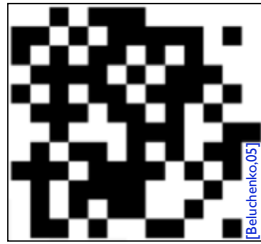


Automates Cellulaires



Automate Cellulaires 1D et 2D

Nicolas Bredèche

Professeur des Universités (ISIR, UPMC)
nicolas.bredeche@upmc.fr

Module: 21013

Dernière mise à jour: 2019-01-21

--

- Contenu
 - Automates Cellulaires à une dimension
 - Automates Cellulaires à deux dimensions
 - Un AC2D particulier: Jeu de la Vie
 - AC2D: autres modèles
- Objectif du cours
 - Notion d'auto-réplication
 - Systèmes dynamiques discrets
 - Modélisation de systèmes complexes

Exemple introductif

Exemple: modéliser les déplacements d'une file de voitures

si case libre : avance avec $p=1$ ← *irréaliste!*
sinon : attend



Dérouler la règle sur 3 itérations
On met à jour toutes les cases en même temps

Exemple: modéliser les déplacements d'une file de voitures

si case libre : avance avec $p=1$ ← *irréaliste!*
sinon : attend



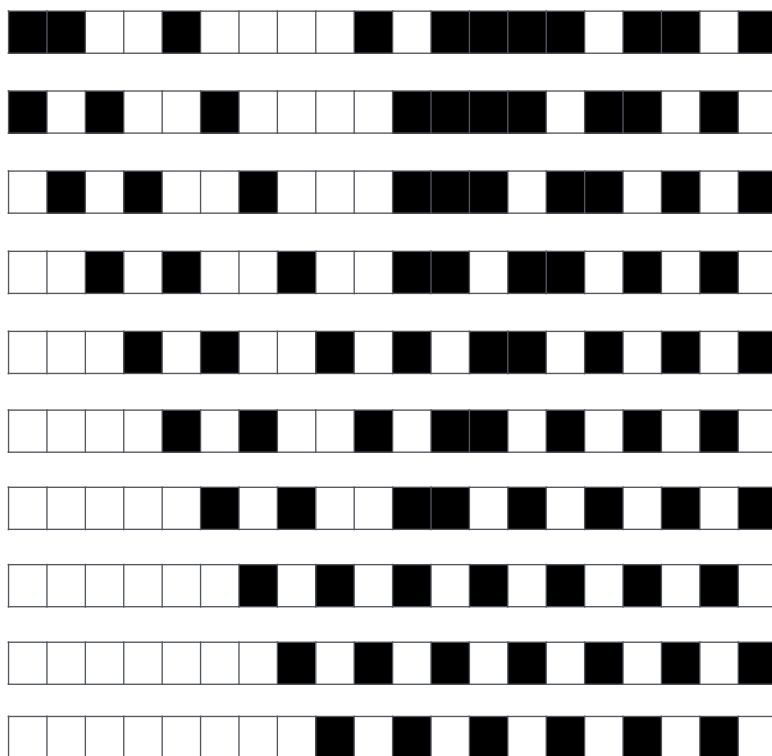
Exemple: modéliser les déplacements d'une file de voitures

si case libre : avance avec $p=1$ ← *irréaliste!*
sinon : attend

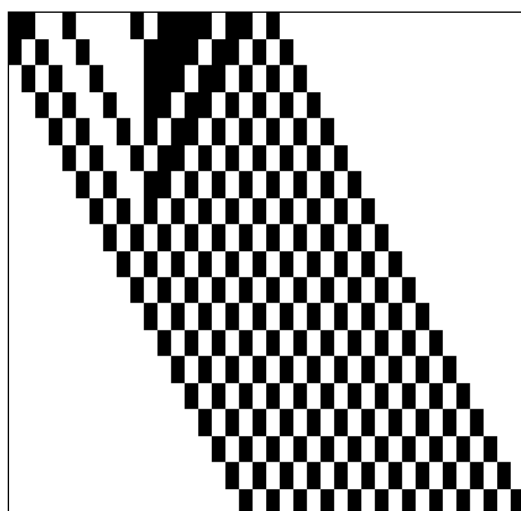


Exemple: modéliser les déplacements d'une file de voitures

si case libre : avance avec $p=1$ ← *irréaliste!*
sinon : attend



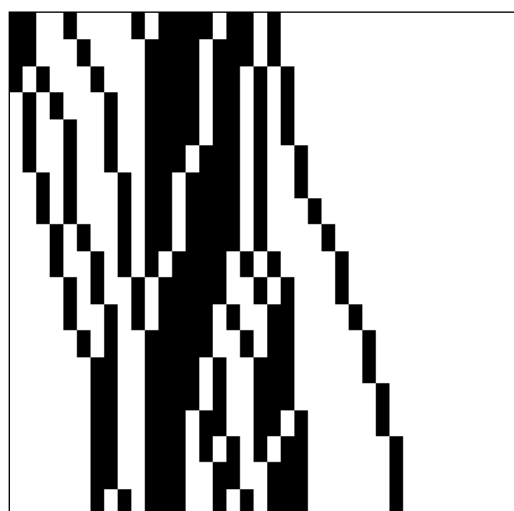
si case libre : avance avec $p=1$
sinon : attend



convergence vers un état stable

hypothèse idéale, mais irréaliste

si case libre : avance avec $p=0.5$
sinon : attend



déplacements non homogènes => embouteillages!

hypothèse réaliste (mais pas idéale...)

Définitions

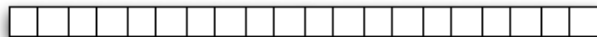
- Éléments statiques (l'environnement)
 - topologie
 - nombre d'états possibles
 - état initial
- Éléments dynamiques (mise à jour)
 - règles
 - voisinage
 - mode de mise à jour
- La mise à jour est faite sur l'ensemble de l'environnement

Applications

- **Modélisation et simulation de systèmes complexes**
 - Modélisation discrète de systèmes dynamiques
 - ▶ biologie: auto-réplication, morphogenèse
 - ▶ écosystèmes: colonies d'individus non mobiles, croissance de structures
 - ▶ physique: gaz, fluides, mécanique, etc.
 - ▶ ingénierie: conception d'architecture, modélisation circulation, etc.
- **Etude de la décidabilité, calculabilité, complexité**
 - décidabilité: existe t'il une solution?
 - calculabilité: si oui, existe t'il un algorithme pour l'obtenir?
 - complexité: si oui, quel est le coût de cet algorithme?

Automates Cellulaires 1D

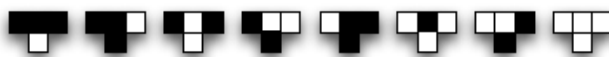
Environnement



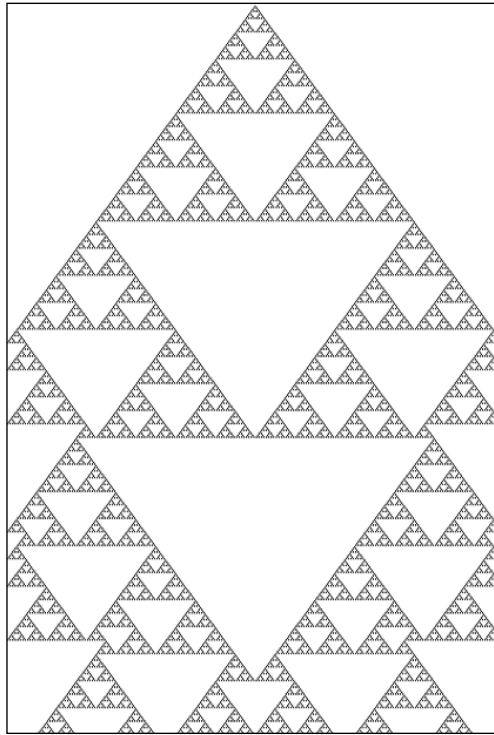
Voisinage d'une cellule



Exemples d'un ensemble de règles



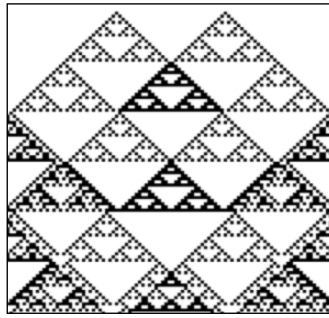
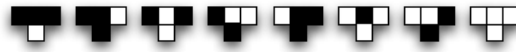
Automate cellulaire 1D à deux états
(entre autres: Wolfram, 1983-...)



