

A decorative border of colored dots surrounds the text. It consists of a vertical line of dots on the left, a horizontal line of dots at the top, and a horizontal line of dots at the bottom. The dots are in various colors including purple, blue, cyan, red, green, yellow, and pink.

Algorithmique Répartie

Propriétés Stables et Terminaison

Université François Rabelais de Tours
Faculté des Sciences et Techniques
Antenne Universitaire de Blois

Master 1 Informatique

Option Systèmes d'Information et Aide à la Décision

Mohamed Taghelit
taghelit@univ-tours.fr

Propriétés Stables

Propriétés stables

- propriétés particulières, dans la vie d'une application, que l'on appelle propriétés stables,
- propriétés dont on sait que, lorsqu'elles deviennent vraies dans une configuration du système, elles restent vraies pour toutes les configurations suivantes.

Propriété stable

Une propriété P d'une configuration S est stable si, pour toute configuration S' accessible à partir de S ,

$$P(S) \Rightarrow P(S')$$

Exemples de propriétés stables

- le système (ou une partie des processus qui composent le système) est en état d'interblocage,
- tous les jetons ont disparus de l'anneau à jetons,
- l'application est terminée,
- ...

Propriétés Stables

Problématique

- il ne s'agit pas de savoir si une propriété est stable, mais plutôt de savoir si une propriété P , que l'on sait stable, est réalisée dans l'état courant,
- il n'est pas possible de connaître de manière instantanée l'état de tous les processus,
- l'alternative est, au lieu d'observer le comportement dynamique de l'application, d'analyser les données statiques,
- l'algorithme ci-dessous n'évalue pas P sur l'état courant de l'application, mais sur le dernier état global calculé.

```
Déetecte_stable() {  
    Répéter  
    |   calculer un état global  $S^*$   
    |   calculer  $P(S^*)$   
    Jusqu'à  $P(S^*)$   
    détectée = vrai  
}
```

Variable booléenne, initialisée à faux, prend la valeur vraie lorsque la propriété a été vérifiée sur l'état global.

Propriétés Stables

Remarques

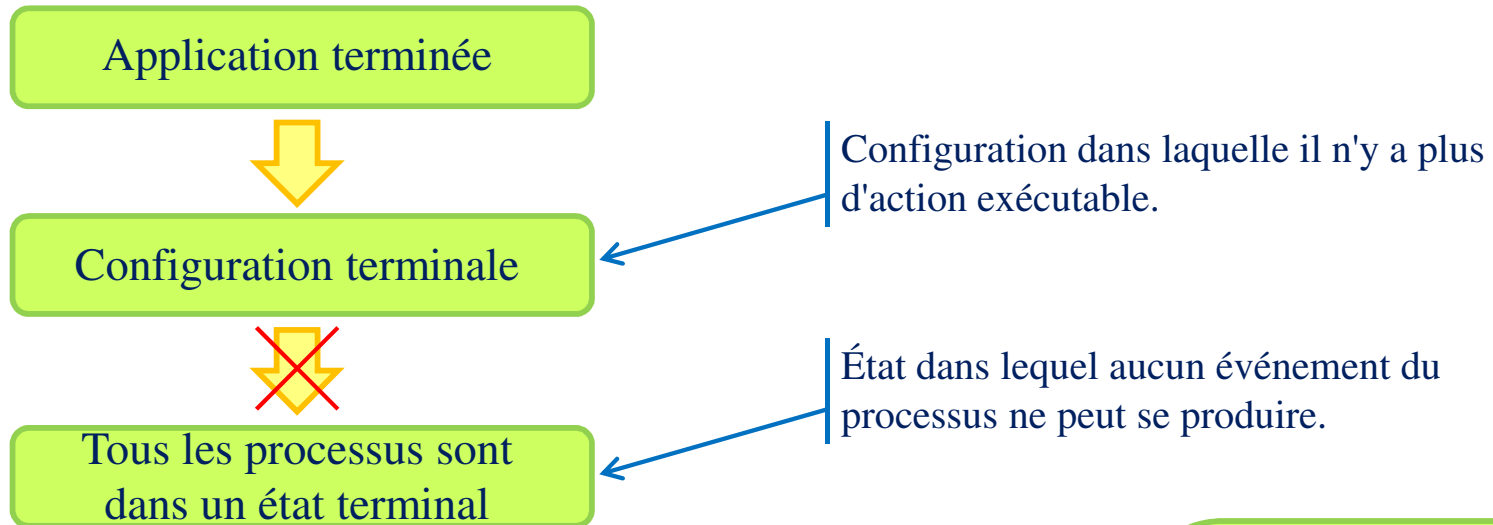
- l'évaluation de \mathbf{P} peut se faire de manière
 - centralisée : chaque processus envoie son état local à un processus particulier qui construit l'état global et évalue \mathbf{P} ,
 - distribuée : tous les processus coopèrent pour évaluer \mathbf{P} ,

