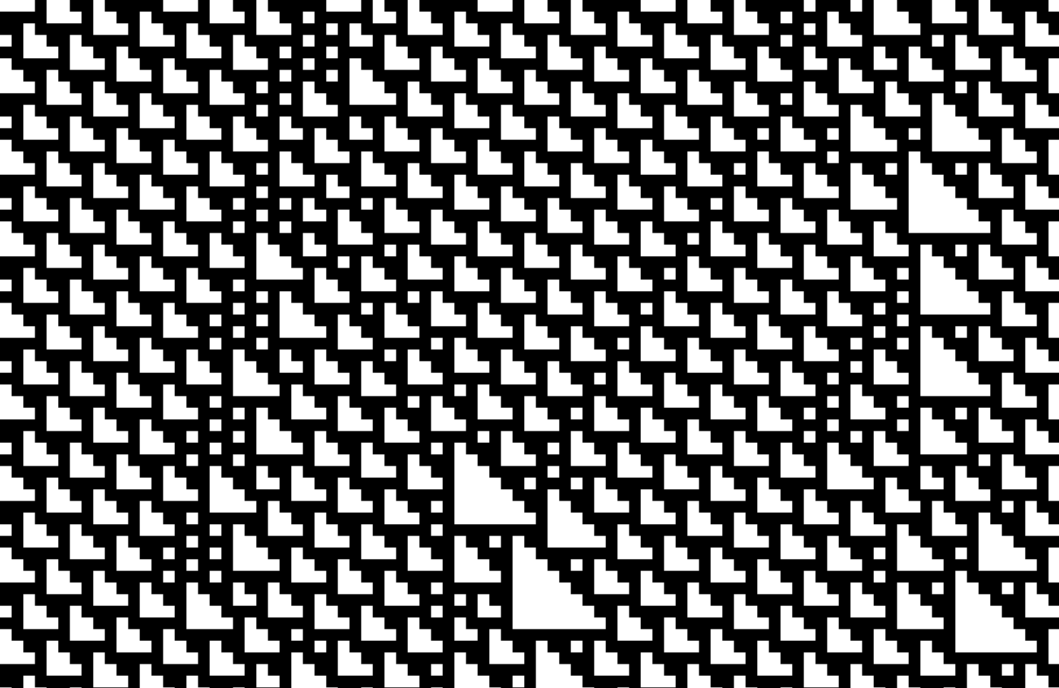


Universality in Elementary Cellular Automata

Matthew Cook

Department of Computation and Neural Systems,[*]
Caltech, Mail Stop 136-93,
Pasadena, California 91125, USA

The purpose of this paper is to prove a conjecture made by Stephen Wolfram in 1985, that an elementary one dimensional cellular automaton known as "Rule 110" is capable of universal computation. I developed this proof of his conjecture while assisting Stephen Wolfram on research for *A New Kind of Science* [1].

1. Overview

The purpose of this pa

Universality in Elementary Cellular Automata

Matthew Cook
*Department of Computation and Neural Systems,**
Caltech, Mail Stop 136-93,
Pasadena, California 91125, USA

Cook 2004 La **règle 110** est **universelle** pour le calcul.
~~**Wolfram 2002**~~
~~**Cook 1998**~~

1. Overview

The purpose of this pa

Universality in Elementary Cellular Automata

Matthew Cook
*Department of Computation and Neural Systems,**
Caltech, Mail Stop 136-93,
Pasadena, California 91125, USA

Cook 2004 La **règle 110** est **universelle** pour le calcul.
~~**Wolfram 2002**~~ En binaire, 110 s'écrit 01101110
~~**Cook 1998**~~

1. Overview

The purpose of this pa

Universality in Elementary Cellular Automata

Matthew Cook
*Department of Computation and Neural Systems,**
Caltech, Mail Stop 136-93,
Pasadena, California 91125, USA

Cook 2004 La **règle 110** est **universelle** pour le calcul.
~~**Wolfram 2002**~~ En binaire, 110 s'écrit 01101110
~~**Cook 1998**~~ 

1. Overview

The purpose of this pa

Machine de Turing universelle

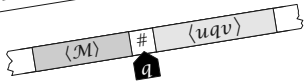
Théorème Il existe une machine de Turing **universelle** U qui **simule** toutes les machines de Turing.