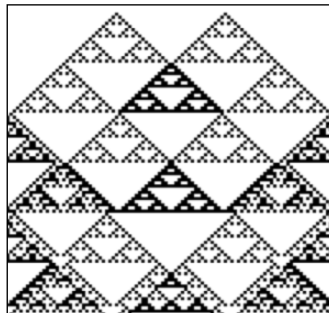
Rule 90 - automate cellulaire 1D 2-couleurs
[Wolfram, 1983]



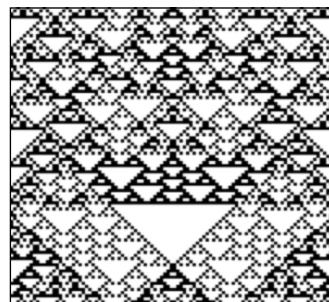
Conus Aulicus



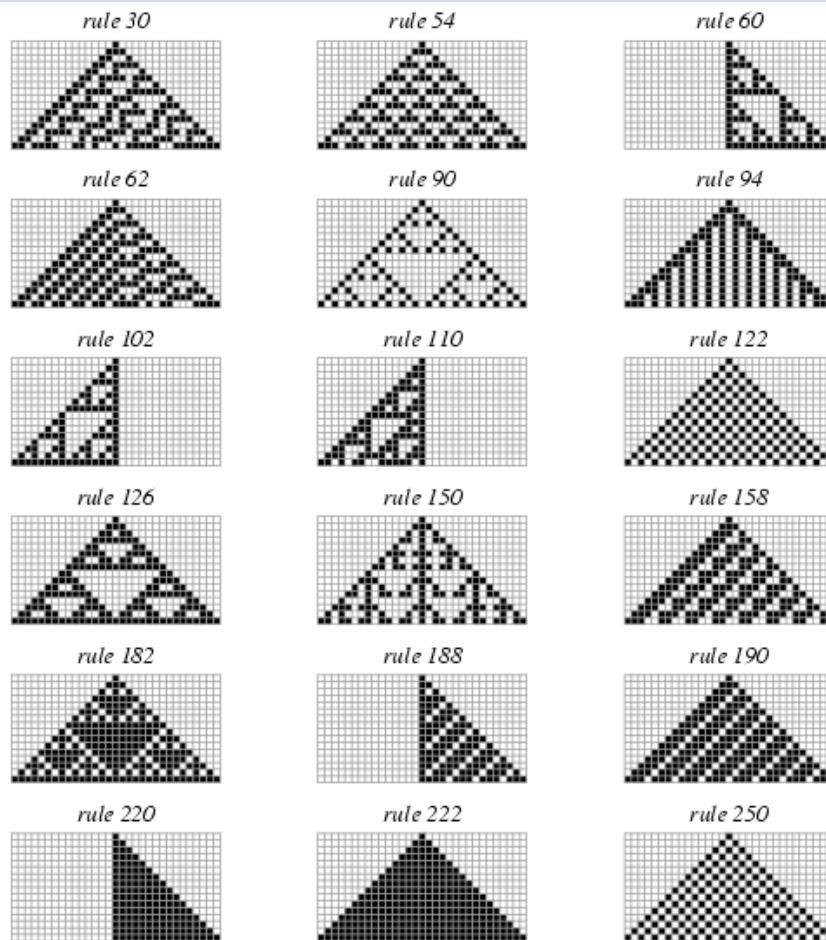
Influence de l'initialisation



...plus tard...



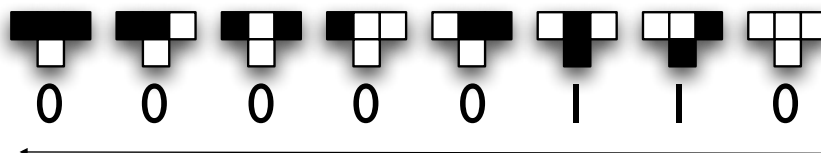
Complexité des dynamiques



Quelques exemples [Wolfram, 1983-2002]

source: <http://mathworld.wolfram.com>

Numérotation des règles



Le numéro d'une règle est déterminé en lisant, de droite à gauche, l'état résultant pour chaque configuration des cellules précédentes. Pour l'ensemble de sous-règles utilisé ci-dessus, on lit de droite à gauche: 00000110 (en binaire, on écrit 00000110b).

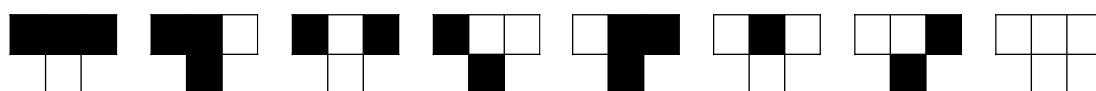
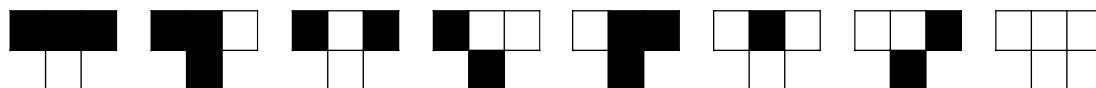
Pour convertir ce numéro depuis la représentation binaire vers une représentation en décimal, on écrit:

$$\begin{aligned}
 &00000110b \\
 &= 0 * 2^7 + 0 * 2^6 + 0 * 2^5 + 0 * 2^4 + 0 * 2^3 + 1 * 2^2 + 1 * 2^1 + 0 * 2^0 \\
 &= 0 + 0 + 0 + 0 + 0 + 2^2 + 2^1 + 0 \\
 &= 6d
 \end{aligned}$$

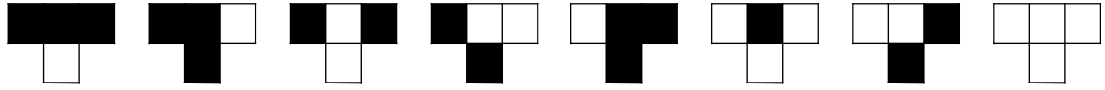
Il s'agit donc de la **règle 6**.

Note: Il n'existe que 256 règles (numérotées de 0 à 255) dans le cadre des automates cellulaires 1D à deux états.

Comment identifier le numéro de la règle ?

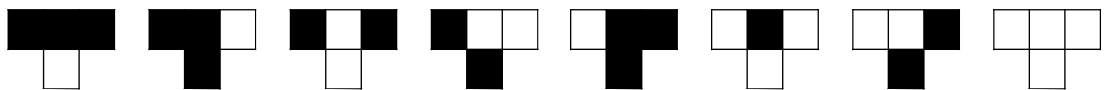


$$0 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$



$$0 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

$$0 \times 128 + 1 \times 64 + 0 \times 32 + 1 \times 16 + 1 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1$$

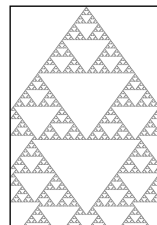


$$0 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

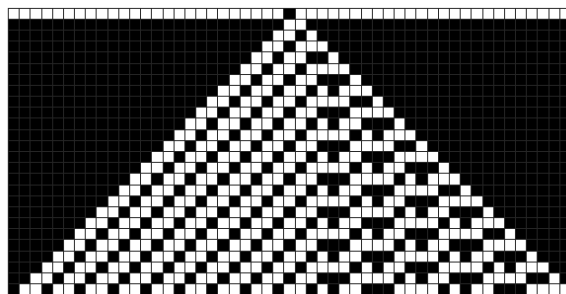
$$0 \times 128 + 1 \times 64 + 0 \times 32 + 1 \times 16 + 1 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1$$

$$0 + 64 + 0 + 16 + 8 + 0 + 2 + 0 = 90$$

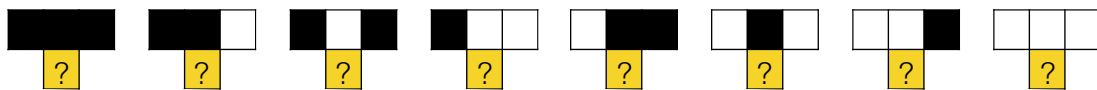
Règle 90



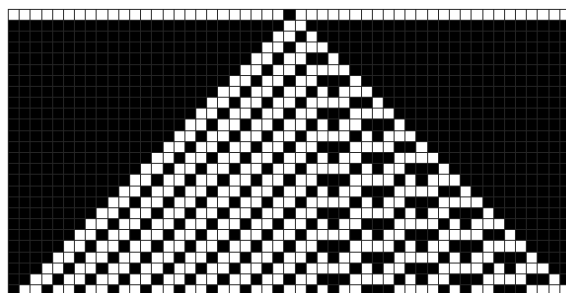
Comment construire la règle n.131?



