

Auto-stabilisation

(Self-stabilization)

Colette Johnen

L.R.I./C.N.R.S.

Université de Paris-Sud

Bat. 490, Campus d'Orsay

F-91405 Orsay Cedex, France.

tel: (+33) 1 69 41 66 29

fax: (+33) 1 69 41 65 86

colette@lri.fr

<http://www.lri.fr/colette/>

Systeme auto-stabilisant

Dijkstra (74) : un systeme est auto-stabilisant quand **indépendamment** de son état initial, le systeme arrivera inévitablement à un état **légitime** en un nombre fini d'étapes.

Lamport (83) : "I regard this as Dijkstra's most brilliant work - at least, his most brilliant published paper. It's almost completely unknown. I regard it to be a milestone in work on fault tolerant."

Clôture

Un **système S** est composé d'un ensemble d'états (E) et d'un ensemble de transitions ($T : E \rightarrow E$).

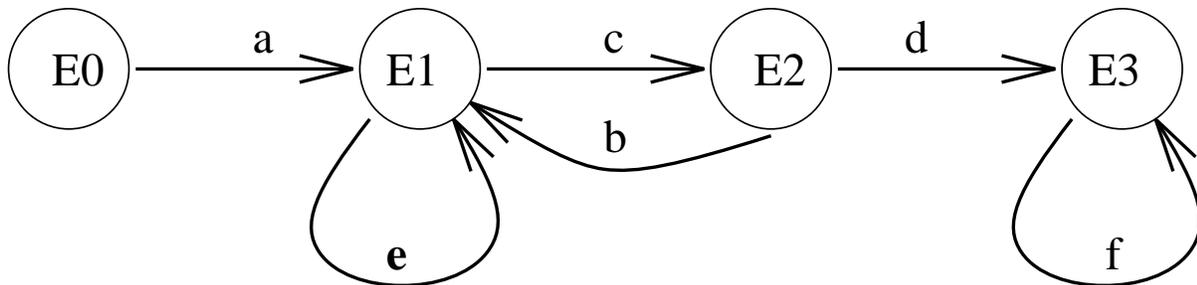
Soit L un sous ensemble de E .

Faible clôture de L pour S :

Toute exécution à partir de L , a un suffixe, non vide, où chaque état atteint est dans L .

Forte clôture de L pour S :

Chaque transition à partir d'un état de L atteint un état de L .



$\{E1, E3\}$ est faiblement clos et non fortement.

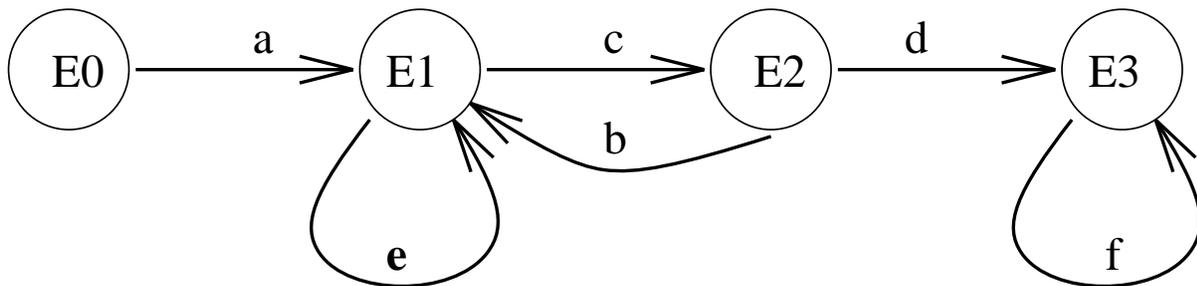
Convergence

Un **système S** est composé d'un ensemble d'états (E) et d'un ensemble de transitions ($T : E \rightarrow E$).

Soit L un sous ensemble de E .

Faible convergence : à partir de tout état atteint lors d'une exécution, il existe une exécution permettant d'atteindre un état de L .

Forte convergence : toute exécution de S atteint un état de L .



Le système converge faiblement vers $\{E3\}$ et non fortement.

Systeme auto-stabilisant

Un **systeme S** est compose d'un ensemble d'etats (E) et d'un ensemble de transitions ($T : E \rightarrow E$).

