamma)$  be a conflict. The expected number of step created in order to resolve this conflict is less than:*

$$\frac{k-1}{k-\Delta}, \quad k > \Delta \quad (3)$$



The expected number of step created is less than:

$$\sum_{i=0}^{\infty} M^i = \sum_{i=0}^{\infty} \left( \frac{\Delta-1}{k-1} \right)^i = \frac{k-1}{k-\Delta}$$

# Convergence (4/4)

**Lemma 4** *Starting from an arbitrary configuration, the expected number of steps to reach a configuration verifying the unidirectional coloring predicate is less or equal to:*

$$\frac{n(k-1)}{k-\Delta}, \quad k > \Delta \quad (4)$$

- Trivial à partir du Lemme 3

# Conclusion

- Avec un nombre minimal de couleurs ( $\Delta+1$ ), ils obtiennent **une espérance de temps de convergence** d'au plus  $n.\Delta$  étapes
- Lorsque  $k$  devient grand ( $k \rightarrow \infty$ ), cette espérance de temps de convergence est proche de  $n$  étapes



# Stabilization

## Faible vs. Self vs. Probabiliste

Stéphane Devismes (MIS, Amiens, France)

Sébastien Tixeuil (LIP6-CNRS & INRIA, France)

Masafumi Yamashita (Kyushu University, Japan)

# Stabilisation faible [Gouda, WSS'01]

- Clôture préservée
- Mais convergence affaiblie :
  - Pour toute configuration, il existe une exécution qui converge



# Pour les problemes

- Stabilisation probabiliste
- Pseudo-stabilisation
- K-Stabilisation
- Stabilisation faible

> **Autostabilisation**

E.g. coloriage de graphes,  
circul. jeton, bit alterné, ...

# Comparaison

- Du point de vu problème,  
Stabilisation faible > Autostabilisation
- Stabilisation faible & Stabiliation probabiliste sont  
fortement liée

# Faible > Auto

(Point de vu problème)

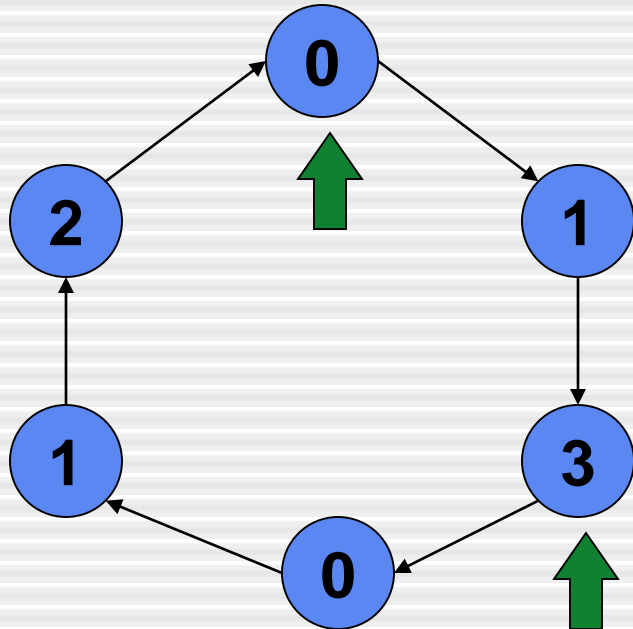
- Deux exemples :
  - **Circulation de jetton** dans un anneau unidirectionnel anonyme sous un démon distribué
  - **Election** dans un arbre anonyme sous un démon distribué (2 algorithmes)

# Circulation de jeton dans un anneau unidirectionnel

- **[Herman, IPL'90]** : « Pas de solution déterministe »
- L'algorithme de **[Beauquier *et al*, DC'07]** est faiblement stabilisant

# L'algorithme

Exemple  $N=6$  ( $m_N=4$ )