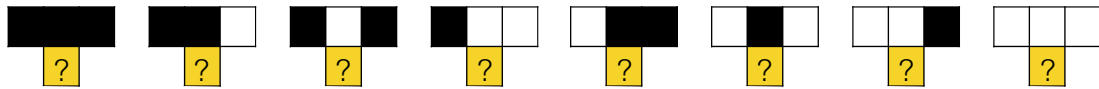
Comment construire la règle n.131?



$$? \times 2^7 + ? \times 2^6 + ? \times 2^5 + ? \times 2^4 + ? \times 2^3 + ? \times 2^2 + ? \times 2^1 + ? \times 2^0 = 131$$

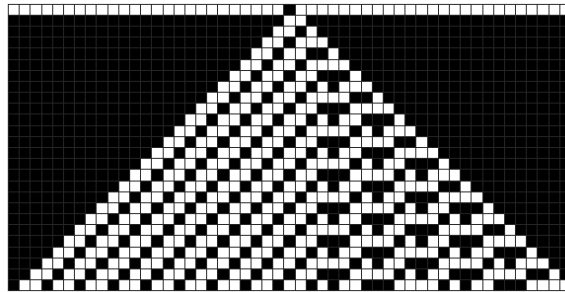


Comment construire la règle n.131?

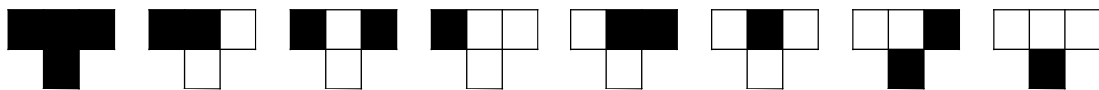


$$? \times 2^7 + ? \times 2^6 + ? \times 2^5 + ? \times 2^4 + ? \times 2^3 + ? \times 2^2 + ? \times 2^1 + ? \times 2^0 = 131$$

$$? \times 128 + ? \times 64 + ? \times 32 + ? \times 16 + ? \times 8 + ? \times 4 + ? \times 2 + ? \times 1 = 131$$

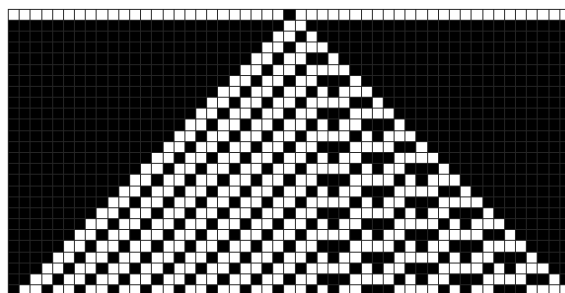


Comment construire la règle n.131?



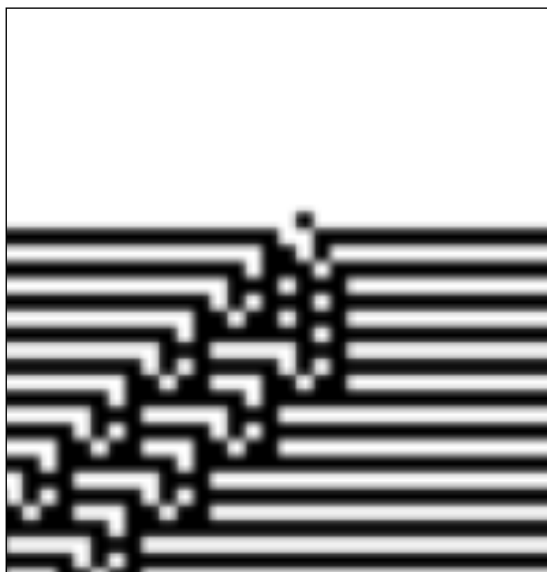
$$1 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 131$$

$$1 \times 128 + 0 \times 64 + 0 \times 32 + 0 \times 16 + 0 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1 = 131$$

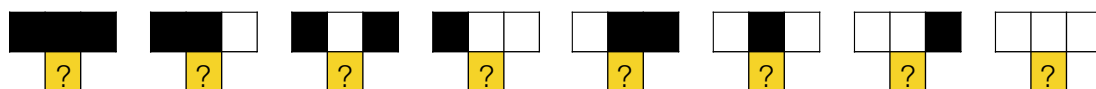
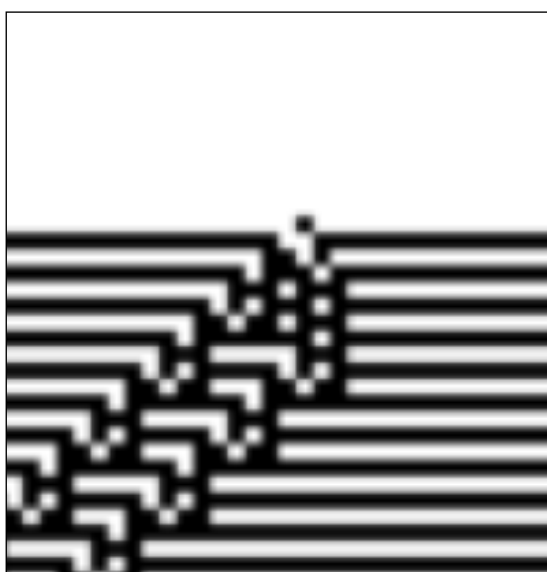


Exercice: identifiez la (les) règle(s) candidate(s)

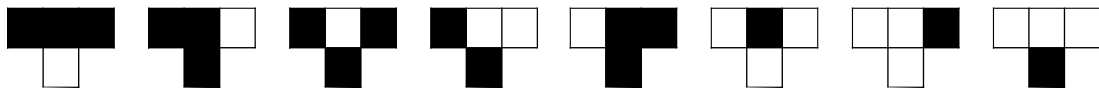
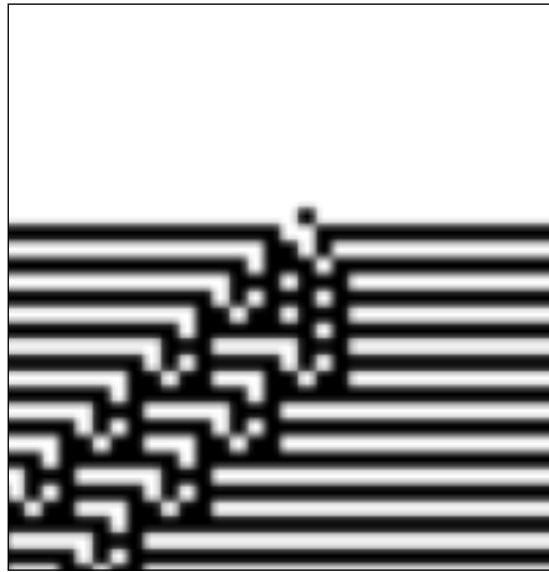
initialisation →



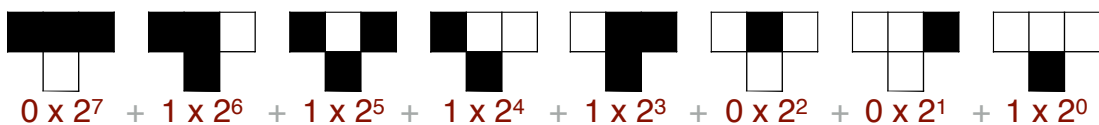
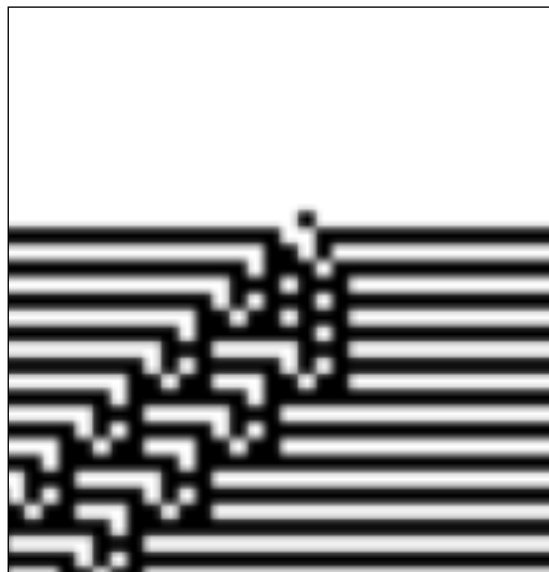
initialisation →



