Automates de sable

Enrico FORMENTI, Benoît MASSON

Laboratoire I3S - Université de Nice

Introduction

- Nouveau modèle, “croisement” entre systèmes existants :
  - automates cellulaires ;
  - tas de sable.
- On s’intéresse à la dynamique de ce système dynamique discret.
  - Comportement défini par des règles simples, locales.
  - Quel sera le comportement global à long terme ?
    - Conservation des grains
    - Stabilisation
    - Nilpotence

Plan

Introduction

1. **Premiers pas**
2. Automates de sable
3. Quelques résultats

Conclusion

Automates cellulaires

➢ Ensemble d'états $S = \{\square, \blacksquare\}$.

➢ Règle locale $\delta : S^{2r+1^d} \mapsto S$, rayon r dimension d.

↪ Cas de base $d = r = 1$:

■■■	■■□	■□■	■□□	□■■	□■□	□□■	□□□
□	□	□	■	□	■	■	□

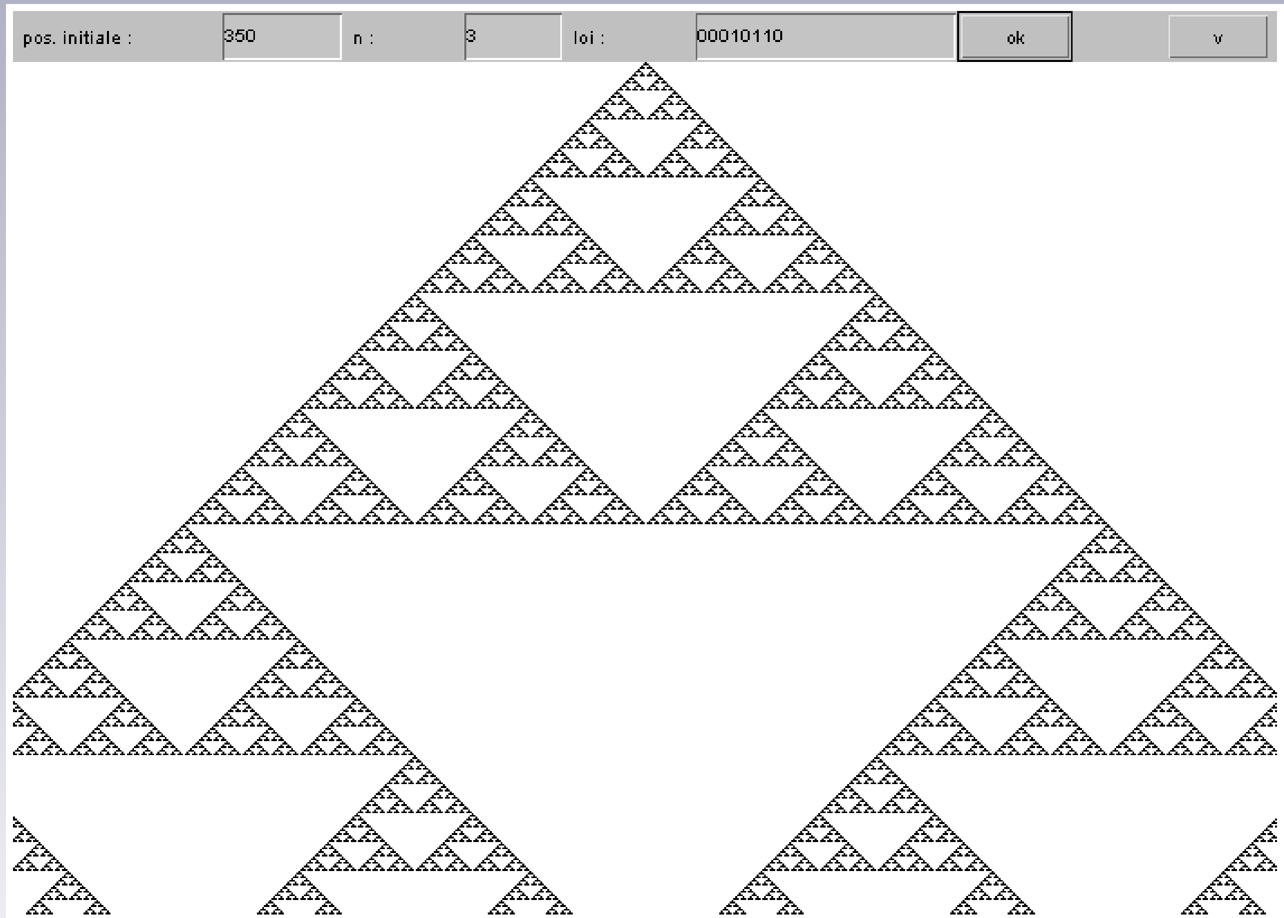
➢ Appliqué à une configuration de $S^{\mathbb{Z}^d}$.

$$\cdots □■■■□■□□□■■□■□■□■■ \cdots$$

$$\downarrow$$

$$\cdots □□□□■■□■□□□■□■□□ \cdots$$

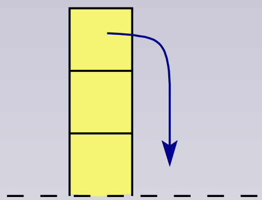
Diagramme espace-temps



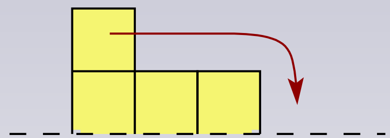
Source : `http://perso.wanadoo.fr/jpq/fractales/autocell/`

Tas de sable

- Associé à un entier $n \in \mathbb{N}$ (nombre de grains).
- Etats possibles : $\{0, \ldots, n\}$.
- Règles locales très simples.