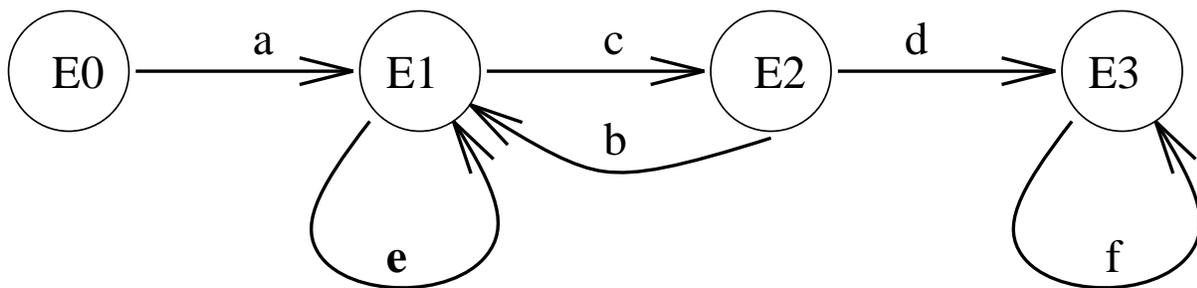
Soit L un sous ensemble de E .

Le systeme (S, T) s'auto-stabilise fortement (faiblement) sur l'ensemble d'etats fortement (faiblement) clos L ssi

- Clôture : L est un ensemble d'etats fortement (faiblement) clos pour S .
- Convergence : toute execution de S converge fortement (faiblement) vers L .

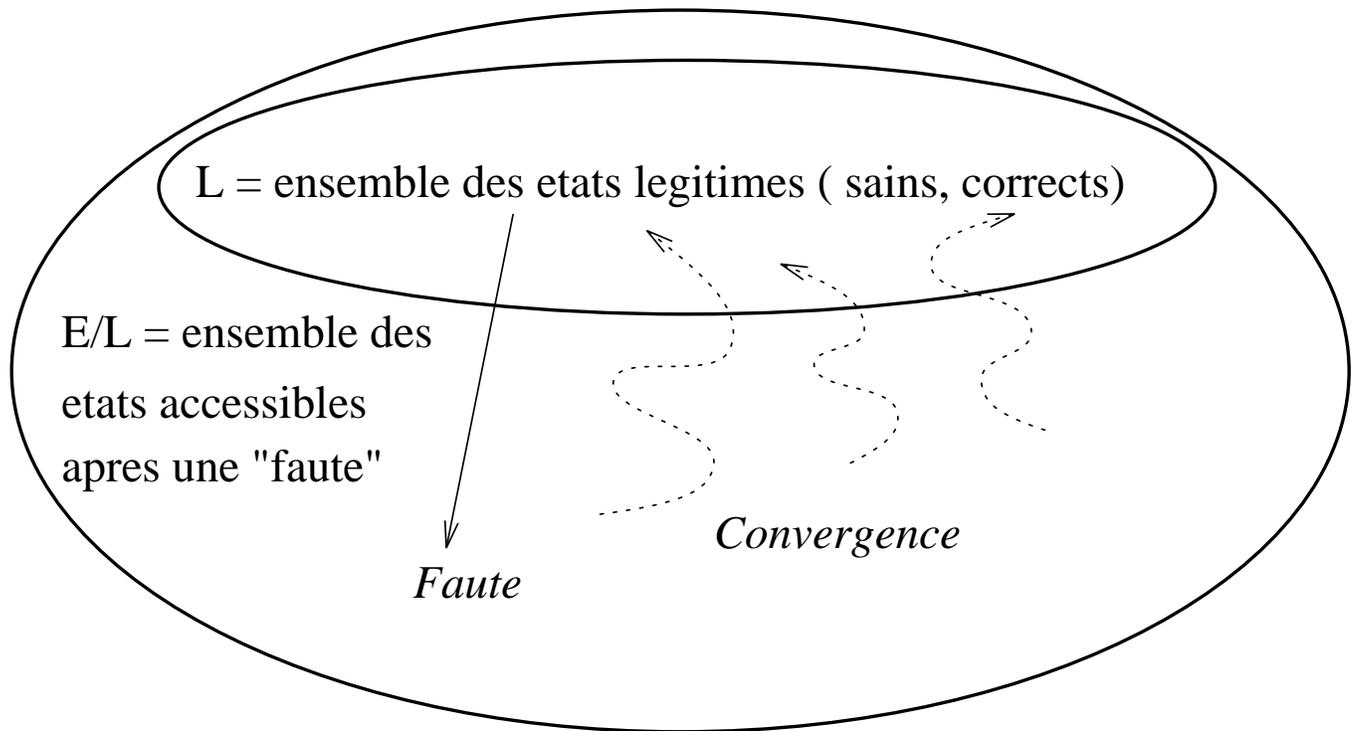
L est un **attracteur** ssi le systeme converge fortement vers L et L est un ensemble fortement clos