L'algorithme permettant une évaluation distribuée de \mathbf{P} est souvent très complexe, mais on trouve ce type d'approche pour la détection d'interblocage par exemple.

- la taille mémoire nécessaire pour stocker les informations sur l'état global peut aller jusqu'à doubler les exigences de l'application,
- même dans le cas où l'application de l'algorithme est simple et efficace, il peut exister des solutions plus simples et plus efficaces qui exploitent la spécificité de la propriété à détecter.
C'est le cas par exemple des algorithmes de détection de la terminaison.

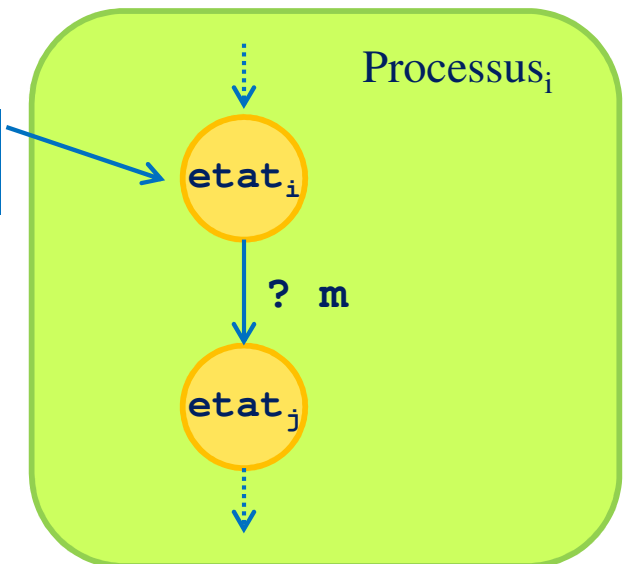
Détection de la Terminaison

Problème de la terminaison



Exemple

- configuration où tous les processus sont dans un état qui autorise les réceptions de messages, mais où il n'y a pas de message en transit,
- l'application est terminée, mais pour chaque processus, l'état dans lequel il se trouve pourrait tout aussi bien se produire en cours d'exécution,
- les processus ne sont pas "conscients" de la terminaison, qui est dite *implicite*,
- la terminaison est *explicite* pour un processus quand il se trouve dans un état terminal.



Détection de la Terminaison

Alternative dans la construction d'un algorithme

- terminaison implicite (plus facile),
- terminaison explicite,

La terminaison explicite est nécessaire pour que le résultat soit définitif

Problématique du passage d'une situation de terminaison implicite à une situation de terminaison explicite

- transition s'effectuant en général en deux phases, par l'application de deux algorithmes :
 - le premier observe l'application et doit détecter, d'une manière ou d'une autre, que l'exécution a atteint une configuration terminale. Il appelle alors le deuxième algorithme,
 - le deuxième algorithme est chargé de diffuser l'information à l'ensemble des processus pour les forcer à atteindre un état terminal.