initialisation →

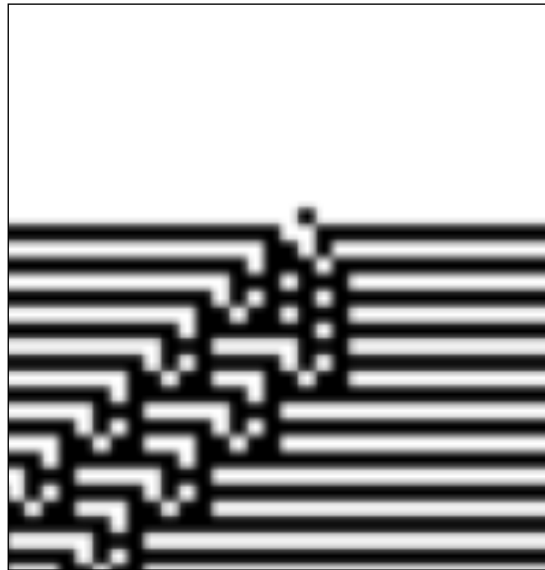


initialisation →

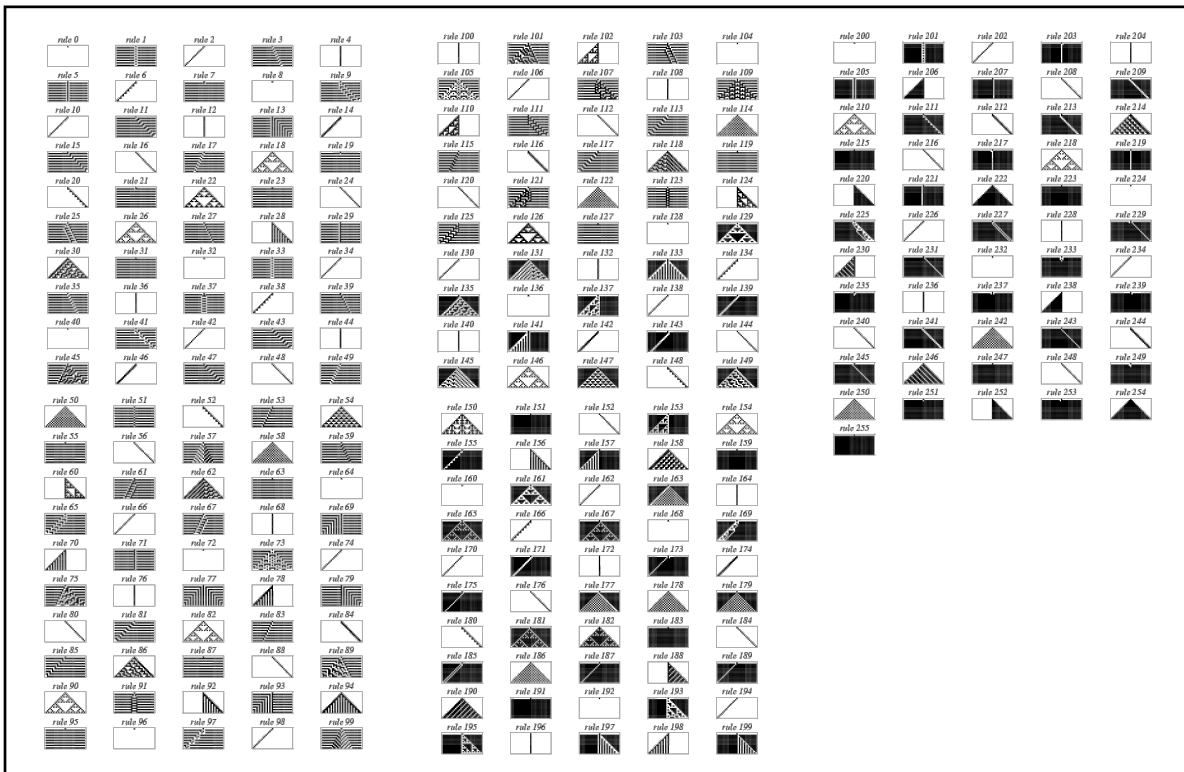


Il s'agit de la règle 121

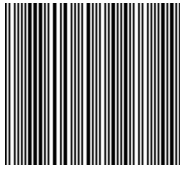
initialisation →



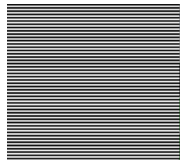
$$\begin{array}{cccccccccccc}
 \begin{array}{|c|} \hline \blacksquare \\ \hline \end{array} & + & \begin{array}{|c|c|} \hline \blacksquare & \blacksquare \\ \hline \end{array} & + & \begin{array}{|c|c|} \hline \blacksquare & \square \\ \hline \end{array} & + & \begin{array}{|c|c|} \hline \blacksquare & \square \\ \hline \end{array} & + & \begin{array}{|c|c|} \hline \square & \blacksquare \\ \hline \end{array} & + & \begin{array}{|c|c|} \hline \square & \blacksquare \\ \hline \end{array} & + & \begin{array}{|c|c|} \hline \square & \square \\ \hline \end{array} & + & \begin{array}{|c|c|} \hline \square & \square \\ \hline \end{array} & + & \begin{array}{|c|c|} \hline \square & \blacksquare \\ \hline \end{array} & + & \begin{array}{|c|c|} \hline \square & \square \\ \hline \end{array} & = & 121
 \end{array}$$



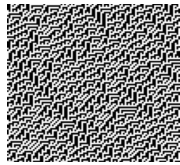
Les 256 règles pour les CA ID 2-couleurs
[Wolfram, 1983-2002]



CLASSE 1 : motif homogène



CLASSE 2 : motifs simples ou périodiques



CLASSE 3 : motif chaotique

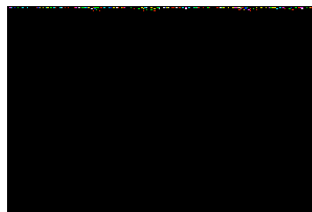


CLASSE 4 : motif répété

Classes de complexité de Wolfram
[Wolfram, 1983]

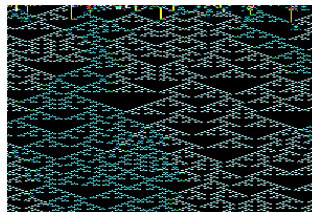
lambda = 1 - nb_transitions_mortelles

$\lambda = 0$



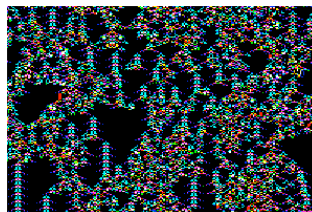
structures homogènes

$\lambda < 0.3$



structures cycliques
(avec auto-similarité)

$0.3 < \lambda < 0.5$



structures complexes

$\lambda > 0.5$

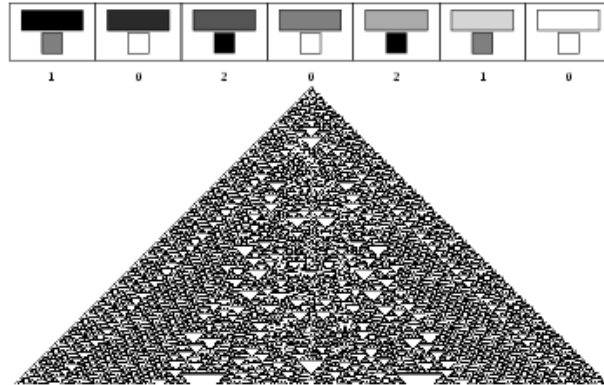


structures chaotiques

$\lambda = 1$

Classes de complexité de Langton
(étude dans le cas 1-D 8-couleurs avec voisinage de 5) [Langton, 1991]