standard : auto-stabilisation forte sur un ensemble fortement clos.



Le systeme ne s'auto-stabilise pas sur $\{E3\}$.

Auto-stabilisation et tolérance aux pannes



"pannes" qui corrompent la mémoire, mais laissent inchanger les "exécutables" :

- pertes ou duplications de message,
- messages erronés,
- corruption de la valeur des variables par des parasites (maille associative),
- changement dans la connectivité du réseau.

Avantages et inconvénients de l'auto-stabilisation

Approche **Optimiste** de la tolérance aux pannes :

- *peu de pannes,*
- *assez de temps entre les pannes pour stabiliser le système.*

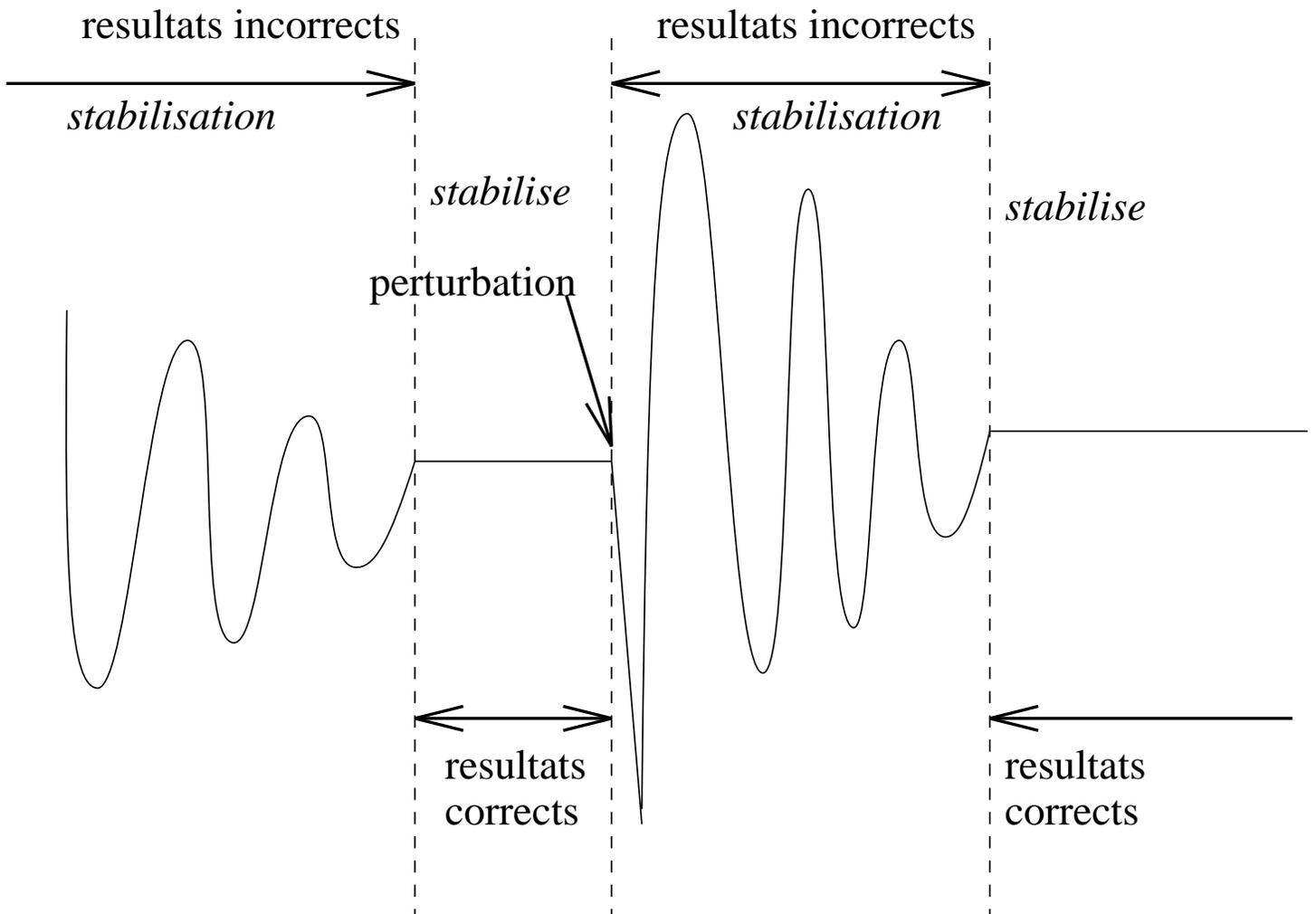
Avantages :