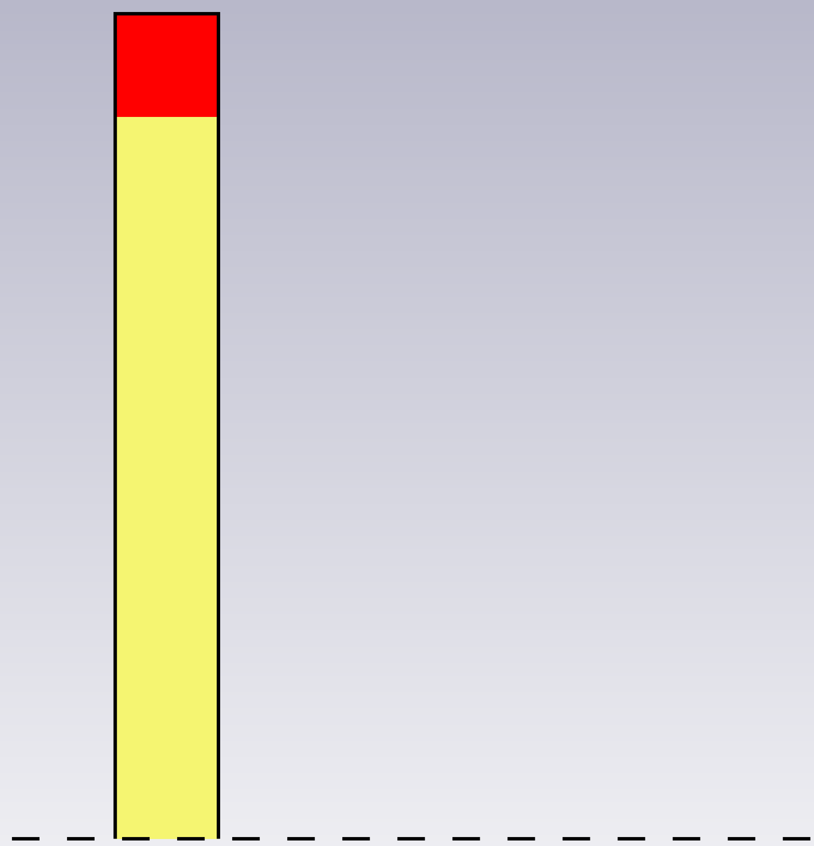


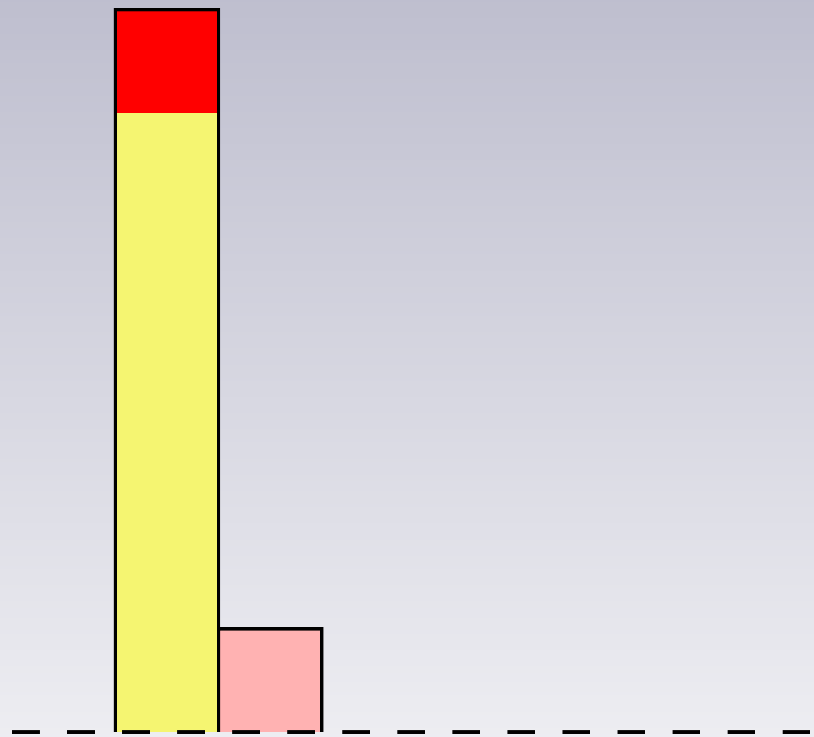
Règle verticale

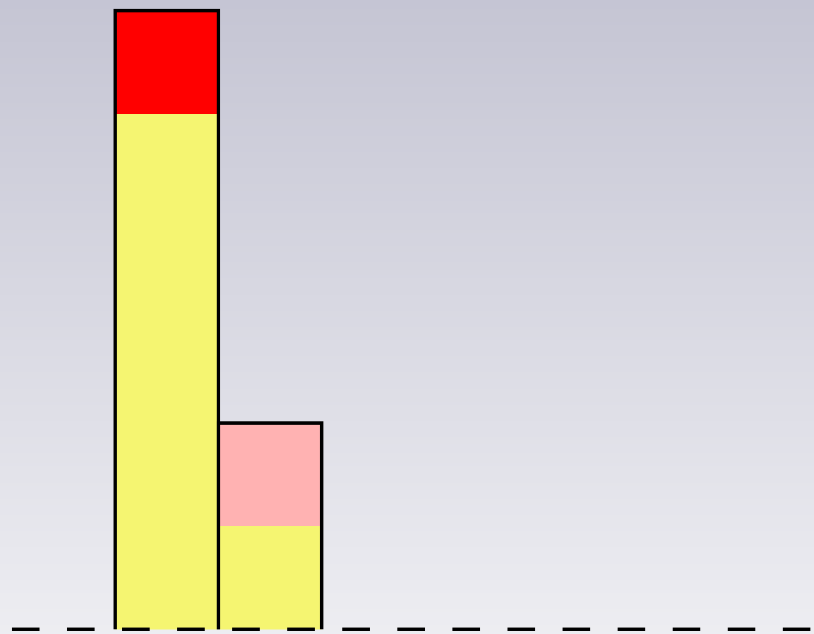


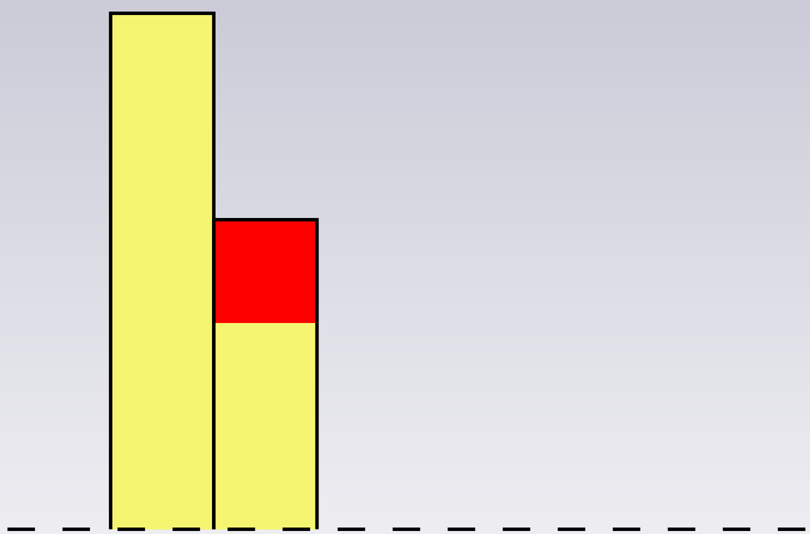
Règle horizontale

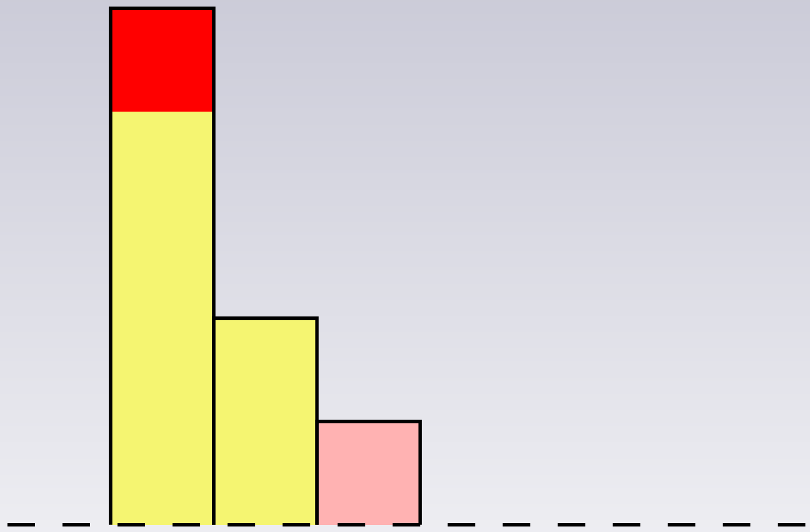
- Deux modèles utilisés principalement :
  - SPM (*Sand Pile Model*), règle verticale ;
  - IPM (*Ice Pile Model*), règles verticale et horizontale.