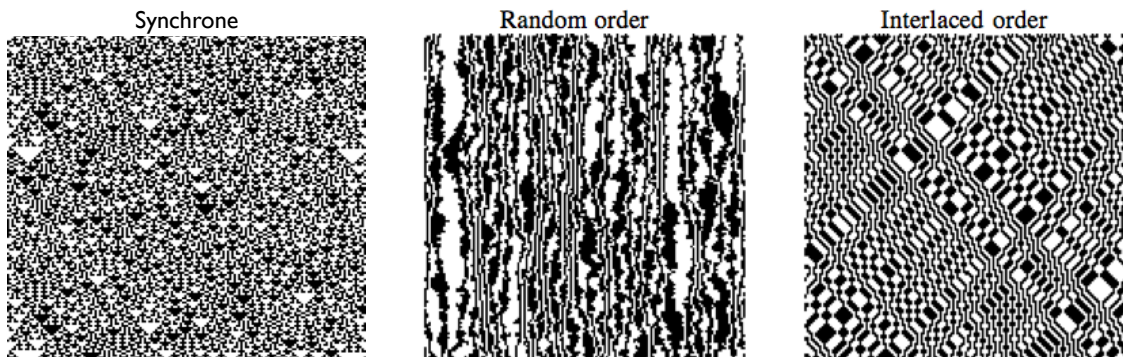
Nombre d'états



code 912 - automate cellulaire totalistique 3-couleurs

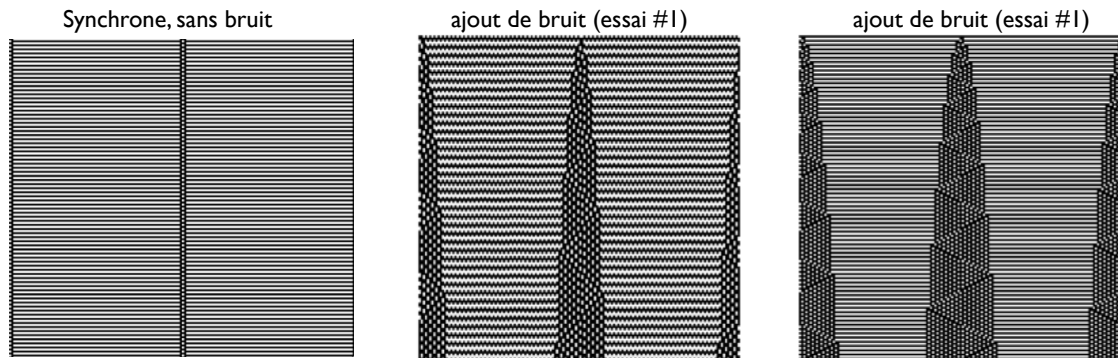
source: <http://mathworld.wolfram.com>

Mise à jour: effet de l'ordre



règle n.150

Mise à jour: effet du bruit



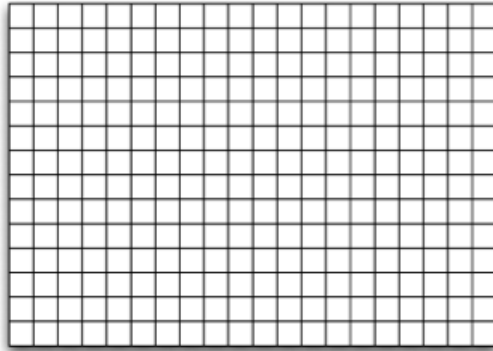
règle n.123

Effects of fluctuation and randomness. Kanada, 1997

Automates Cellulaires 2D

AC-2D

Environnement (2D)



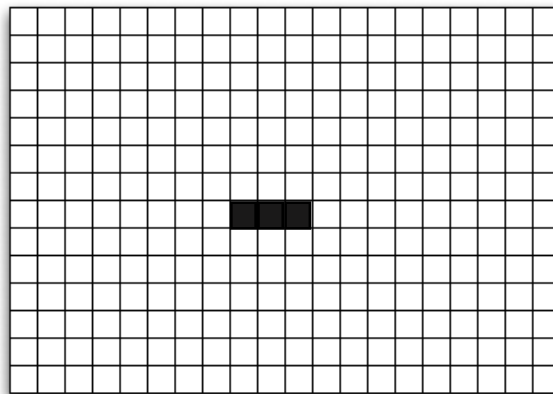
- Type de voisinage
 - Von neumann, Moore, Margolus, ...
- Prise en compte du voisinage
 - Localisé, Totalistique, ...
- Et toujours...
 - Nombre d'états, bruit, etc.



voisinage de Moore (8 voisins)

voisinage de Von Neumann (4 voisins)

Environnement (2D)



Voisinage d'une cellule

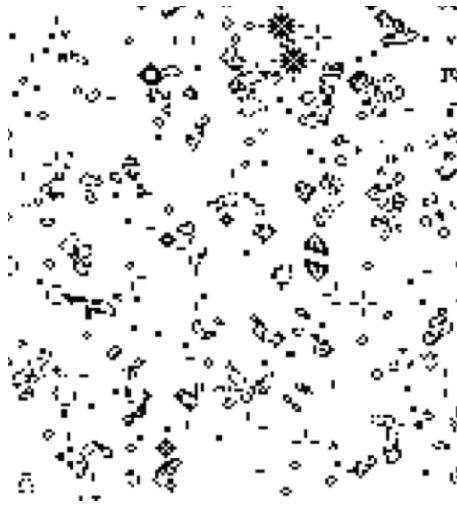


voisinage de Moore (8 voisins)

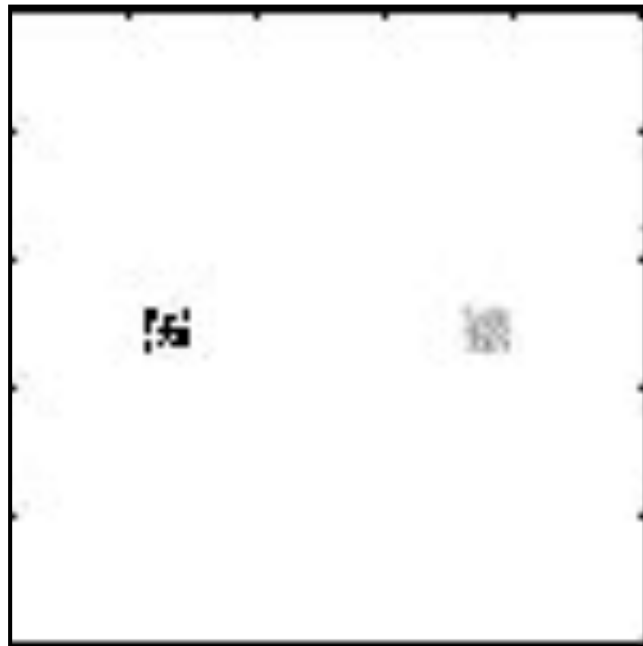
Trois Règles (automate dit "totalistic CA")

1. Si et soit 2 ou soit 3 voisins vivants \longrightarrow
2. Si et **exactement** 3 voisins vivants \longrightarrow
3. Sinon: \longrightarrow

Jeu de la Vie [Conway, 1970]

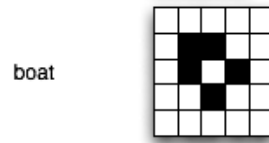
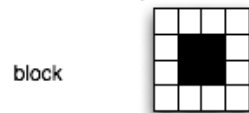


Une instance du jeu de la vie

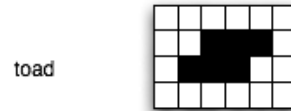


(remarque : vitesse rapide, toutes les itérations n'apparaissent pas)

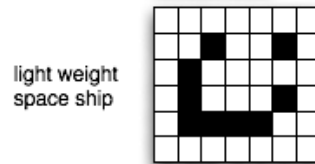
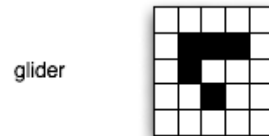
Structures statiques








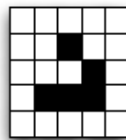
Structures dynamiques: oscillateurs à 2 états



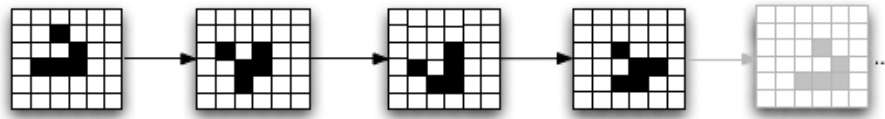
Structures dynamiques: oscillateurs avec déplacement



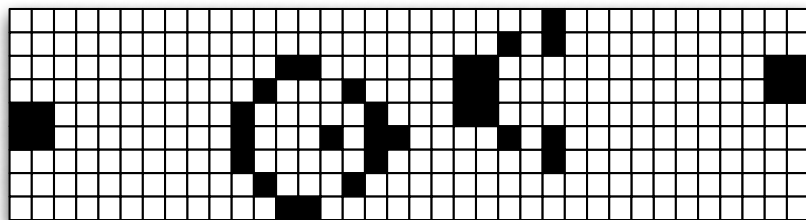
1. Si  et *soit 2 ou soit 3 voisins vivants* → 
2. Si  et *exactement 3 voisins vivants* → 
3. Sinon: → 



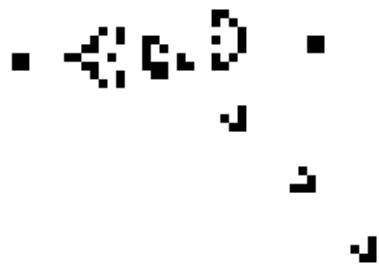
un planeur (“glider”)