- *pas de contrôle systématique des données en entrée ;*
- *adapté à toute panne qui corrompt la mémoire ;*
- *pas d'initialisation du système ;*
- *rétablissement automatique en cas de panne ;*
- *s'adapte au changement de la configuration du réseau (réseau dynamique).*

Inconvénients :

- *vérification de la propriété d'auto-stabilisation ;*
- *fragilité de la propriété auto-stabilisation ;*
- *temps de stabilisation ;*
- *détection de la terminaison de la stabilisation.*

Auto-stabilisation : validité des résultats



Auto-stabilisation et système réparti

- *Un système est un réseau de processeurs interconnecté.*

- *Un état du système est défini par l'état local de tous les processeurs (nœuds).*

un état d'un nœud est la valeur de ses variables.

- *L'algorithme est proposé sous la forme d'un ensemble de **règles**.*

** chaque règle a deux parties :*

- le **privilege** qui est une fonction booléenne de

- l'état du nœud

- l'état de ses voisins

- le **mouvement** : changement de l'état du nœud.

=> Chaque processeur a une vision partielle du système.

Protocole uniforme ou semi-uniforme

- *Le protocole est **uniforme** si tous les processeurs exécutent le même algorithme.*
- *Le protocole est **semi-uniforme** si tous les processeurs exécutent le même algorithme sauf un nœud.*

Théorème : *il n'existe pas de protocole déterministe d'exclusion mutuelle auto-stabilisant sur un anneau de taille non premier qui soit uniforme.*

Protocole séquentiel, synchrone ou asynchrone

Protocole synchrone (démon réparti)

Le calcul est fait par étapes successives synchronisées :

- *évaluation des privilèges*
- *exécution par certains processeurs d'une règle dont ils satisfont le privilège.*

Le démon choisit les processeurs et pour chaque processeur, la règle à effectuer.

Protocole asynchrone : (démon complètement réparti, write/read atomicity)

Chaque processeur, de manière asynchrone : évalue ses privilèges, exécute une règle dont il satisfait le privilège.

=> le mouvement d'une règle peut être effectué alors que le privilège de la règle n'est plus satisfait.

Protocole séquentiel : (démon centralisé)

Le calcul est fait par mouvements successives.