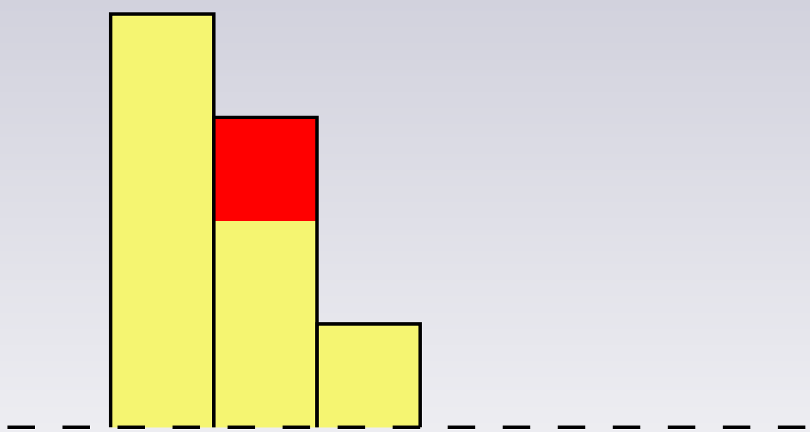


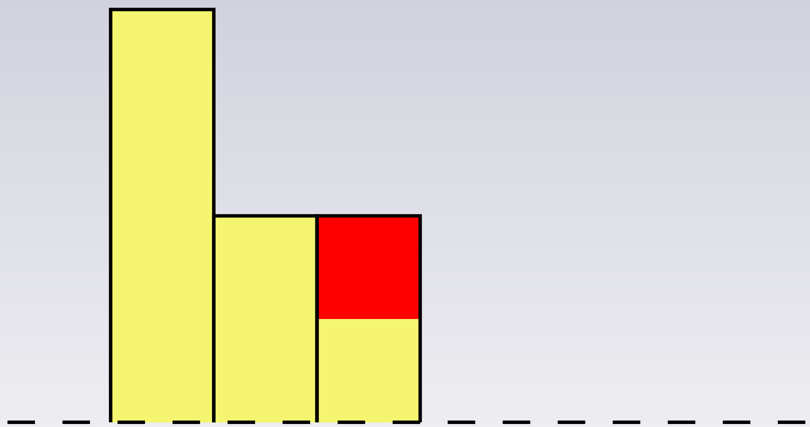




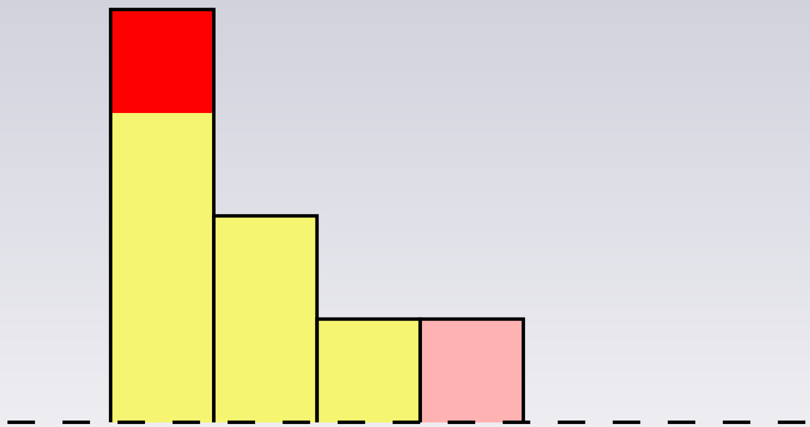


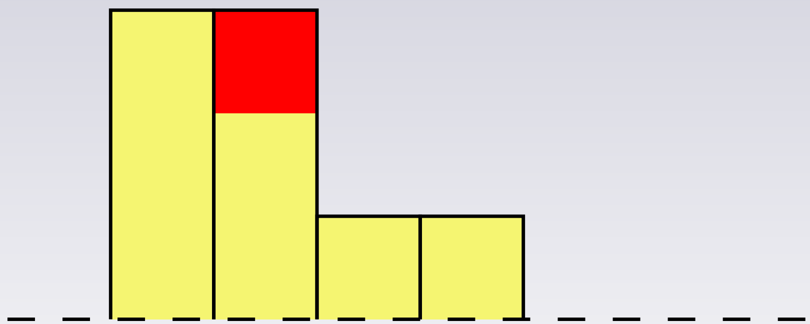




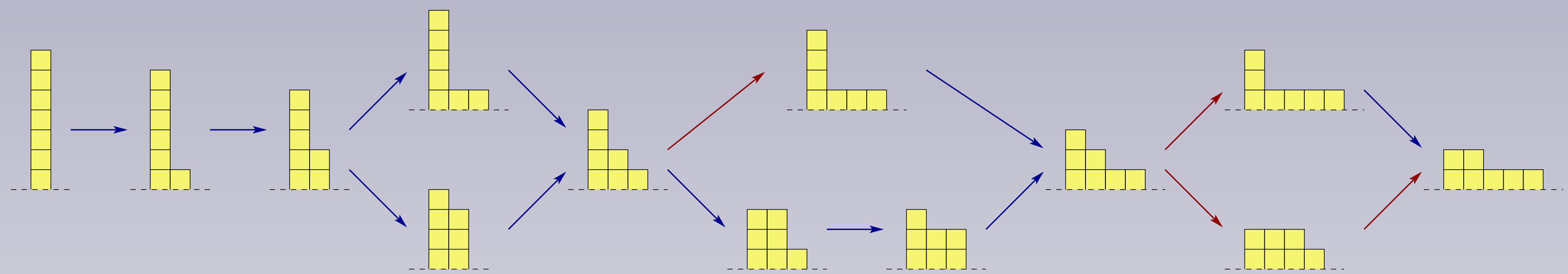


SPM





Evolution de IPM



➢ Treillis $\Rightarrow$ unicité du point fixe.

Récapitulatif

Automates cellulaires	Tas de sable
configurations infinies	configurations finies
nombre d'états borné	nombre d'états non borné
évolution synchrone	évolution asynchrone
règle locale	règle locale