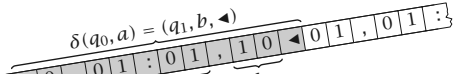
Pour toute machine de Turing $\mathcal{M}$ sur l'alphabet Σ et toute entrée $u \in \Sigma^*$, la machine U s'arrête sur l'entrée $\langle \mathcal{M}, u \rangle$ si et seulement si $\mathcal{M}$ s'arrête sur l'entrée u. Elle accepte si et seulement si $\mathcal{M}$ accepte et dans ce cas

$$f_U(\langle \mathcal{M}, u \rangle) = \langle f_{\mathcal{M}}(u) \rangle .$$

Remarque Né

Principes de construction



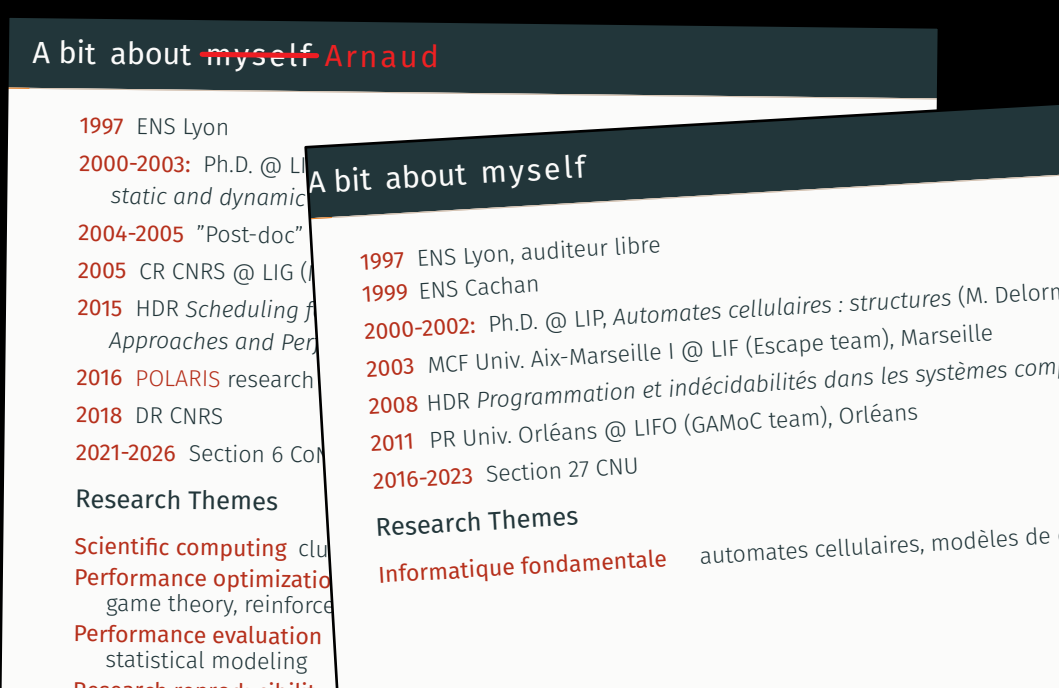


Automates cellulaires, dynamique et calculabilité

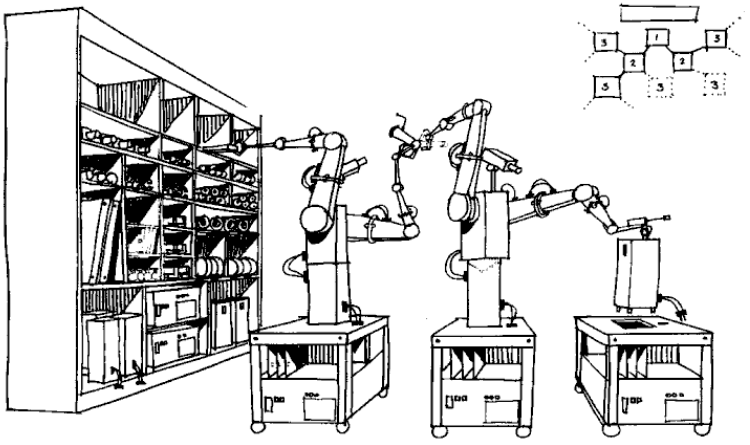
Nicolas Ollinger

LIFO, Université d'Orléans
IRIF, CNRS et Univ. Paris Cité (en délégation CNRS 2023/2024)

Semaine Ski, Le Pleynet — 26 janvier 2024


A bit about ~~myself~~ Arnaud
1997 ENS Lyon
2000-2003: Ph.D. @ LI
static and dynamic
2004-2005 "Post-doc"
2005 CR CNRS @ LIG (
2015 HDR Scheduling f
Approaches and Per
2016 POLARIS research
2018 DR CNRS
2021-2026 Section 6 Co
Research Themes
Scientific computing clu
Performance optimizatio
game theory, reinforce
Performance evaluation
statistical modeling
A bit about myself
1997 ENS Lyon, auditeur libre
1999 ENS Cachan
2000-2002: Ph.D. @ LIP, Automates cellulaires : structures (M. Delorm
2003 MCF Univ. Aix-Marseille I @ LIF (Escape team), Marseille
2008 HDR Programmation et indécidabilités dans les systèmes com
2011 PR Univ. Orléans @ LIFO (GAMoC team), Orléans
2016-2023 Section 27 CNU
Research Themes
Informatique fondamentale automates cellulaires, modèles de