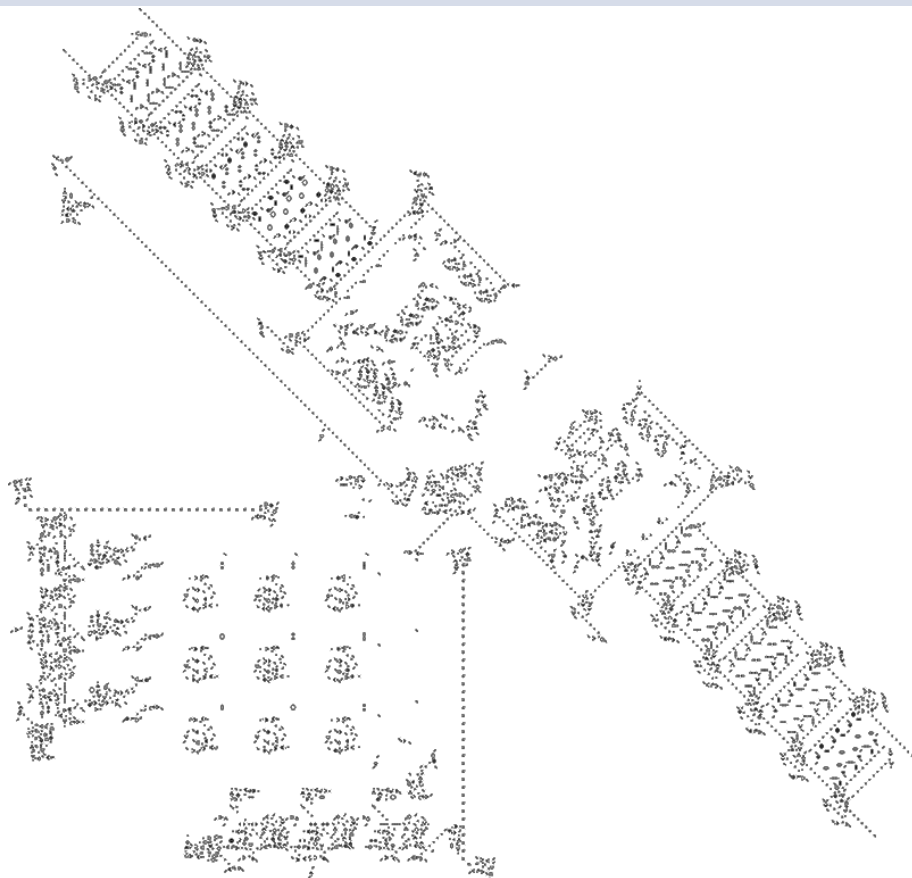
les 4 états d'un planeur



une structure à croissance continue: le canon ("glider gun")

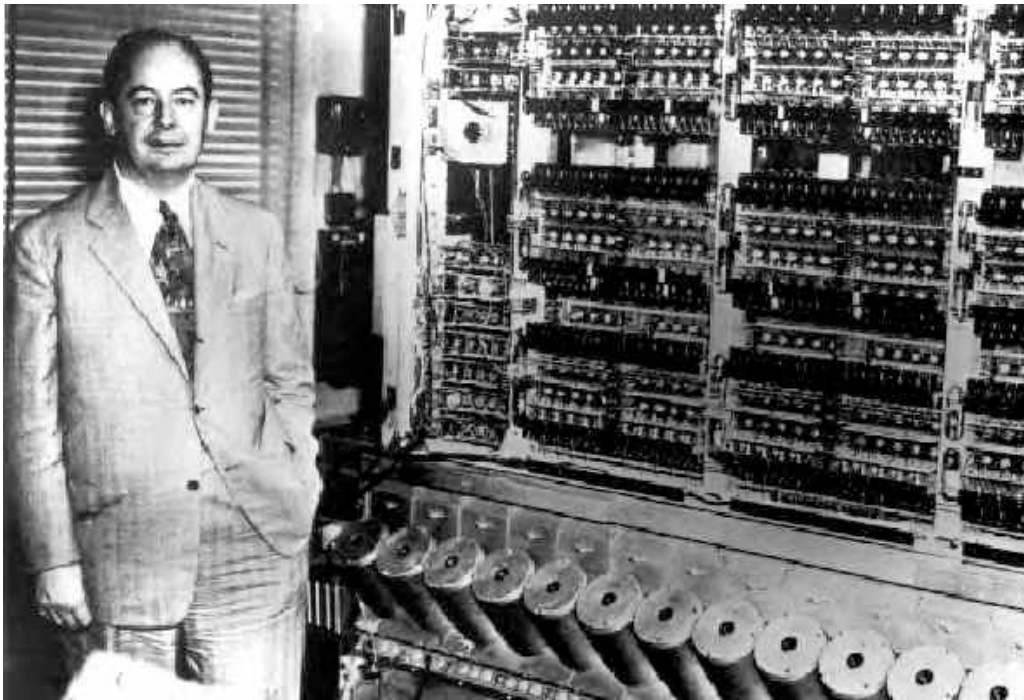


Canon et planeurs

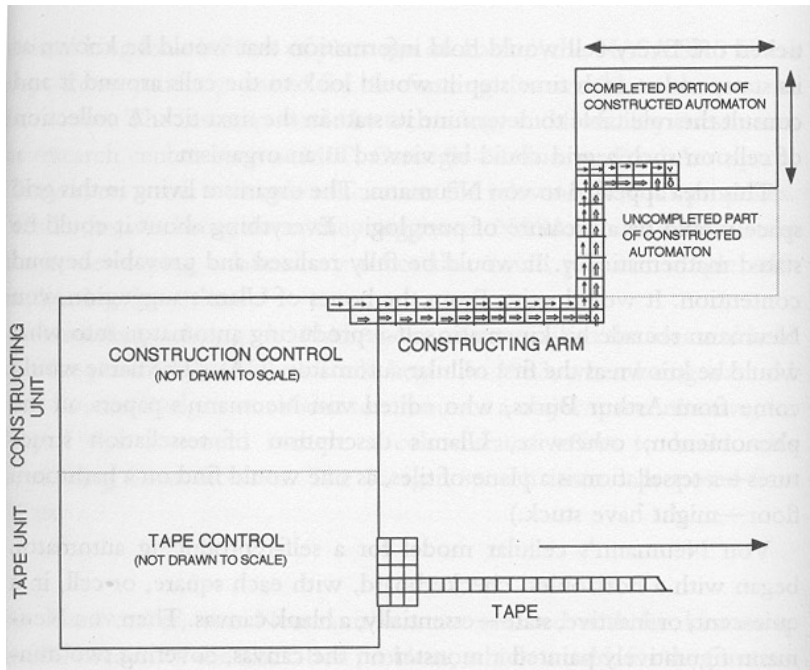


une machine de Turing dans le jeu de la vie
(design par P. Rendell, 2002)

Auto-réplication



John Von Neumann (à gauche)
(1903-1957)



Automate cellulaire constructeur universel
(Von Neumann, ~1940)

Environnement

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	2	2	2	2	3	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	2	5	4	4	4	4	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	2	2	2	2	3	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Voisinage d'une cellule

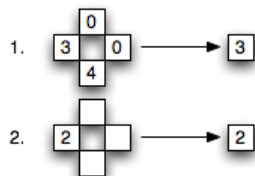


voisinage de Moore (8 voisins)



voisinage de Von Neumann (4 voisins)

Exemples de règle (avec voisinage de Von Neumann)



Automates Cellulaires (John Von Neumann et Stanislas Ulam, ~1950)

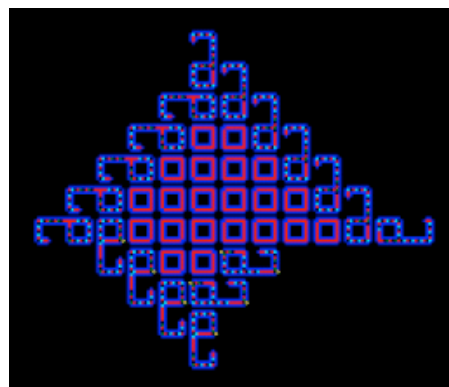
```

  2 2 2 2 2 2 2 2
  2 1 7 0 1 4 0 1 4 2
  2 0 2 2 2 2 2 2 0 2
  2 7 2           2 1 2
  2 1 2           2 1 2
  2 0 2           2 1 2
  2 7 2           2 1 2
  2 1 2 2 2 2 2 2 1 2 2 2 2 2
  2 0 7 1 0 7 1 0 7 1 1 1 1 1
  2 2 2 2 2 2 2 2 2 2 2 2 2

```

Structure auto-répliquante de Langton (1984)

CA 2D 8-couleurs, voisinage de Von Neumann, 29 règles, 86 cellules au départ



Après quelques itérations...
[Langton, 1984]