Plan

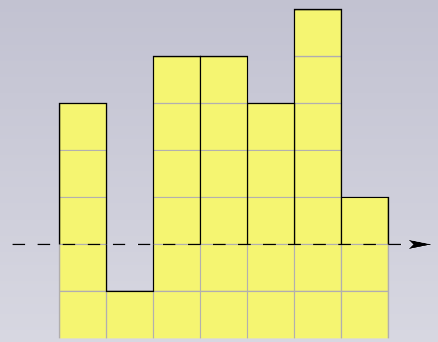
Introduction

1. Premiers pas
2. **Automates de sable**
3. Quelques résultats

Conclusion

Configurations

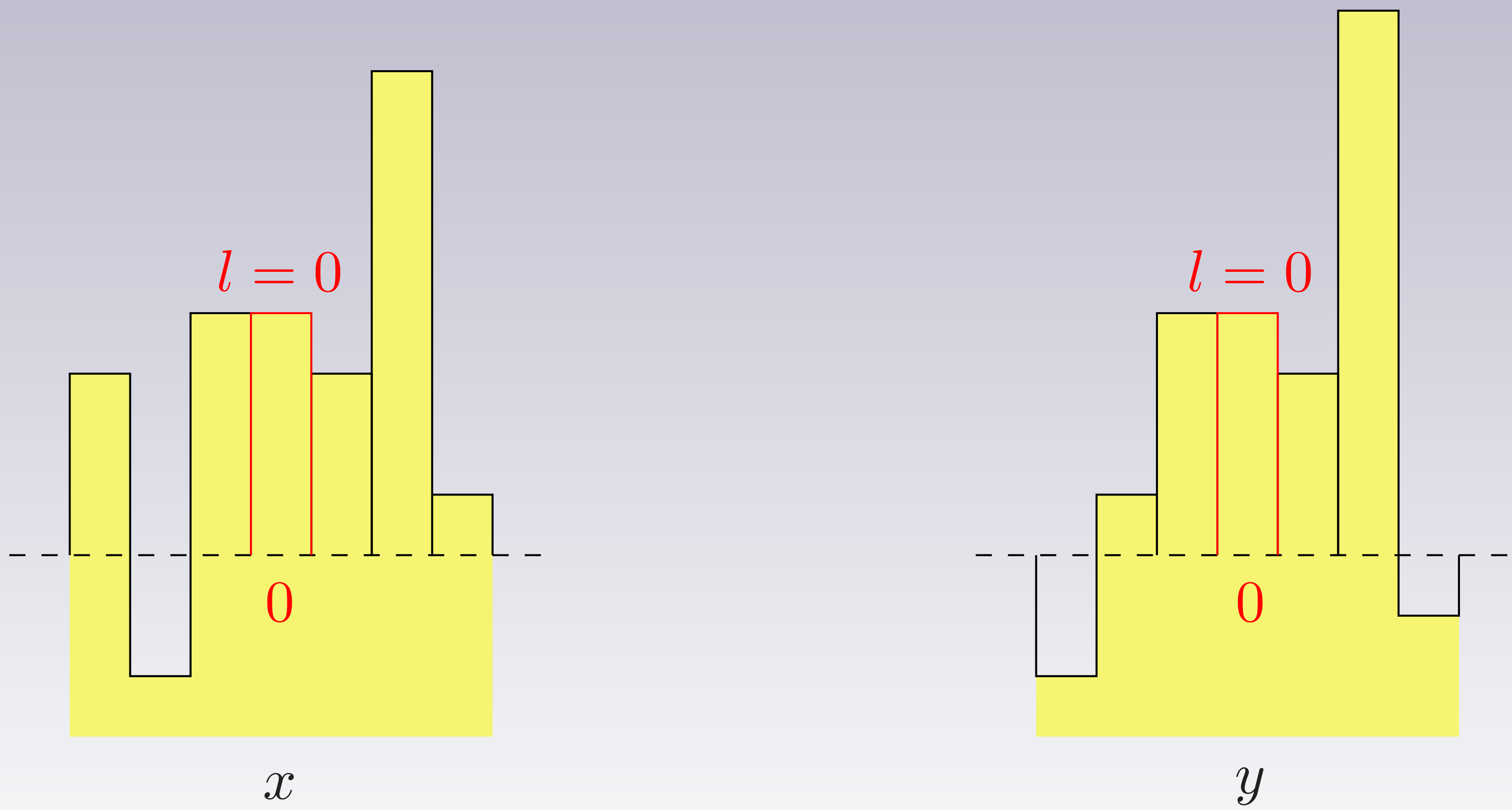
- En dimension 1, une configuration est un élément de $\mathbb{Z}^{\mathbb{Z}}$.
- A tout point est associé un entier indiquant le nombre de grains.



- Ajout de sources et puits pour plus de généralité et pour des raisons de compacité : $\left(\mathbb{Z} \cup \{-\infty, +\infty\}\right)^{\mathbb{Z}}$
- Besoin d'une topologie pour pouvoir étudier la dynamique.
  - Mise en place d'une distance.

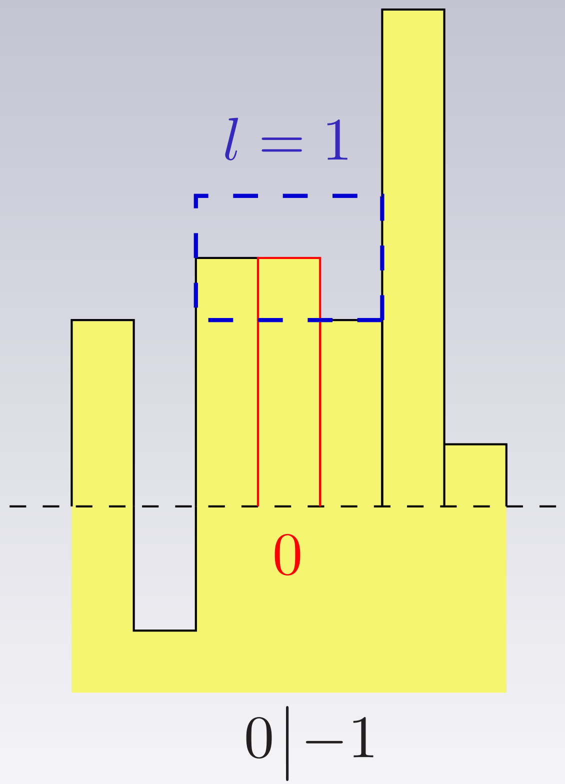
Distance

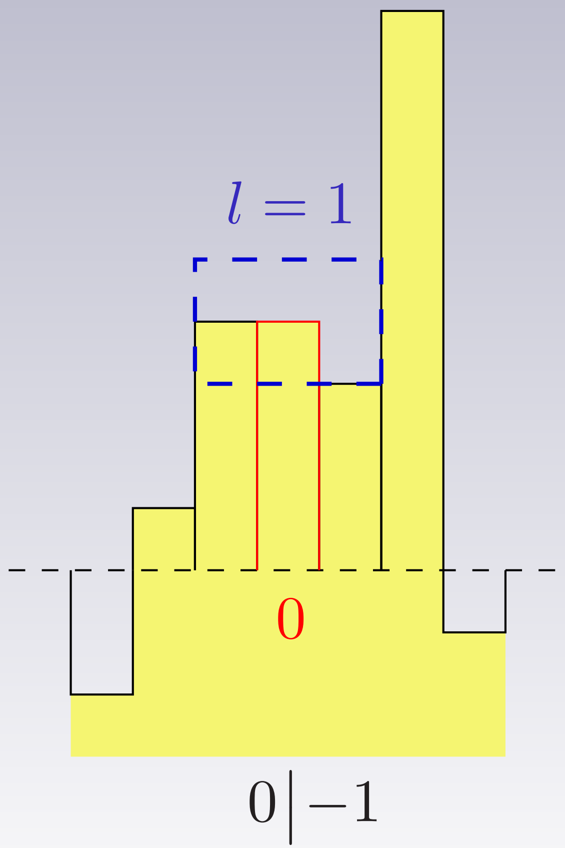
- $d(x,y) = 2^{-l}$, l défini par :



Distance

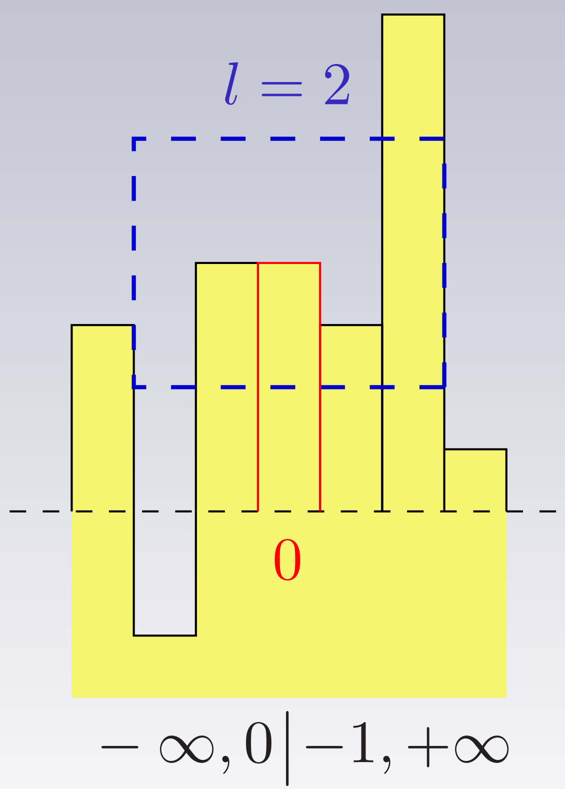
➢ $d(x, y) = 2^{-l}$, l défini par :