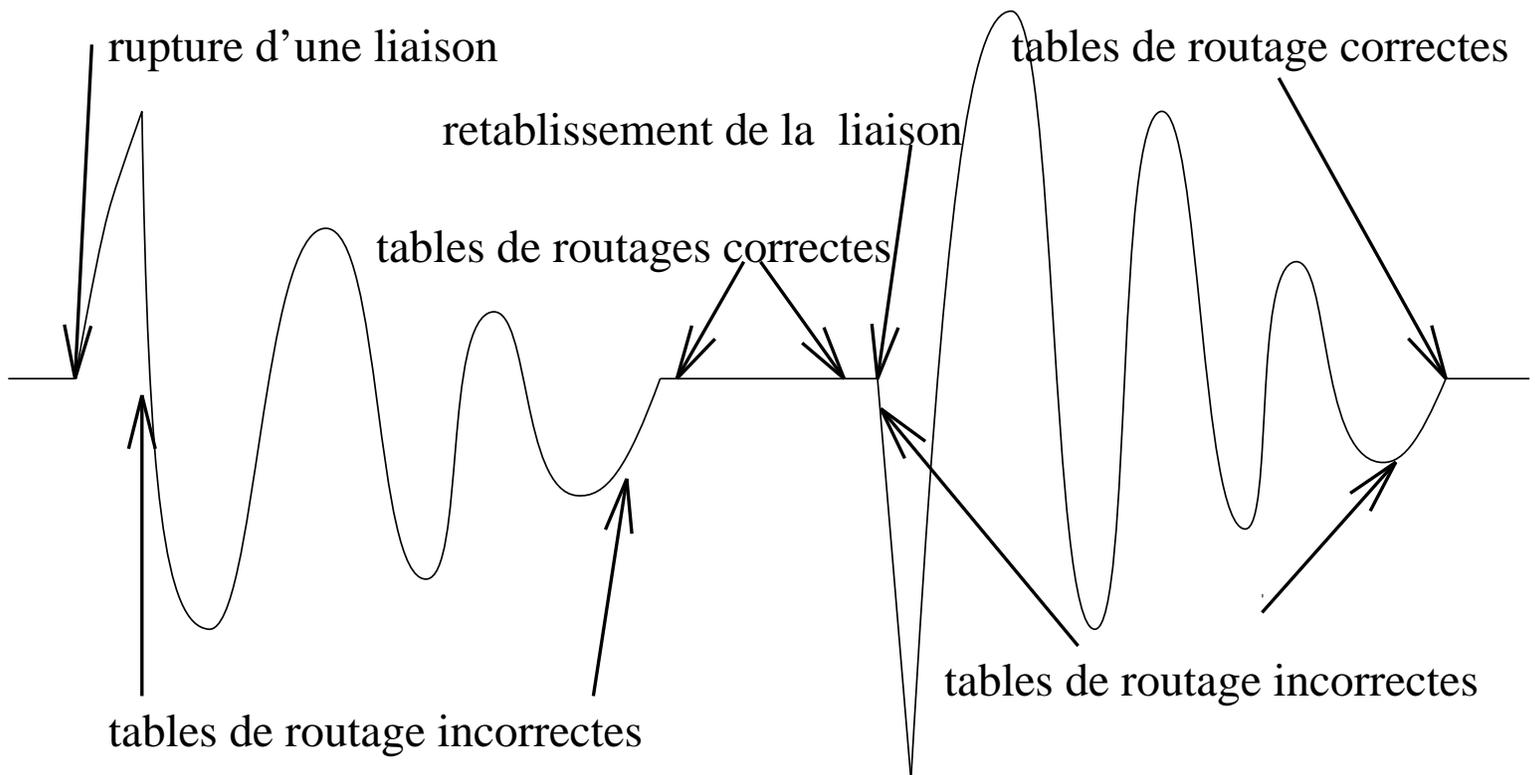
Auto-stabilisation et Internet

Les algorithmes de routage dans Internet

(ref: *Huitema 95: Le routage dans l'Internet, ed. Eyrolles*).

- *protocole à vecteurs de distance (RIP)*
- *protocole à état des liaisons (OSPF)*
- ...

sont des algorithmes auto-stabilisants.



Validite des tables de routage du reseau

Bibliographie

Dijkstra 74 : Self-stabilizing systems in spite of distributed control, *Communications of the ACM*, Vol. 17, No. 11, 1974.

Lamport 84 : Solved problems, unsolved problems, and non-problems in concurrency, in *Proceedings of the 3rd Symposium on Principles of Distributed Computing*, 1984.

Schneider 93 : Self-stabilization *ACM Computing Surveys*, Vol. 25, No. 1, 1993.

Gouda 95 : The triumph and tribulation of system stabilisation, *LNCS 972, WDAG'95*, 1995.