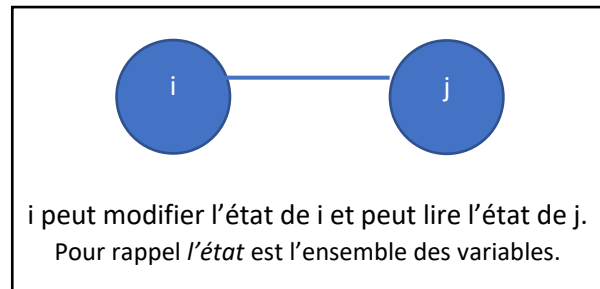


TD

Rappels

Modèle à Etats :



Un algorithme écrit dans le modèle à état s'écrit en fait sous la forme de règles gardées.

Une règle gardée : <conditions> : instructions

Une exécution d'un algorithme est structurée par un ordonnanceur (= démon).

L'ordonnanceur **choisit** une ou plusieurs règles dont la garde (= la condition) est évaluée à vraie et déclenche l'exécution des instructions associées.

- Si une règle est **infiniment continument** déclenchable elle finira par être choisie par un démon **faiblement équitable**.
- Si une règle est **infiniment** déclenchable elle finira par être choisie par un démon **fortement équitable**.
- Un démon **non équitable** peut ne **pas choisir** une règle **infiniment continument déclenchable**.

Algorithme de Dijkstra (1974)

Variable :

M_i : entier

Règles :

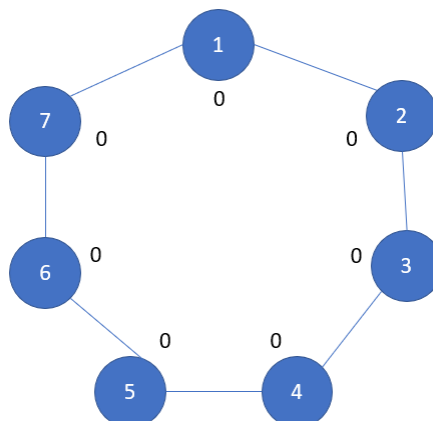
$\forall i, 2 \leq i \leq n, M_i \neq M_{i-1} : M_i \leftarrow M_{i-1}$

$i = 1, M_1 = M_n : M_1 \leftarrow (M_1 + 1) \% k \text{ avec } k > n$

Application

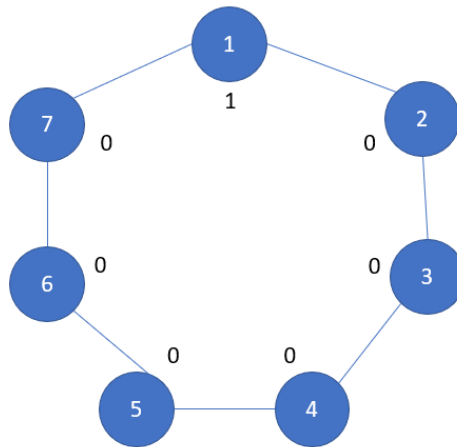
Enoncé

Avec $k = 9$

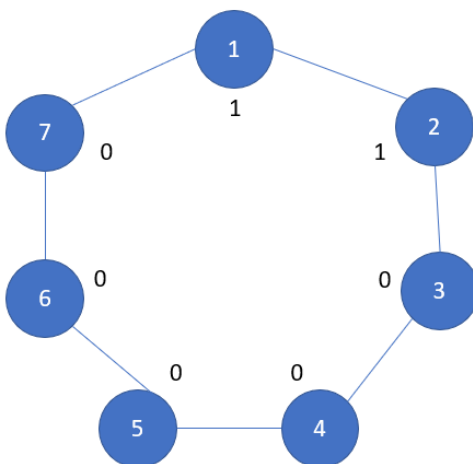


Déroulement de l'algorithme

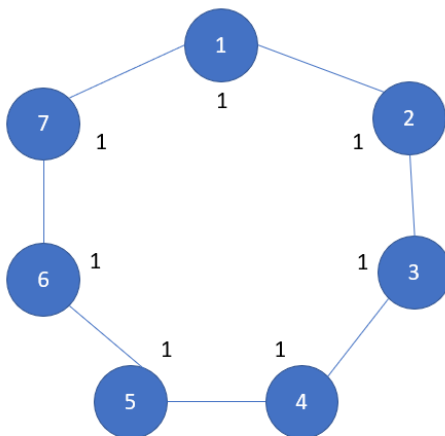
On peut appliquer la deuxième règle :