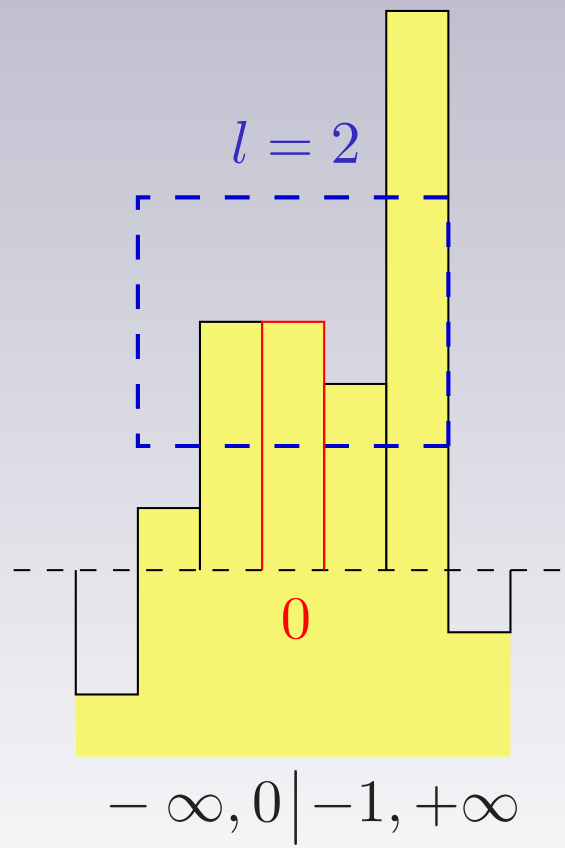




Distance

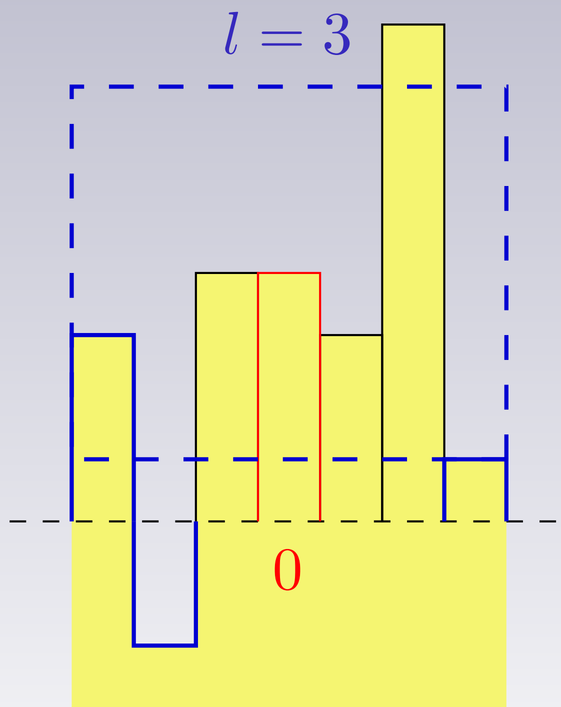
- $\boxed{d(x,y) = 2^{-l}}$, l défini par :





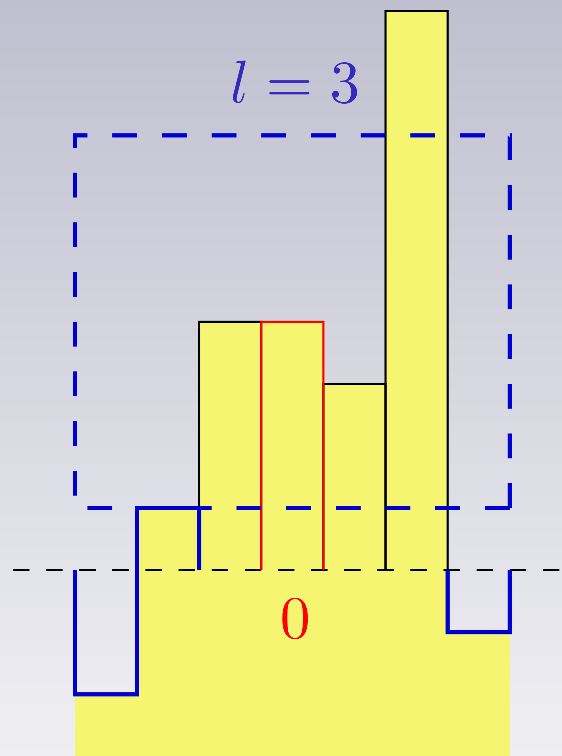
Distance

- $\boxed{d(x,y) = 2^{-l}}$, l défini par :



$-\mathbf{1}, -\infty, 0 \,|\, -1, +\infty, -\mathbf{3}$

$d(x,y) = 2^{-3}$



$-\infty, -\mathbf{3}, 0 \,|\, -1, +\infty, -\infty$

Topologie

Proposition 1

L'espace des configurations $\mathfrak{C}$ n'est pas compact.

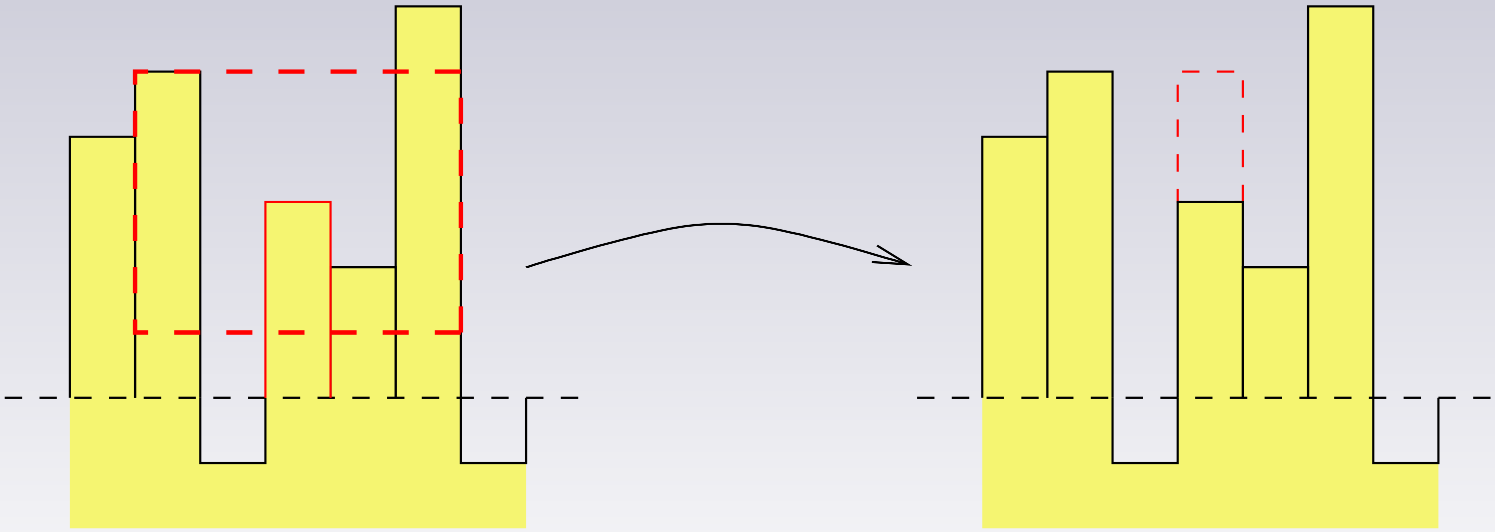
Théorème 2

$\mathfrak{C}$ est localement compact.

- Plus précisément, $\mathfrak{C}_x = \{c \in \mathfrak{C} | c_0 = x\}$ est compact.

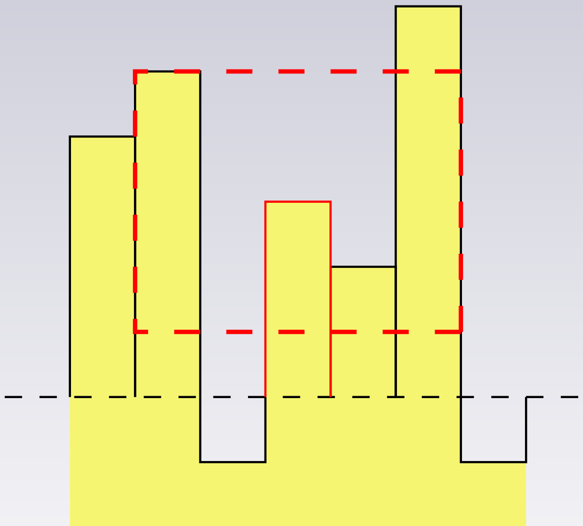
Automates de sable

- $\mathcal{A} = \langle r, \lambda \rangle$, r est le rayon et λ la règle locale.
- La règle locale retourne la variation d'une colonne en fonction du tableau des différences de hauteur avec les voisins.