Théorie des machines autoréplicatives

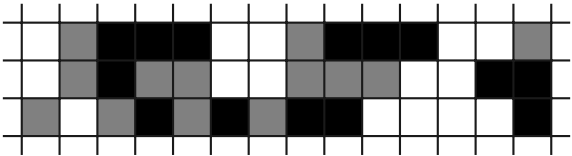


Les **automates cellulaires** émergent à la fin des années 40 des travaux d'Ulam et von Neumann.

Automates cellulaires

Un **automate cellulaire** est un système dynamique discret.

L'**espace** est discret et consiste en une grille régulière infinie de cellules. Chaque cellule est décrite par un **état** appartenant à un **ensemble fini d'états** commun.



Le **temps** est discret. À chaque top d'horloge, les cellules changent d'état de manière **déterministe**, **synchrone** et **uniforme**, selon une **règle locale de transition** commune.

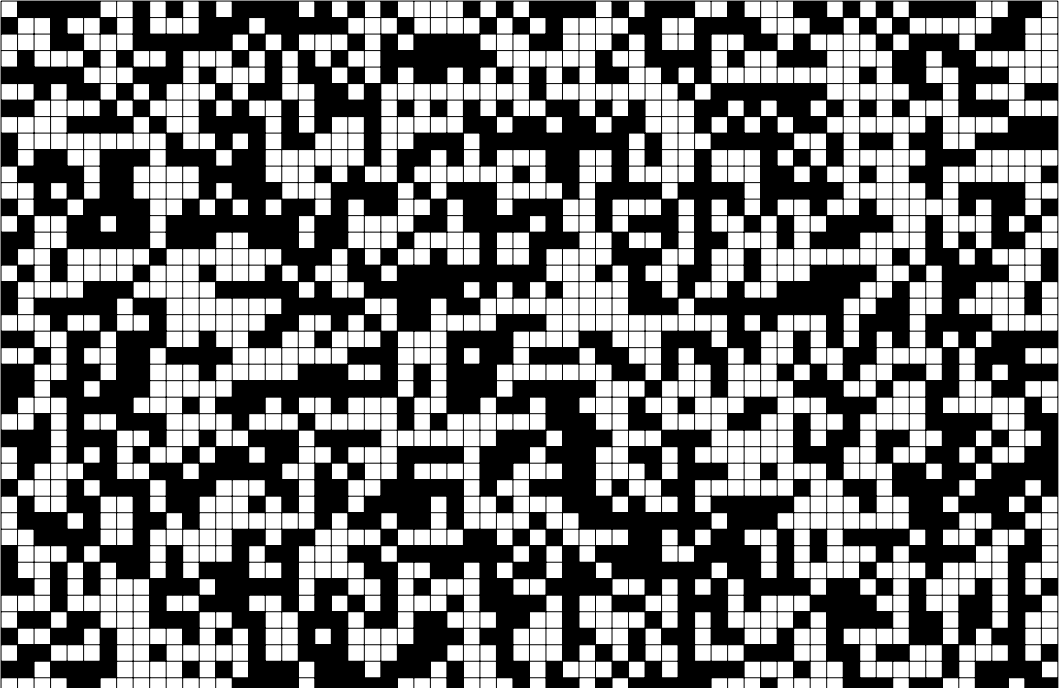
Le jeu de la vie de Conway

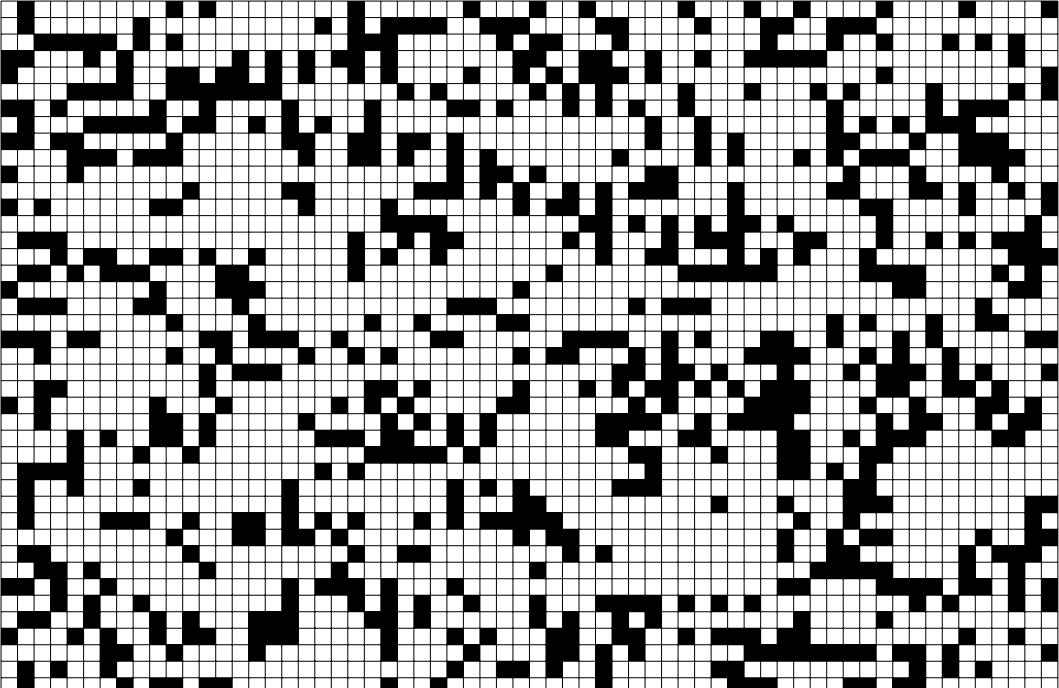
Le **jeu de la vie** est un AC inventé par Conway en 1970.

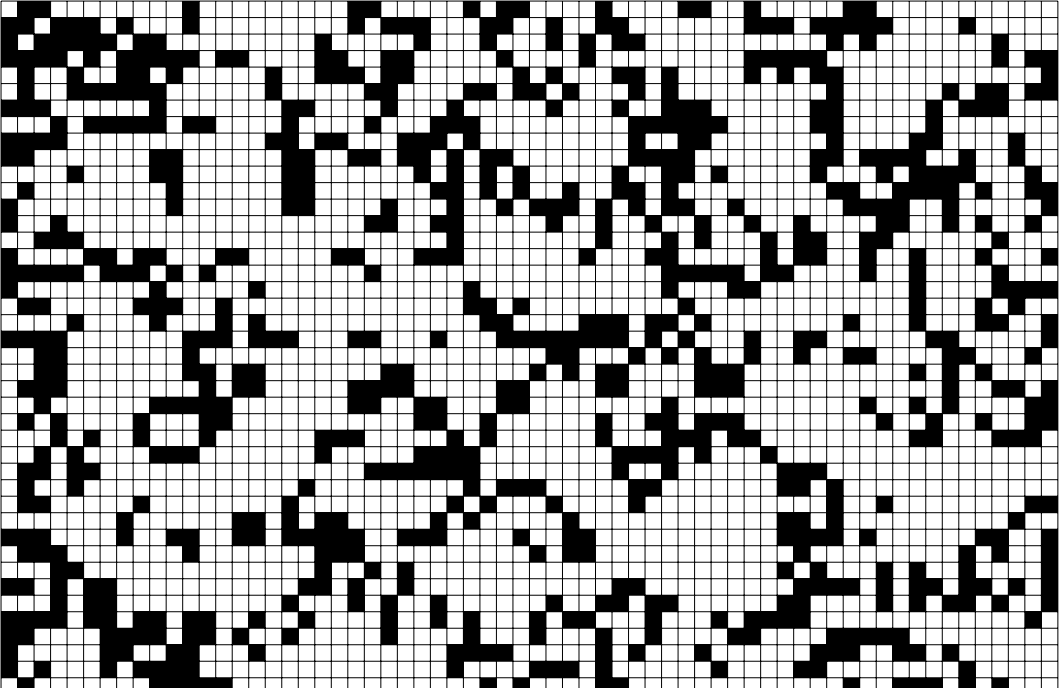
L'**espace** est une grille infinie de cellules **mortes** ou **vivantes**.