Automates Cellulaires

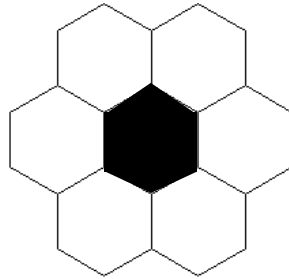
Modélisation avancée

Trois cas d'école

- Flocon de neige
- WireWorld
- Feu de forêt

Flocon de neige

“Snowflake growth”



AC totalistique sur grille hexagonale

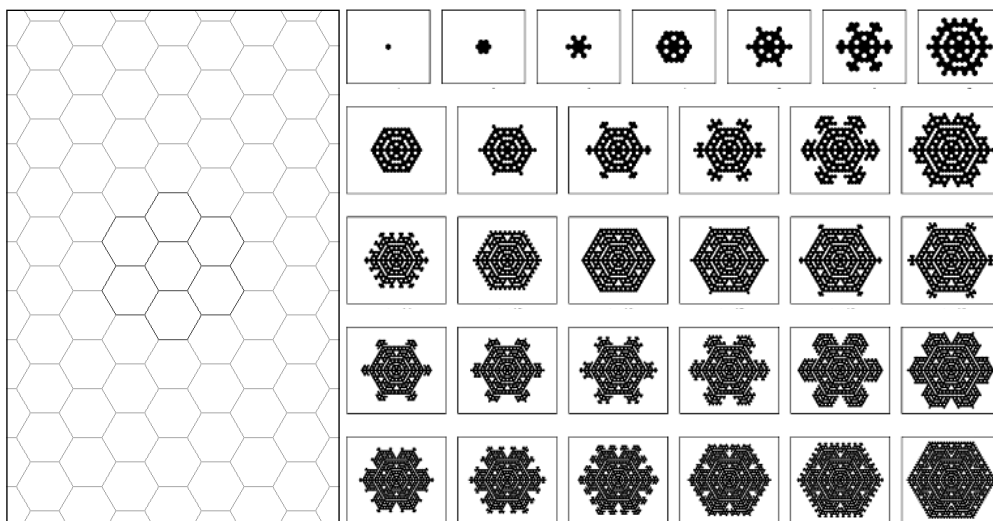
Règle: actif si un seul voisin actif

une fois activée, une case le reste pour toujours

cf. <http://www.wolframscience.com/nksonline/page-371>

Flocon de neige

“Snowflake growth”



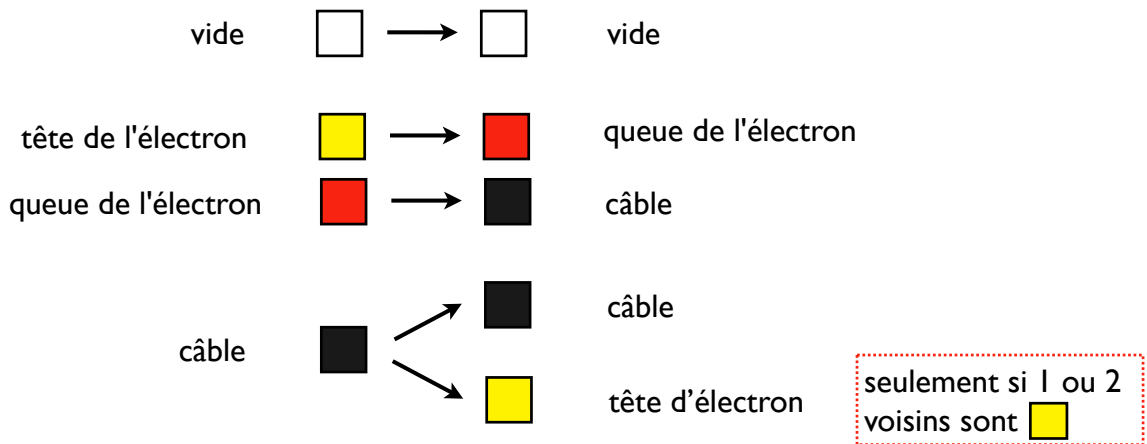
AC totalistique sur grille hexagonale

Règle: actif si un seul voisin actif

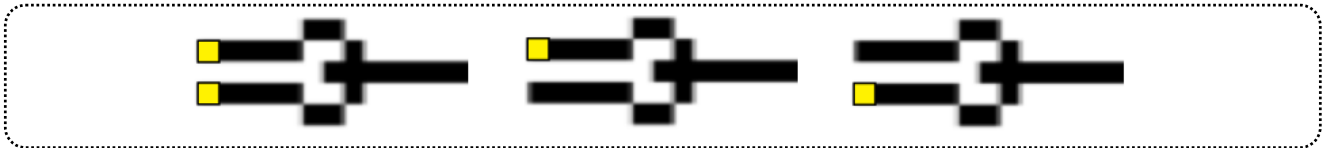
cf. <http://www.wolframscience.com/nksonline/page-371>

WireWorld

[Silverman, 1987]

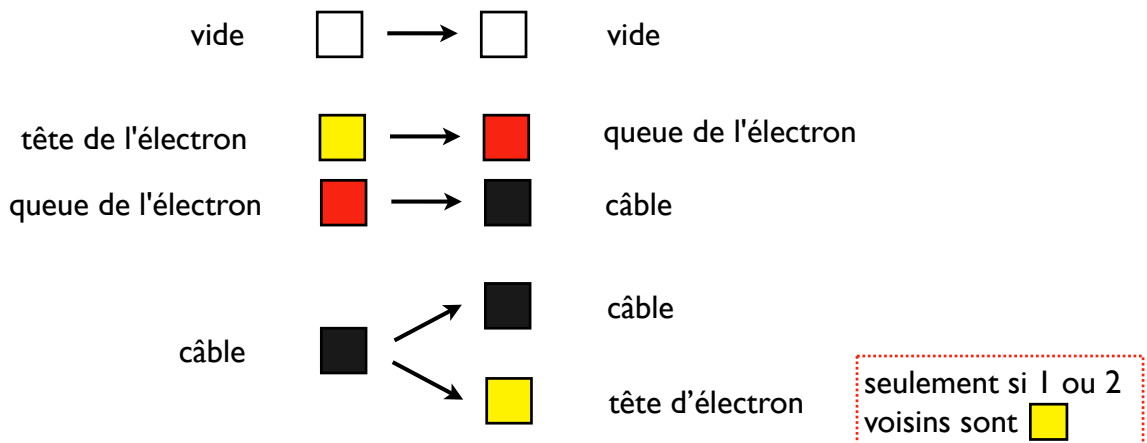


voisinage de Moore

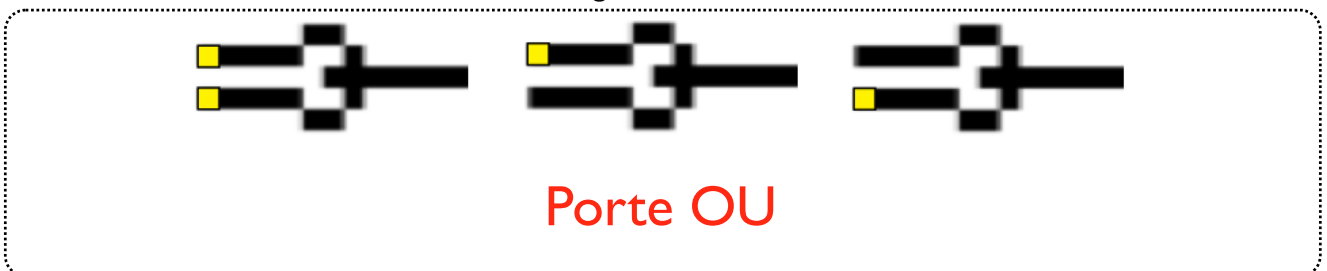


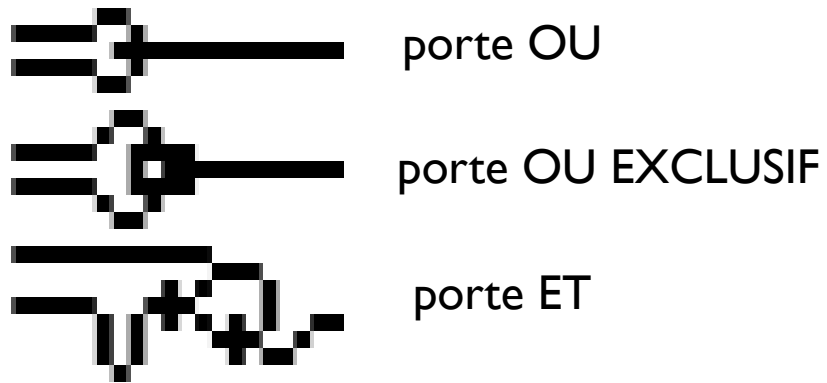
WireWorld

[Silverman, 1987]