Règle locale

- “Locale” à cause du rayon r :
  - nombre de voisins pris en compte fixe ;
  - différences bornées, considérées comme infinies si trop grandes.



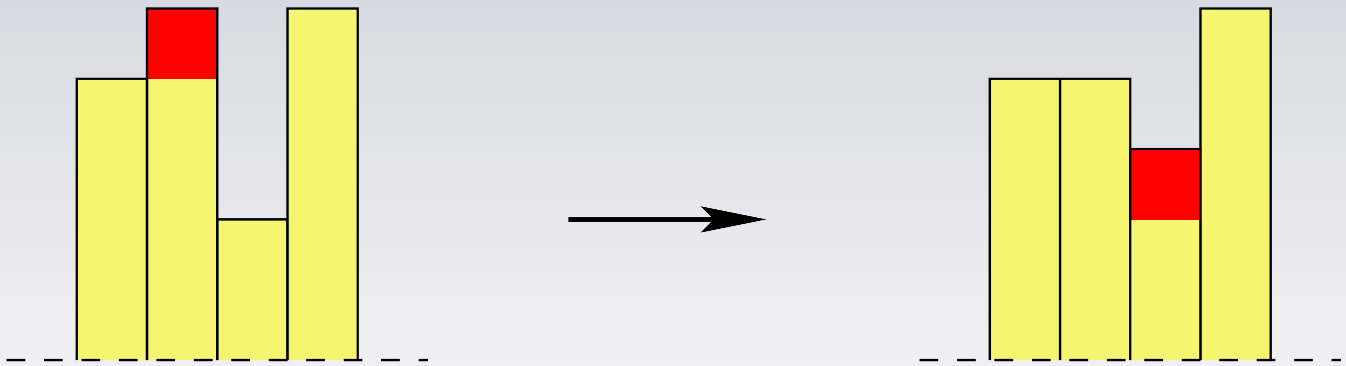
La règle est appliquée sur le voisinage $(+2, -\infty, -1, +\infty)$.

Règle globale

- La règle locale est appliquée de manière synchrone, alors le comportement de l'automate est déterminé par sa règle globale $f : \mathfrak{C} \mapsto \mathfrak{C}$.
- $$f(c)_i = \begin{cases} c_i + \lambda(N(c_i)) & \text{si } |c_i| < \infty \\ c_i & \text{sinon.} \end{cases}$$
- La règle globale est continue dans l'espace métrique $\mathfrak{C}$.

Simulation de SPM

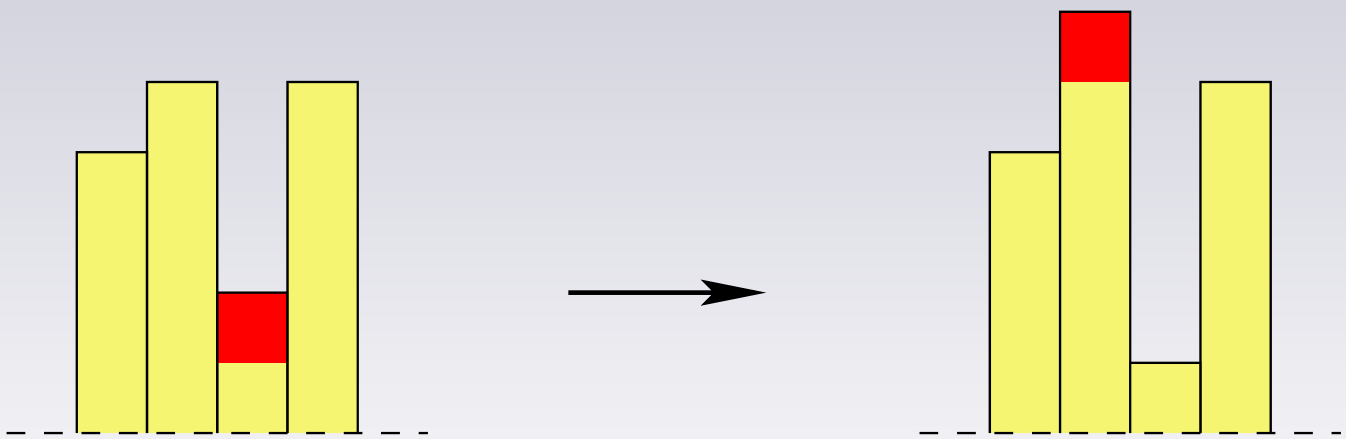
$$r = 1, \quad \lambda(a,b) = \begin{cases} +1 & \text{si } a = +\infty \text{ et } b \neq -\infty \\ -1 & \text{si } a \neq +\infty \text{ et } b = -\infty \\ 0 & \text{sinon.} \end{cases}$$



$$a = -1, \quad b = -\infty$$

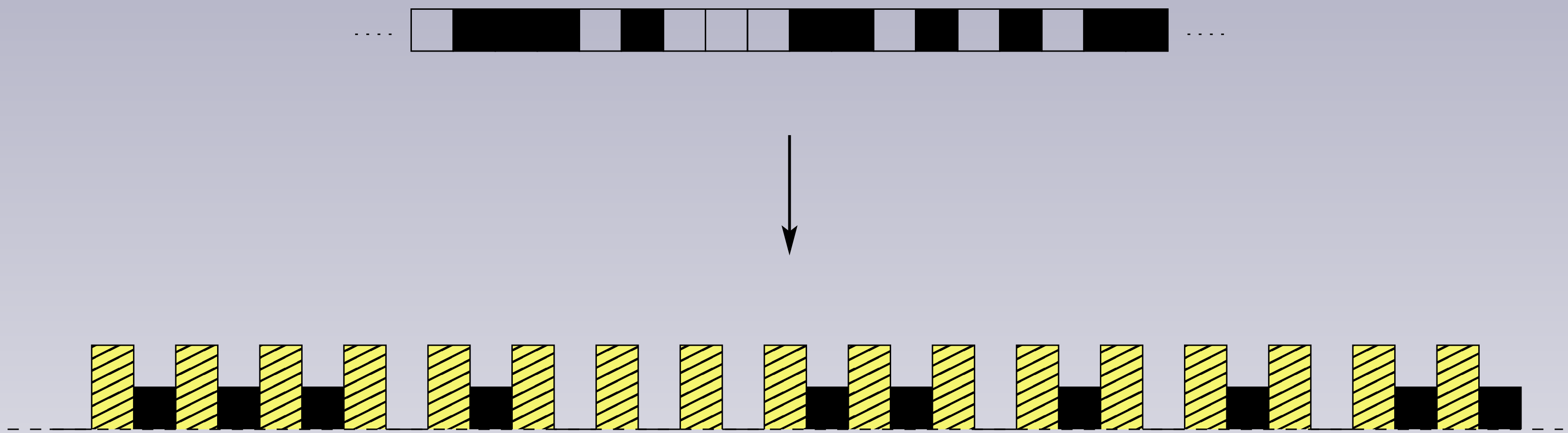
SPM$^{\mathrm{r}}$ (inverse à droite)

$$r = 1, \quad \lambda(a,b) = \begin{cases} -1 & \text{si } a = +\infty \text{ et } b \neq -\infty \\ +1 & \text{si } a \neq +\infty \text{ et } b = -\infty \\ 0 & \text{sinon.} \end{cases}$$