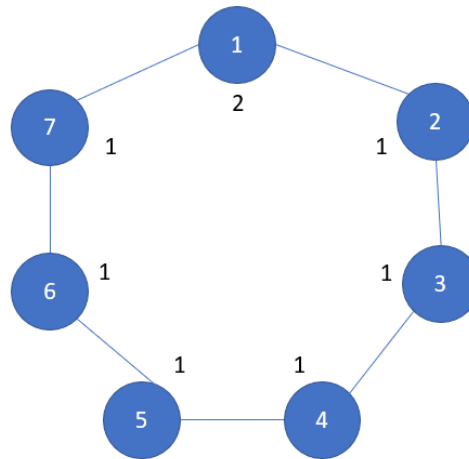
On regarde les règles déclenchables, ici la première est déclenchable :



La première est toujours déclenchable :



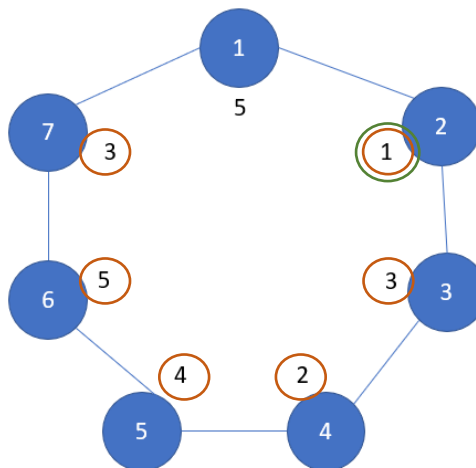
Et à nouveau la deuxième est déclenchable :



Remarques :

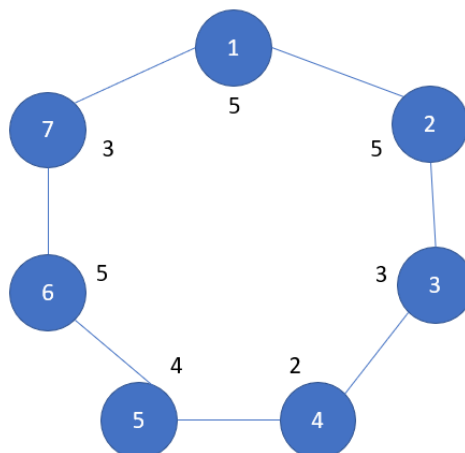
Cet algorithme est plus évolué que l'algorithme de Le Lann.

Dérouler l'algorithme suivant :

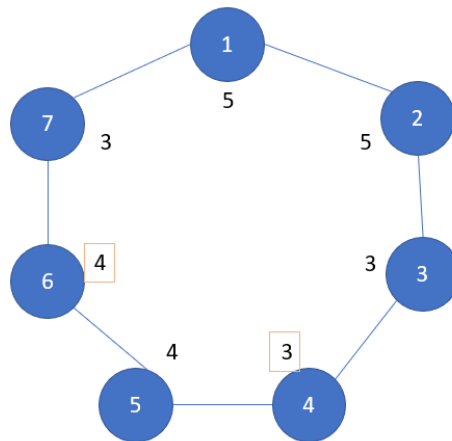


En **orange** les états déclenchables* au départ. On choisit de déclencher 1 état seulement (celui-ci). On aurait pu tous les déclencher ou seulement quelques un.