Étape simple à réaliser : on peut par exemple utiliser un parcours en parallèle.

Comment détecter la terminaison ?

Détection de la Terminaison

Particularité de la terminaison par rapport à une autre propriété stable (interblocage)

- elle concerne l'ensemble des processus impliqués dans un calcul distribué, ou en tout cas un sous-ensemble défini à priori,
 - détecter un interblocage revient à chercher un ensemble de processus qui vérifient une propriété,
 - détecter la terminaison revient à montrer qu'un ensemble de processus vérifient une propriété,

Détection de la terminaison

- utilisation d'algorithmes classiques de détection d'une propriété stable,
- utilisation d'algorithmes spécifiques plus efficaces, généralement basés sur le modèle transactionnel. Ce modèle utilise les hypothèses classiques :
 - communication par messages, un processus est vu comme un nœud dans le système de communication.

et possède les propriétés suivantes :

- un nœud est soit actif, soit passif,
- seul un nœud actif peut envoyer des messages,
- un nœud actif peut spontanément devenir passif,
- la réception d'un message est le seul phénomène qui rend actif un nœud passif.

Détecter la terminaison revient à vérifier que tous les nœuds sont passifs et qu'il n'y a pas de messages en transit.

Détection de la Terminaison

Difficultés de la détection de la terminaison

- la terminaison est une propriété basée sur l'état global de l'application, or chaque processus n'est conscient, à un instant donné, que de son état local,
- un processus ne peut savoir s'il a terminé son calcul ou s'il n'est passif que temporairement. En effet, la réception d'un message peut relancer le calcul d'un processus passif.

Phases d'une détection de terminaison

- visiter l'ensemble des processus pour voir s'ils sont passifs,
- visiter l'ensemble des canaux de communications pour voir s'ils sont vides,
- effectuer ces visites de manière à obtenir un résultat cohérent,

Définir un mode de parcours des sites qui soit cohérent avec la structure de contrôle :

- si la structure de contrôle est un arbre de recouvrement, le parcours se fera par calcul diffusant,
- si la structure de contrôle est un anneau, on utilisera la circulation d'un jeton.

Les solutions au problème de la terminaison distribuée peuvent être classées selon certains critères :

- le type de communication considéré (synchrone, asynchrone),
- la structure de contrôle (anneau, étoile, ...),
- le degré de répartition du contrôle (centralisé → totalement symétrique).

Algorithme de Misra

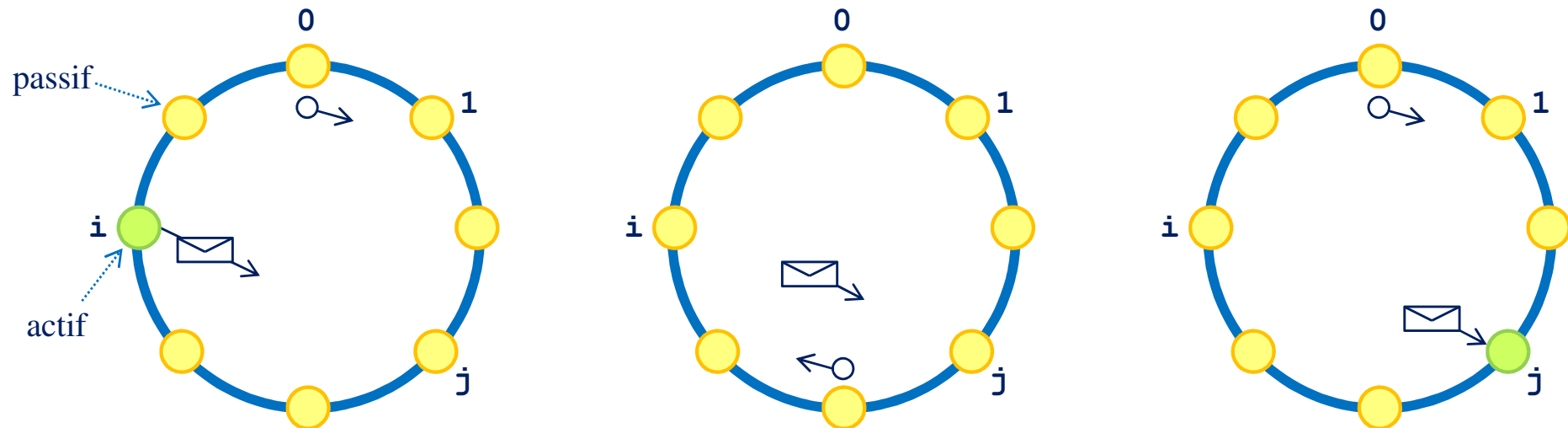
Détection de la terminaison dans le cas asynchrone : Algorithme de Misra (1983)