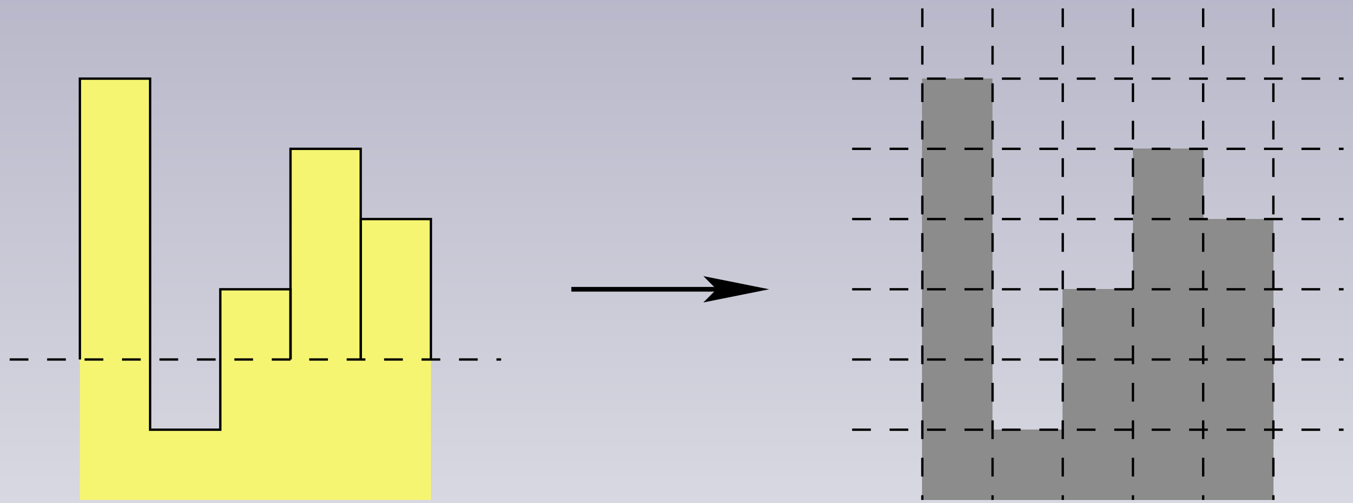
$a = -1, \quad b = -\infty$

Simulation d'un AC par un AS



➠ Rayon $r \longrightarrow 2r$.

Simulation d'un AS par un AC



- Rayon $r \longrightarrow 2r$.
- Dimension $d \longrightarrow d+1$.

Plan

Introduction

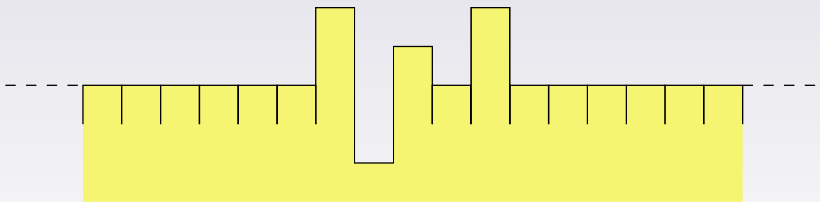
1. Premiers pas
2. Automates de sable
3. **Quelques résultats**

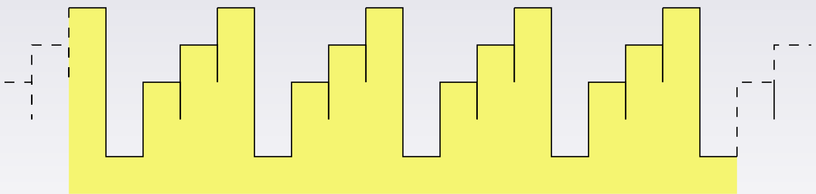
Conclusion

Propriétés ensemblistes

- Etude de la surjectivité et de l'injectivité d'un automate, *i.e.* de sa règle globale.
  - Souvent utilisé pour définir la chaoticité d'un automate.
  - Est-ce que ces propriétés sont décidables ?

- On recherche des relations entre injectivité et surjectivité.
  - Dans le cas général.
  - Restriction à des configurations finies et périodiques.





Théorème 3

$$\begin{tikzcd} I \arrow[rr, Rightarrow] \arrow[dr, Rightarrow] & & I_{\mathfrak{P}} \arrow[dl, Rightarrow] \\ & I_{\mathfrak{F}} & \end{tikzcd} \qquad \begin{tikzcd} S \arrow[rr, Rightarrow, shift left, "1"] & & S_{\mathfrak{P}} \arrow[ll, Rightarrow, shift left] \\ & S_{\mathfrak{F}} \arrow[ul, Rightarrow] \arrow[ur, Rightarrow, "1"'] & \end{tikzcd}$$

➠ La décidabilité reste ouverte.

Conservation des grains

Définition 1 (FGC)

Un automate conserve les gains sur les finis ssi pour toute configuration x finie, $\sum_{i \in \mathbb{Z}^d} f(x)_i = \sum_{i \in \mathbb{Z}^d} x_i$.

Définition 2 (PGC)

Un automate conserve les gains sur les périodiques ssi pour toute configuration x périodique de période p, $\sum_{i=(0,\ldots,0)}^{p} f(x)_i = \sum_{i=(0,\ldots,0)}^{p} x_i$.

Résultats

Proposition 4

Les définitions FGC et PGC sont équivalentes.

Théorème 5

La conservation des grains est décidable.

- En dimension 1, rayon 1, $\mathcal{A} = \langle 1, \lambda \rangle$ est GC ssi pour tous $a, b, c \in \mathbb{Z}$,

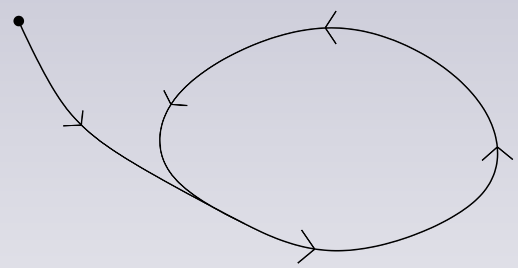
$$\lambda(a,b,c) = \lambda(0,0,b) - \lambda(0,0,a) + \lambda(0,b,c) - \lambda(0,a,b) \ .$$

Périodicité ultime

Problème ULT

Instance: Un automate de sable $\mathcal{A}$.

Question: $\mathcal{A}$ va-t-il toujours atteindre une configuration temporellement périodique, à partir d'une configuration finie (ou périodique) quelconque ?

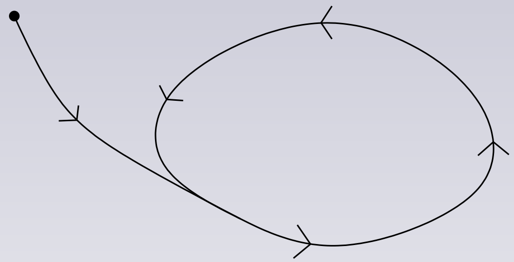


Périodicité ultime

Problème ULT

Instance: Un automate de sable $\mathcal{A}$.

Question: $\mathcal{A}$ va-t-il toujours atteindre une configuration temporellement périodique, à partir d'une configuration finie (ou périodique) quelconque ?



Théorème 6

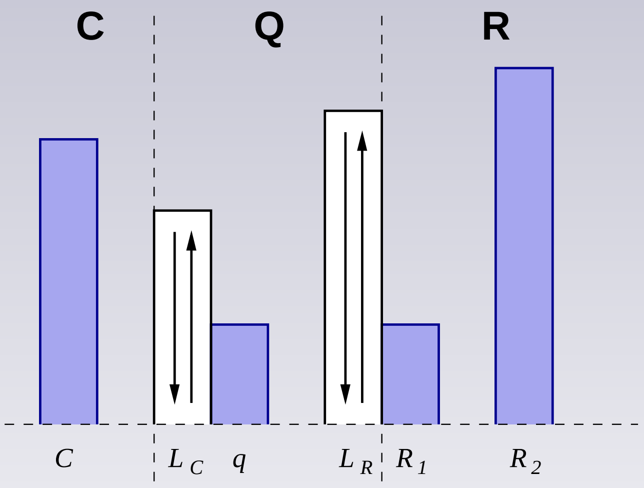
ULT est indécidable.