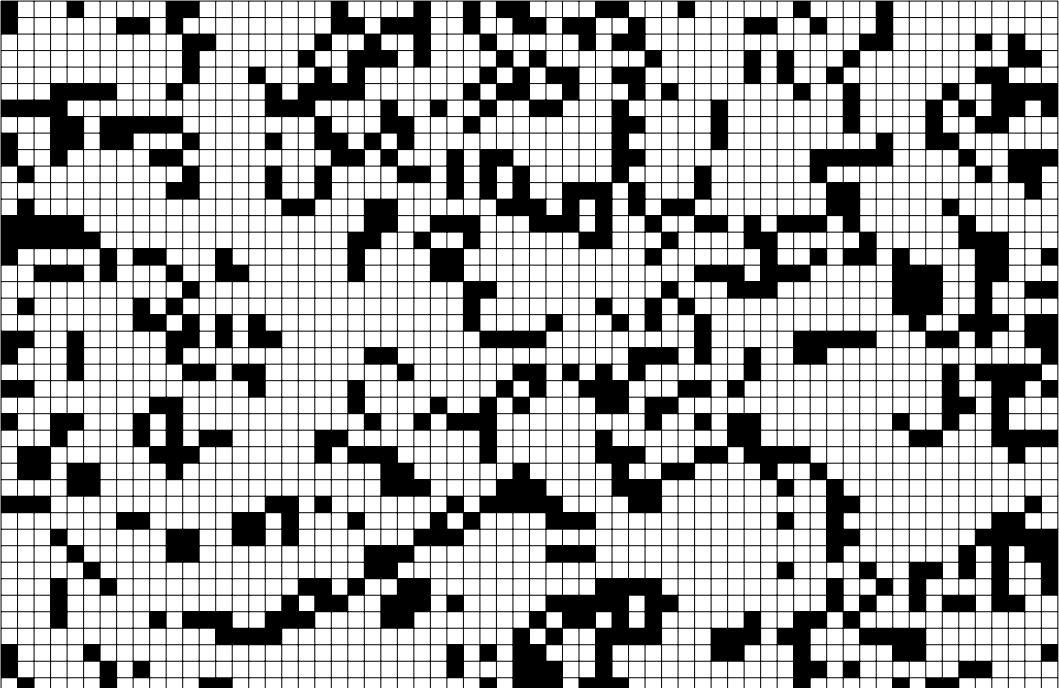
La **règle locale de transition** compte le nombre de cellules voisines vivantes parmis les huit qui entourent la cellule :

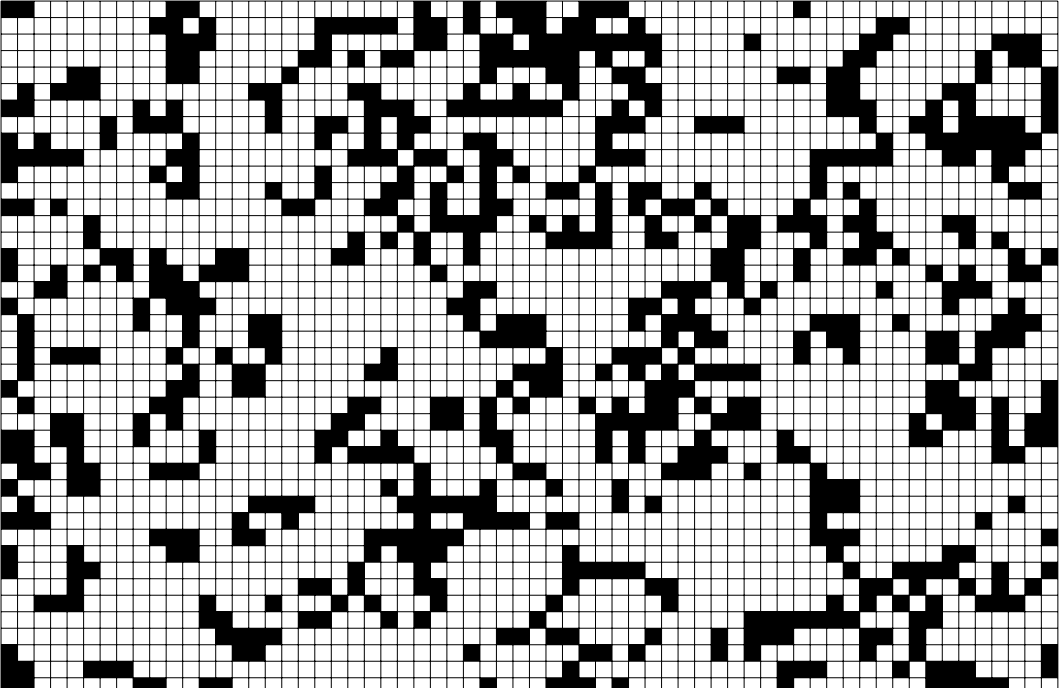
- **exactement trois** voisines vivantes font **naître** une cellule ;
- **moins de deux** voisines vivantes font mourir d'**isolement** ;
- **plus de trois** voisines vivantes font mourir d'**étouffement** ;
- dans les autres cas, la cellule conserve son état.