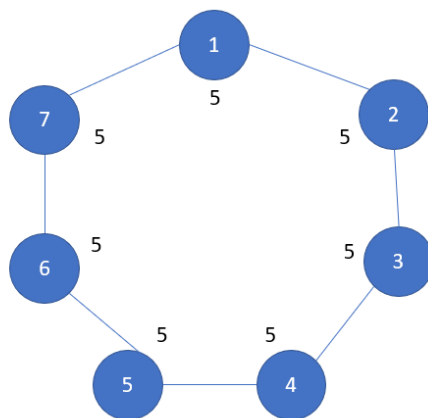
On applique la première règle (*la deuxième n'étant pas déclenchables puisque la valeur de 1 est 5, et la valeur de 7 est 3) :



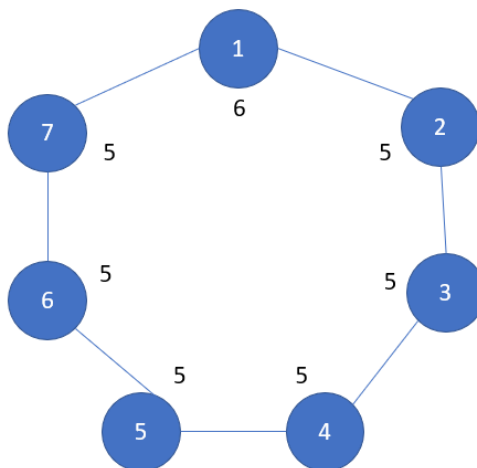
On peut en déclencher plusieurs en même temps comme nous l'avons dit précédemment :



Ici on peut continuer à appliquer la première règle. On obtient après plusieurs tours :



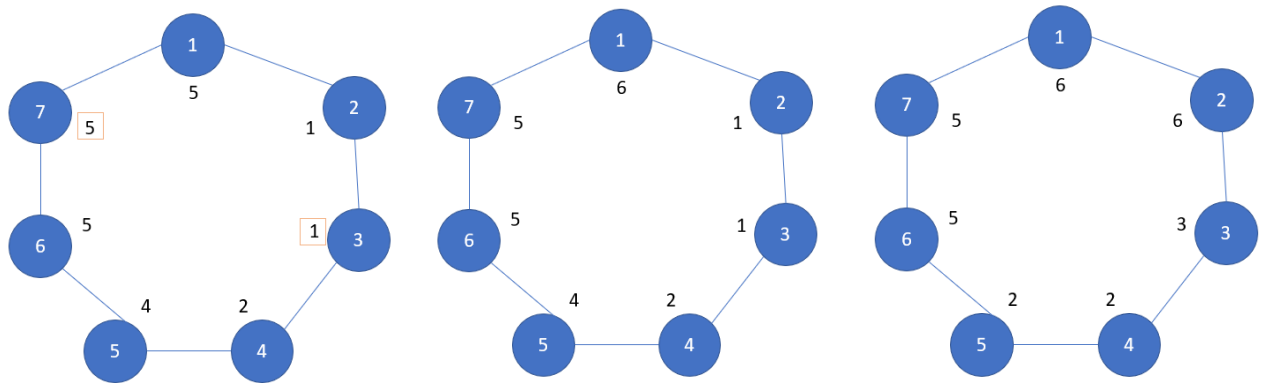
On peut appliquer la deuxième règle (car $1 = 5$ et $n = 7 = 5$) :



Etc.

Bilan : on peut appliquer plusieurs règles au « même tour » ou une seule. La règle deux n'est pas l'initialisation, elle peut s'appliquer si les conditions sont présentes et si le démon choisit de l'appliquer.

Autre exemple de déroulement :



Bilan

On remarque qu'au bout d'un moment on a toujours qu'une seule règle qui est déclenchable en même temps. Quelque soit l'initialisation on revient à un mode de fonctionnement où tout se passe bien.

C'est cette dernière propriété qui rend cet algorithme « meilleur » que l'algorithme de Le Lann.

Cet algorithme part d'un état arbitraire et arrive à arriver à une configuration où un seul site est déclenchable.

Cela implique que si on introduit des erreurs l'algorithme arrive à **se corriger lui-même**.

C'est un algorithme **auto stabilisant**. Il gère les fautes de manière automatique, sans intervention extérieure.

Algorithme auto stabilisant :

- La **convergence** : quelque soit l'état initial de l'algorithme, dans l'exécution future, on obtient une configuration légale (= qu'une seule règle qui possible à la fois).
- La **clôture** : « si je suis en configuration légale je resterai en configuration légale ».