Preuve

- Réduction à partir d'un machine à deux compteurs, initialisée à 0.

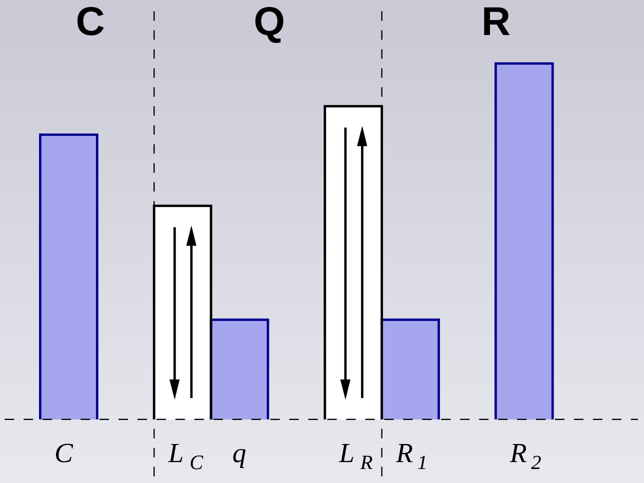
  ↪ $\mathcal{M} = \langle Q, q_0, q_f, \delta \rangle$.

- Principe de la simulation :



Preuve

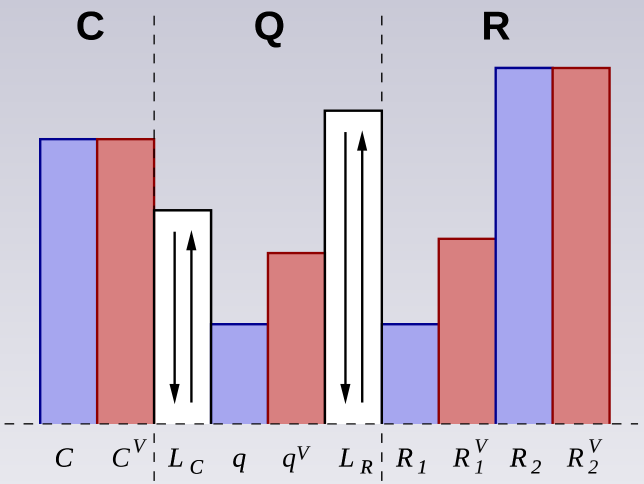
- Réduction à partir d'un machine à deux compteurs, initialisée à 0.
  - ↪ $\mathcal{M} = \langle Q, q_0, q_f, \delta \rangle$.
- Principe de la simulation :



1. Simulation;

Preuve

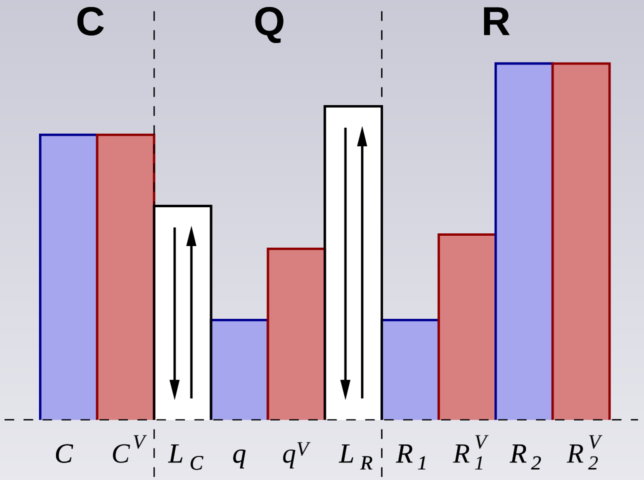
- Réduction à partir d'un machine à deux compteurs, initialisée à 0.
  - $\mathcal{M} = \langle Q, q_0, q_f, \delta \rangle$.
- Principe de la simulation :



1. Simulation;
2. Vérification;

Preuve

- Réduction à partir d'un machine à deux compteurs, initialisée à 0.
  - $\mathcal{M} = \langle Q, q_0, q_f, \delta \rangle$.
- Principe de la simulation :



1. Simulation;

2. Vérification;

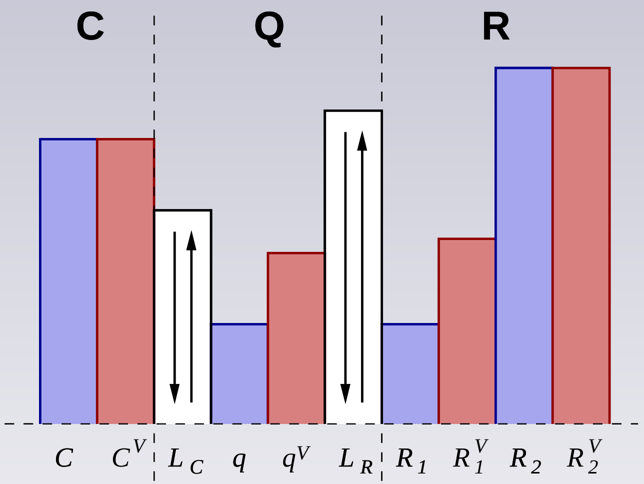
3. Comparaison.

Preuve

➢ Réduction à partir d'un machine à deux compteurs, initialisée à 0.

↳ $\mathcal{M} = \langle Q, q_0, q_f, \delta \rangle$.

➢ Principe de la simulation :



1. Simulation;
2. Vérification;
3. Comparaison.

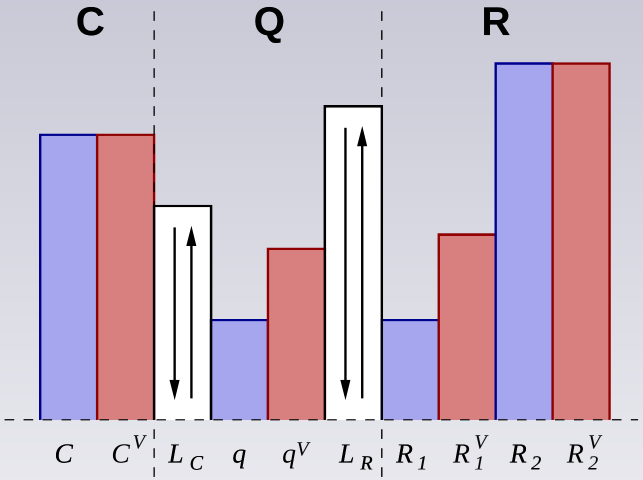
Les configurations malformées arrêtent d'évoluer.

Preuve

➢ Réduction à partir d'un machine à deux compteurs, initialisée à 0.

↪ $\mathcal{M} = \langle Q, q_0, q_f, \delta \rangle$.

➢ Principe de la simulation :



1. Simulation;
2. Vérification;
3. Comparaison.

Les configurations malformées arrêtent d'évoluer.

➠ $\mathcal{M}$ termine $\iff$ $\mathcal{A}$ est ultimement périodique.

Conclusion

- Modèle simple dans sa description, mais comportement très complexe.

- Beaucoup de travail reste à faire !

- Autre voie : reconnaissance de langages sur alphabet infini.
  - Définir un langage sur $\mathbb{Z}$.
  - Choisir un mode de reconnaissance.
  - Classer les automates selon les langages reconnus.
  - *etc.*