voisinage de Moore

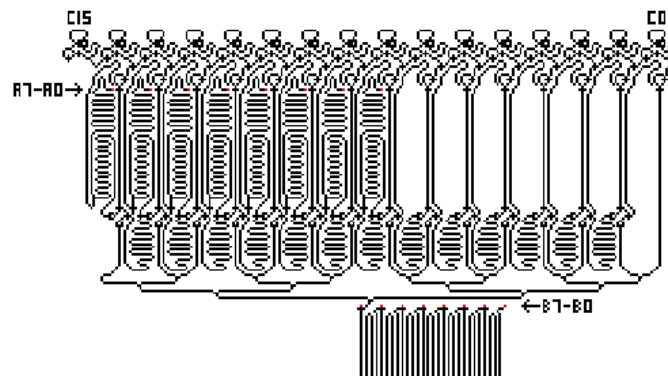




porte OU

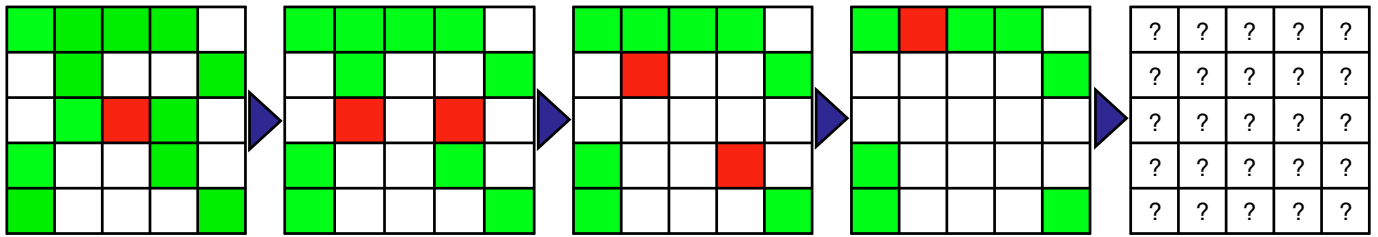
porte OU EXCLUSIF

porte ET



Multiplication de deux nombres 8 bits dans WireWorld
 [Gardner, 2002]

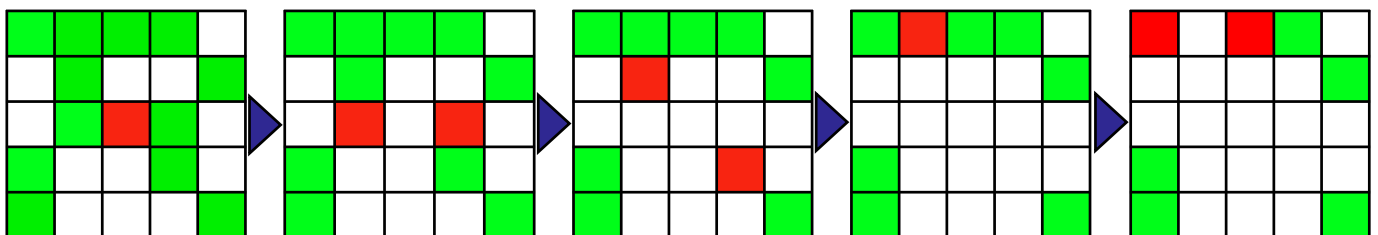
“Feu de Forêt”



Exercice:

Quelles propriétés ?
Quelles règles ?
Quel est l'état suivant?

“Feu de Forêt”



Propriétés:

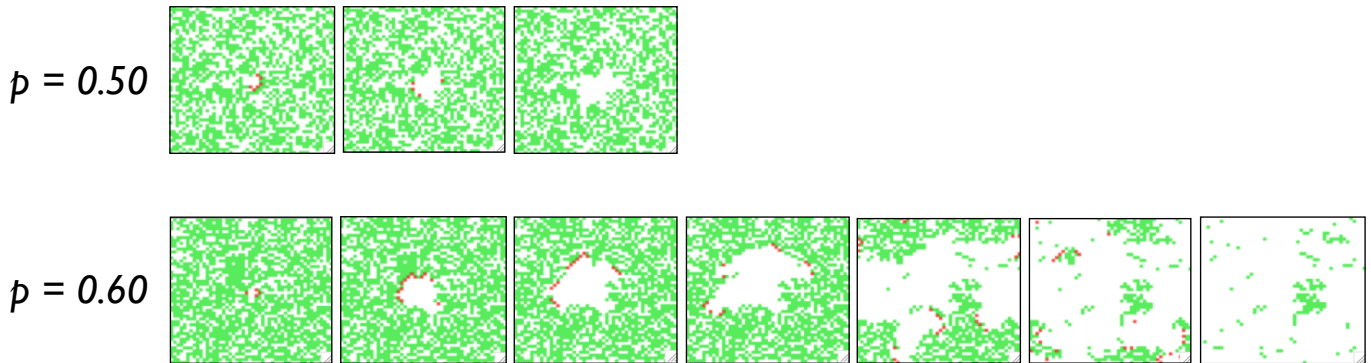
- ▶ 3 états (blanc,vert,rouge)
- ▶ voisinage Von Neumann
- ▶ totalistique

Règle:

- ▶ **si** vert et au moins un voisin est rouge **alors** devient rouge
- ▶ **si** rouge **alors** devient blanc

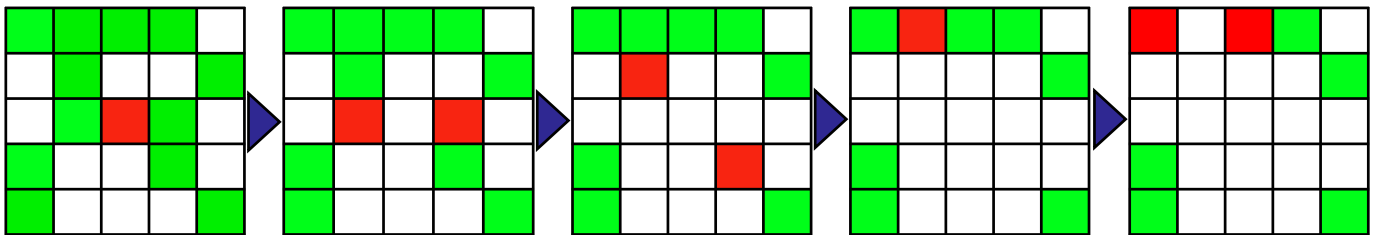
Seuil de percolation

Percolation: La percolation est un processus physique critique qui décrit pour un système, une transition d'un état vers un autre (wikipedia).



Seuil de percolation pour le feu de forêt:
densité d'arbres ≈ 0.55

Extensions



- Mise à jour

- Synchrones (par défaut) ;
- Asynchrone ordonnée (à éviter)
- Asynchrone randomisée

- Règles

- Proximité von Neumann (par défaut), Moore, entrelacée, etc.
- Proximité + probabiliste
- Proximité + propriétés internes/externes (ex.: type de bois, humidité, météo, etc.)

Synthèse

Automates Cellulaires : propriétés

- Définition de l'environnement et des règles
 - Topologie (2D, 3D, pavage de la grille, torique ou non)
 - Nombre d'états (2 couleurs, 5 couleurs...)
 - Voisinage (3,5,...)
- Mise à jour
 - Synchrones vs. asynchrones
 - ▶ synchrone: toutes les cellules sont mises à jour en même temps
 - ▶ asynchrone: mise à jour des cellules les unes après les autres (ordre déterminé ou non)
 - Déterministe vs. stochastique
 - ▶ Déterministe: donnera toujours le même résultat
 - ▶ Stochastique: deux simulations donneront des résultats différents (ex.: bruit, mutation)

Ces définitions s'étendent à tous les automates cellulaires

Synthèse

- Ce que l'on a vu
 - Automates Cellulaires 1D
 - Automates Cellulaires 2D : le jeu de la vie
 - Jeu de la Vie, WireWorld, Feu de forêt, Flocon de neige
- Ce qu'il faut retenir
 - Construire et savoir identifier un AC-1D
 - Savoir caractériser la classe de complexité d'un AC-1D
 - Savoir construire et identifier les règles d'un AC-2D
 - Savoir concevoir un modèle dynamique discret avec un AC-2D

Bibliographie

- Adami (1998) Artificial Life. Springer.
- Ganguly et. al (1993) A survey on cellular automata.
- Fatès (2001) Automates cellulaires: vers une nouvelle épistémologie?
- Wolfram (2002) A new kind of science.
- <http://mathworld.wolfram.com>
- <http://www.rennard.org/alife>
- <http://www.cs.bgu.ac.il/~sipper/selfrep/>
- nombreuses ressources sur le web