- algorithme basé sur la circulation d'un jeton,
- aucun site ne joue un rôle privilégié,
 - chaque site peut lancer une détection,
 - les jetons sont identifiés par l'identité du site initiateur.
- le réseau est fortement connexe
 - sa topologie peut être quelconque,
 - on construit alors un circuit qui passe au moins une fois par chaque nœud et chaque arc du réseau.
- le délai de transfert des messages est quelconque,
- on suppose qu'il n'y a pas de perte de messages, et pas de déséquencelement.

Algorithme de Misra

Principe de l'algorithme

- vérifier que tous les sites sont restés en permanence passifs entre deux tour du jeton,
En effet, après un tour où il a vu tous les sites passifs, le jeton ne peut pas conclure sur la terminaison : certains sites ont pu être réactivés, ou il peut y avoir des messages en transit.



Par contre, si le jeton fait un deuxième tour et s'aperçoit qu'aucun site n'est passé par l'état actif depuis sa dernière visite, alors il peut décider que le calcul est terminé.

En effet,

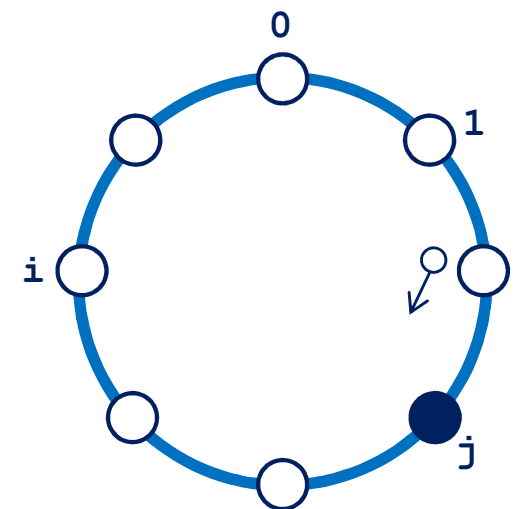
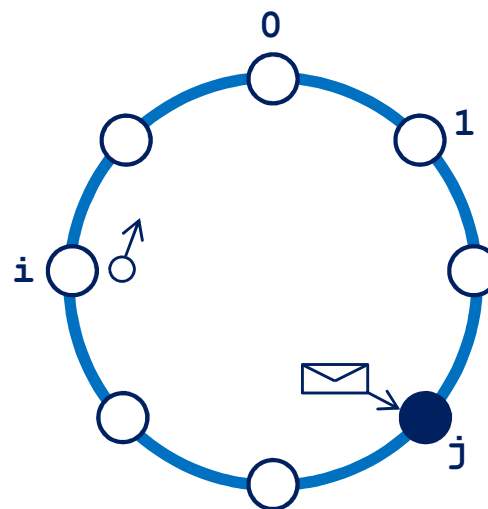
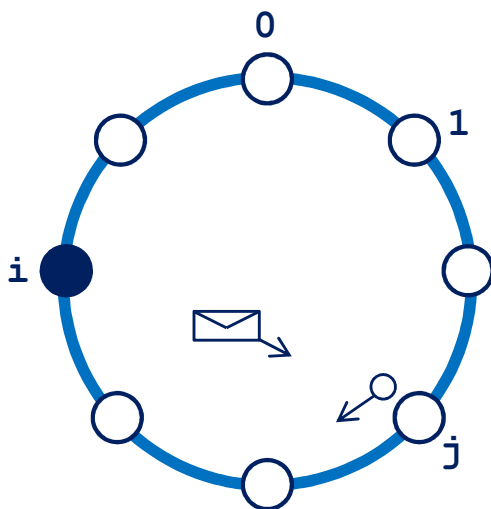
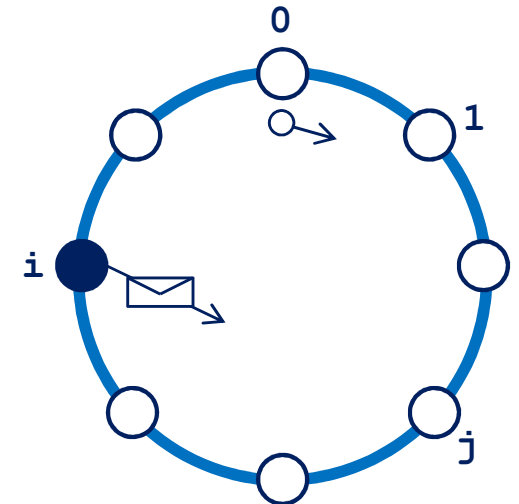
- aucun site n'a pu envoyer de message depuis sa première visite (ils sont tous restés passifs) et
- tous les messages qui auraient pu être envoyés avant sa première visite et en transit à ce moment sont, du fait de l'absence de déséquencement, forcément arrivés.