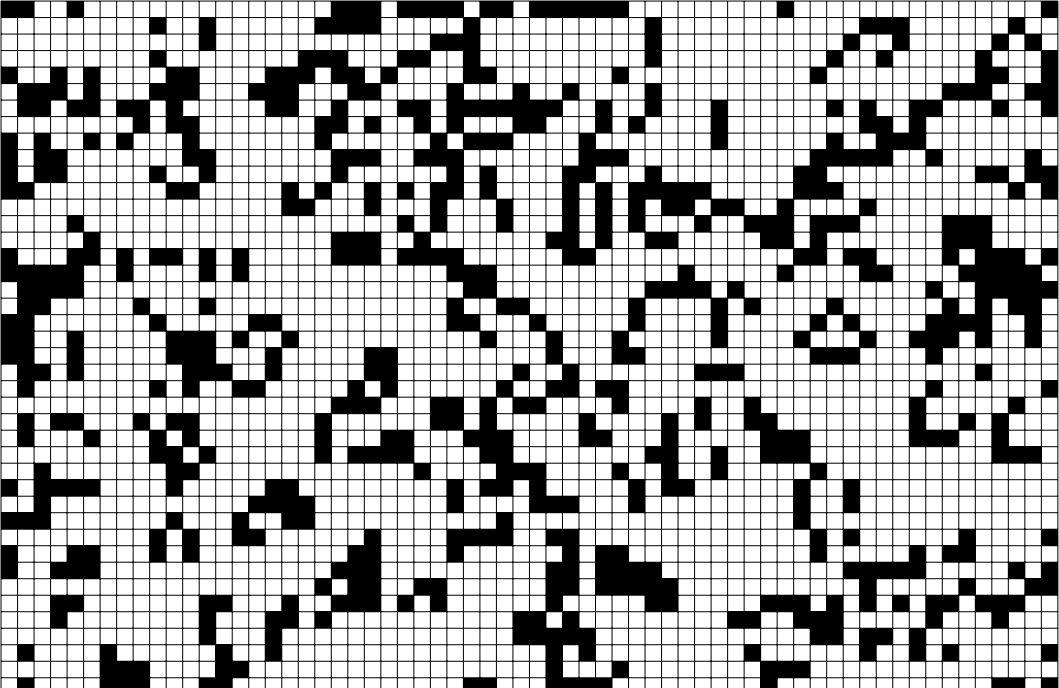


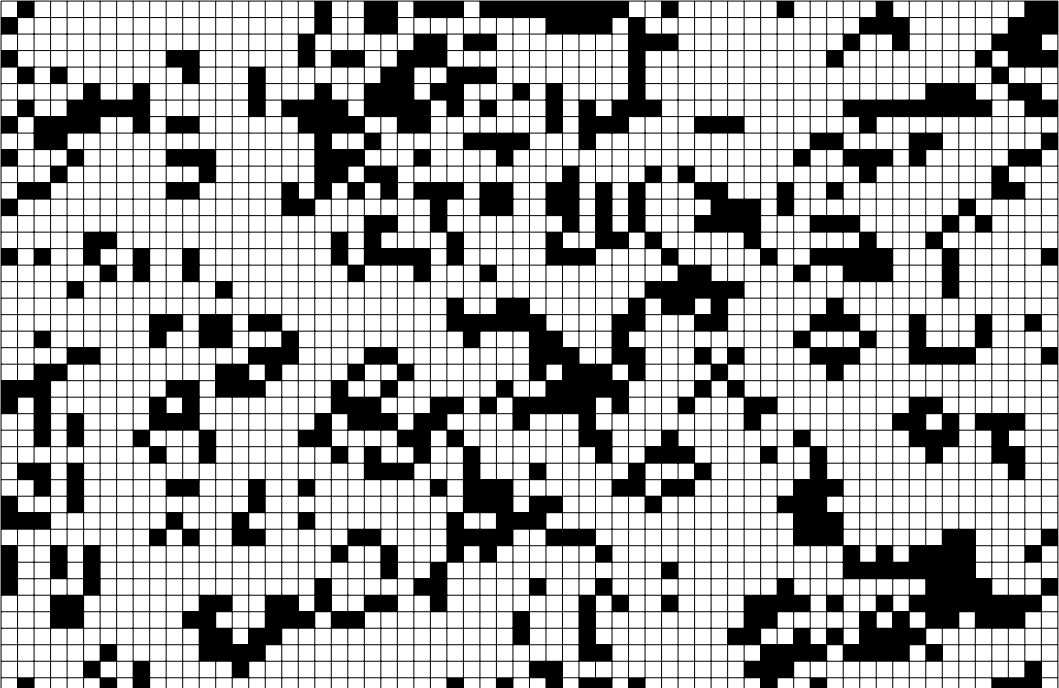