**Variable:**  $dt_p \in [0 \dots m_N - 1]$

**Macro:**

$$PassToken_p = dt_p \leftarrow (dt_{Pred_p} + 1) \bmod m_N$$

**Predicate:**

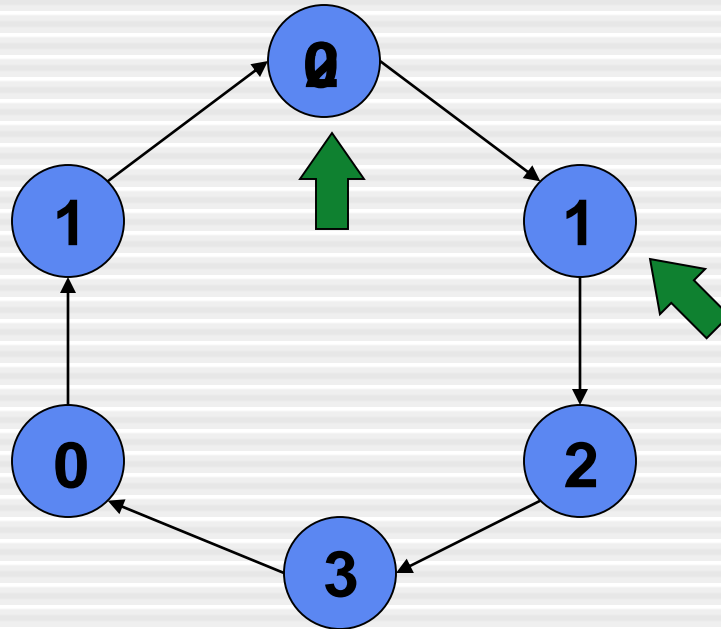
$$Token(p) \equiv [dt_p \neq ((dt_{Pred_p} + 1) \bmod m_N)]$$