Tous les sites sont passifs et il n'y a pas de messages en transit : le calcul est donc terminé.

Algorithme de Misra

Comment savoir si un site est resté constamment passif entre deux passages du jeton ?

- utilisation d'une coloration des sites :
 - le jeton ne peut quitter un site que lorsque celui-ci est passif, et le site prend alors la couleur blanche,
 - si par la suite le site est réactivé, il prend la couleur noire
 - si le jeton retrouve le site blanc lors de son deuxième passage, c'est donc que celui-ci est resté passif en permanence.



Algorithme de Misra

Initialement :

- tous les sites sont actifs et noirs,
- on gère un compteur avec le jeton qui permet de savoir si celui-ci a effectué un tour complet du circuit.

L'algorithme est spécifié par les 6 règles suivantes :

R0 : Un site qui reçoit un message devient actif et noir.

R1 : Un site en attente de message est passif.

R2 : Tant qu'il est actif, le site **i** garde le jeton. Quand il est passif, il le transmet à son successeur sur le circuit.

R3 : Un site noir met le compteur du jeton à zéro avant de le transmettre à son successeur. Un site blanc incrémente le compteur avant de transmettre le jeton.

R4 : Après avoir transmis le jeton, un site devient ou reste blanc.

R5 : Un site blanc qui reçoit le jeton avec un compteur à **n** détecte la terminaison.

La terminaison est détectée lorsque le jeton a visité les **n** processus et les a tous trouvés blancs.

Références

Cours d'Algorithmique Répartie, "Propriétés Stables et Terminaison", Claude Dutheillet et Claude Girault, Laboratoire Lip6, Université Paris 6.

P. Blanc "Détection de Propriétés de Repos Globales dans les Systèmes Répartis avec Déséquence de Messages. Application au Problème de Terminaison", Thèse de Doctorat de l'Université Paris 6, mars 1990

M. Raynal "Algorithmes Distribués et Protocoles", Eyrolles, 1985

G. Tel "Introduction to Distributed Algorithms" Second Edition Cambridge University Press. ISBN 0-521-79483-8. 2000