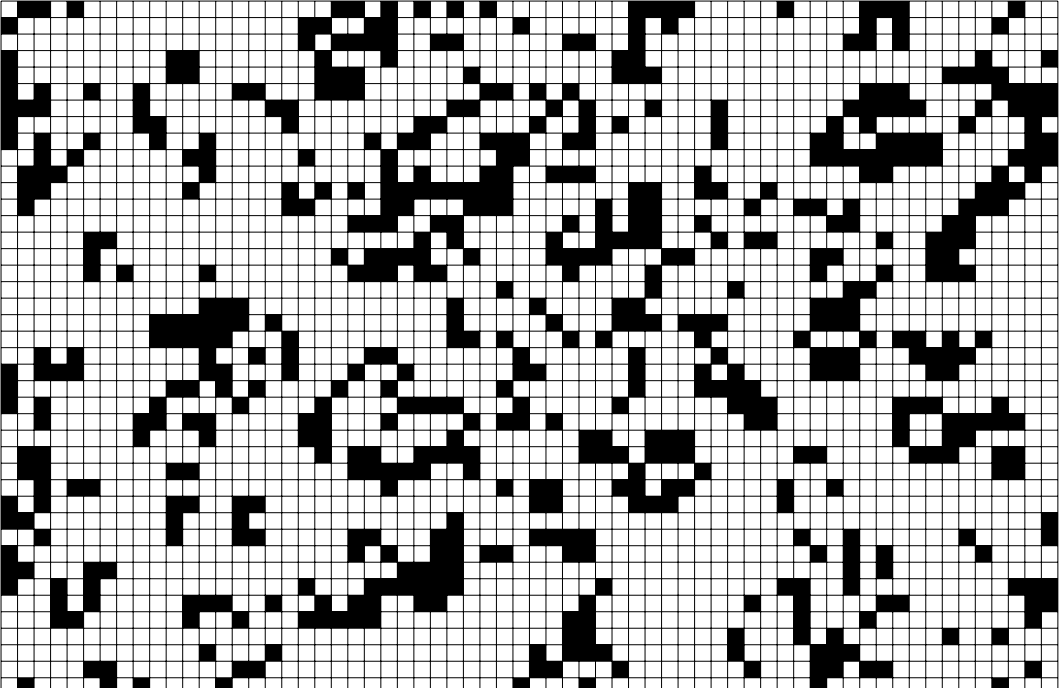


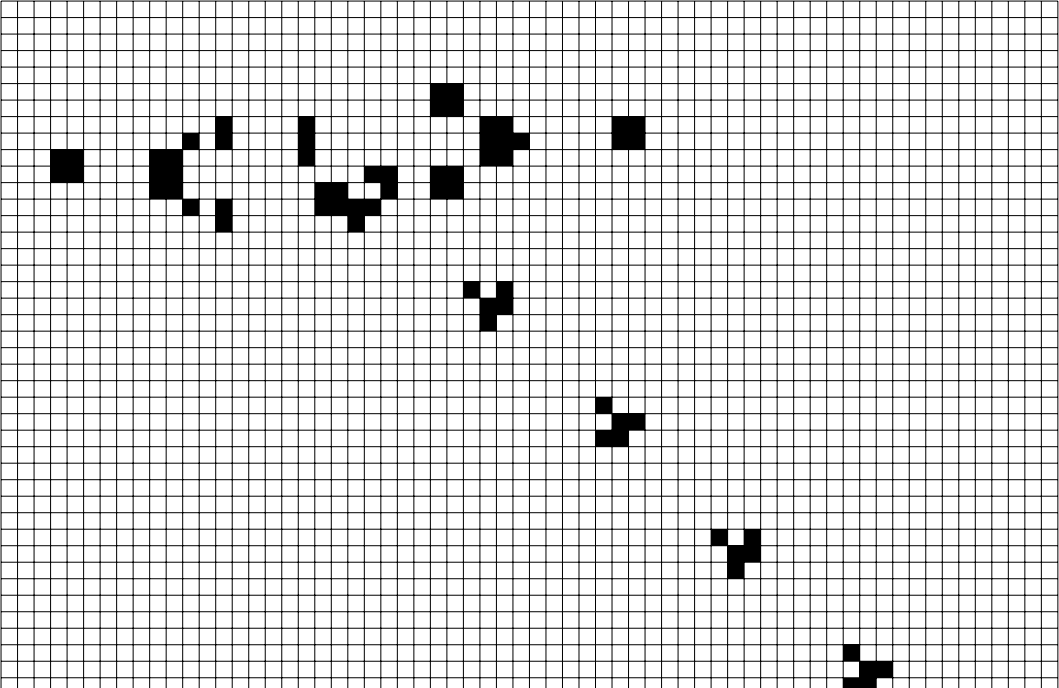