**Action:**

$$A :: Token(p) \rightarrow PassToken_p$$

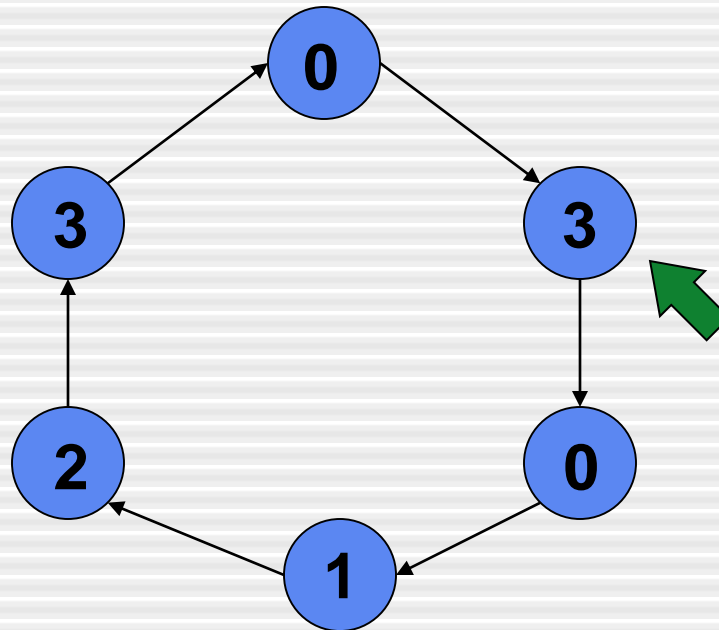


# Clôture



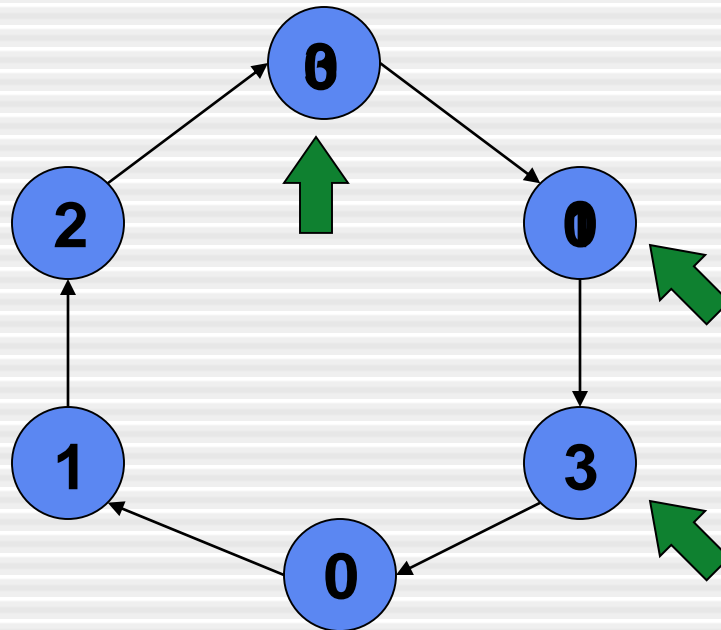
# Convergence (1/2)

- Il y a toujours au moins un jeton



# Convergence (2/2)

- Si le nombre de jetons est supérieur à 1, Il est possible de le faire baisser

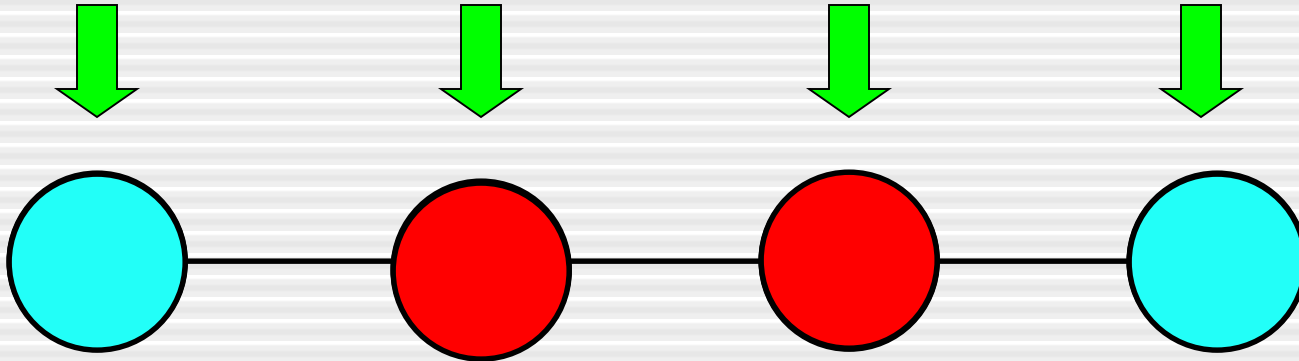


# Election dans un arbre Anonyme avec un démon Distribué

- Pas de solution déterministe autostabilisante
- Deux solutions faiblement stabilisante

# Impossibilité de l'élection

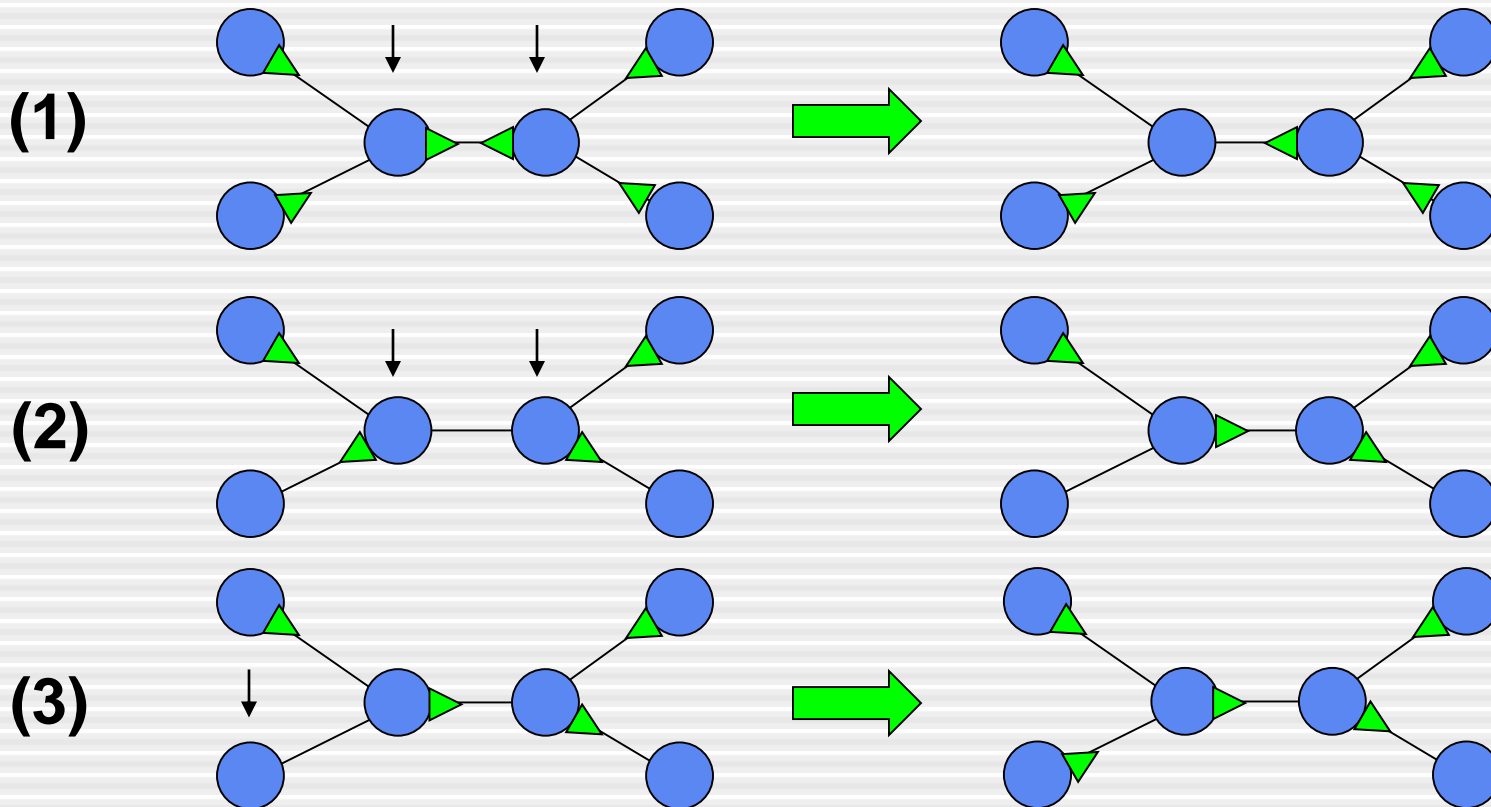
(Avec un démon distribué)



**Exécution synchrone**

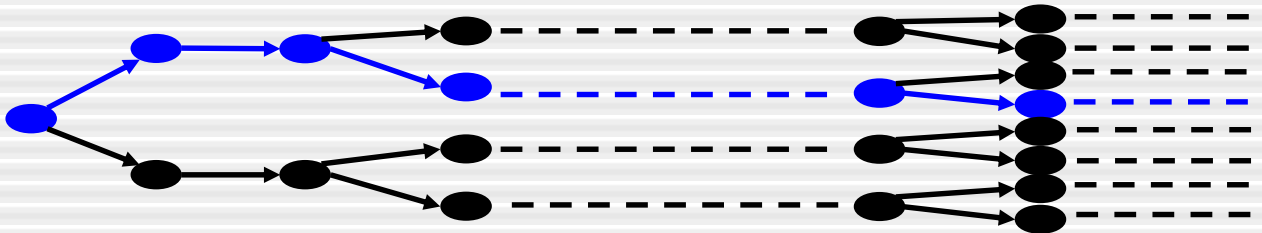
# Stabilisation faible et élection

- Utilisation du pointeur parent  $Par \in Neig \cup \{\perp\}$ , 3 cas :

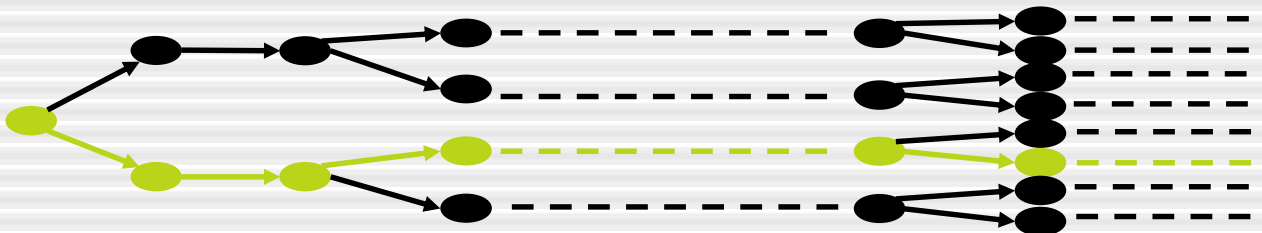


# Version faible plus simple que la version auto

- Démon pour autostabilisation : *Adversaire*



- Démon pour stabilisation faible : *Ami*



- Démon synchrone : **faible = Auto**



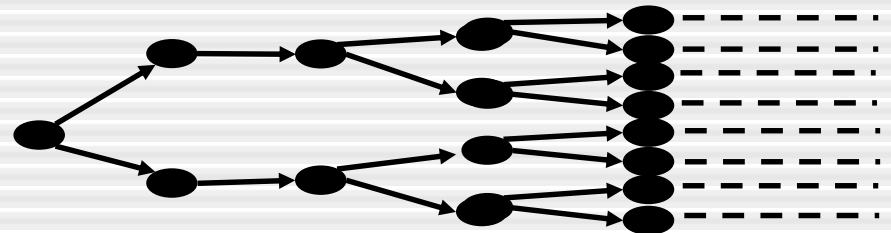
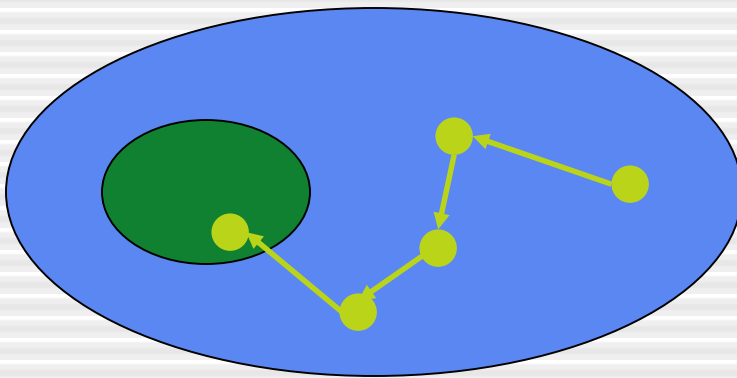
# Observation : faible vs. probabiliste

Si un protocole  $P$  a un nombre fini de configurations, alors

$P$  est faiblement stabilisant *ssi*

$P$  est probabilistiquement stabilisant avec un démon aléatoire

i.e. Exécution: marche aléatoire sur un ensemble fini (de config)





# Problème : Cas Synchrones

**Stabilisation faible avec un démon distribué**

**Démon aléatoire  
(Asynchrone)**



**Synchrone**



**Stabilisation probabiliste  
Dans tous les cas**

**Pas de stabilisation probabiliste  
en général**

**Solution : Lorsqu'on est  
activable jouer à pile ou face**

# Conclusion

- Du point de vu problème,  
***Stabilisation faible > autostabilisant***
- ***Stabilisation faible = Stabilisation probabiliste*** avec un demon aléatoire et un ensemble fini de configurations
  - Intéressant en pratique :
    - La stabilisation faible est plus facile à obtenir que la probabiliste
    - Dans les systèmes réels, les demons ont un comportement aléatoire
    - On peut les *transformer facilement* pour un demon synchrone
- **Perspective** : Evaluer l'espérance du temps de convergence

**Fin de la première partie**