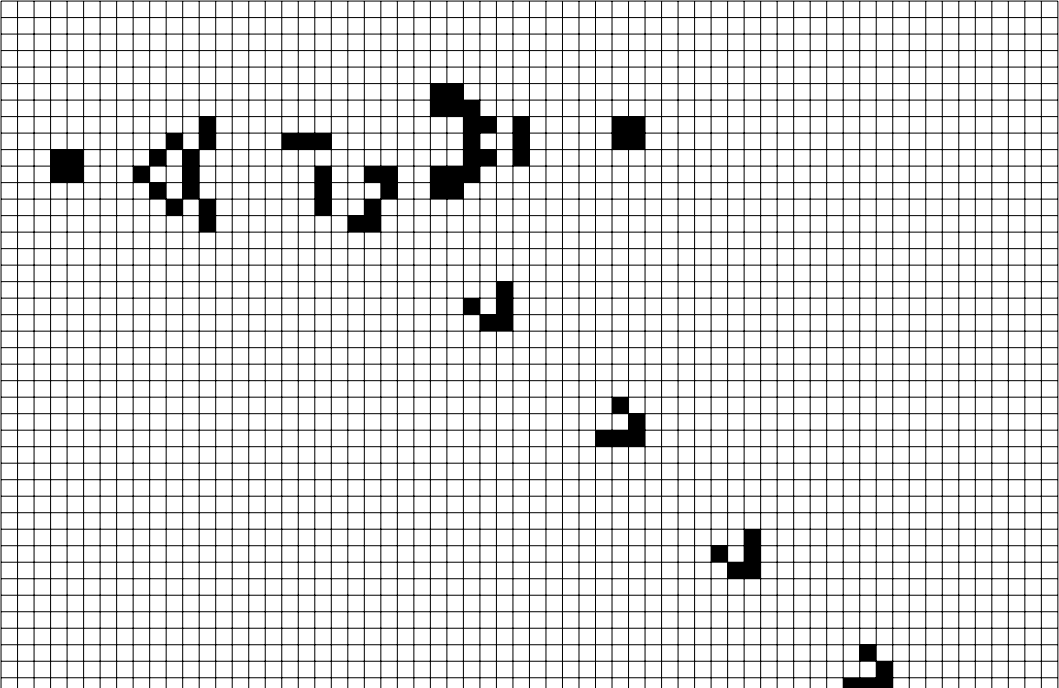


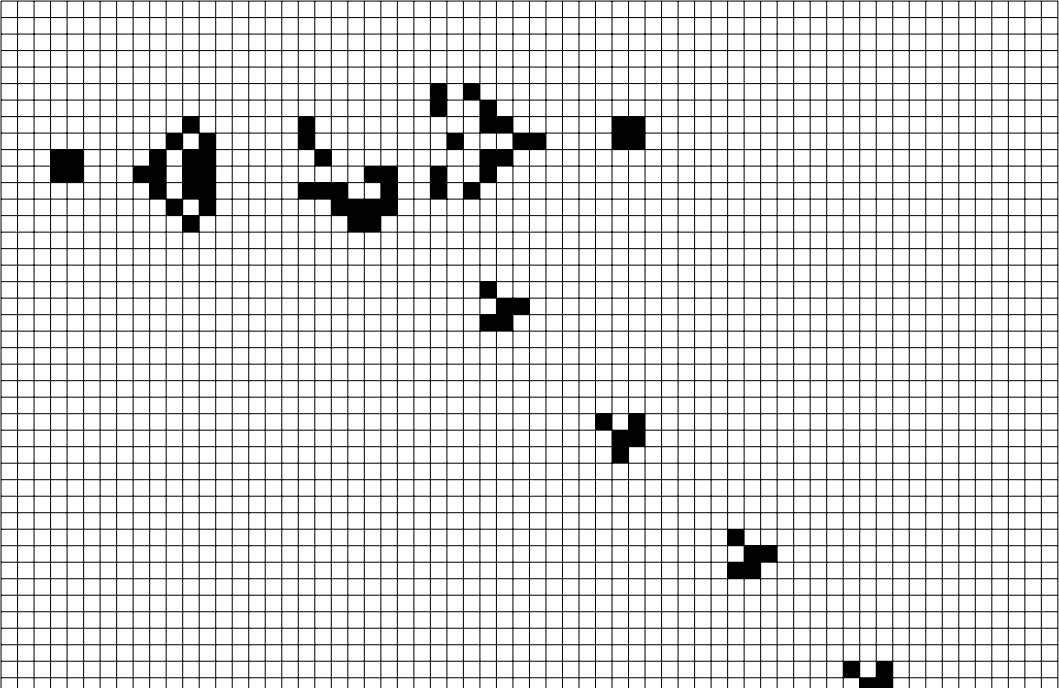


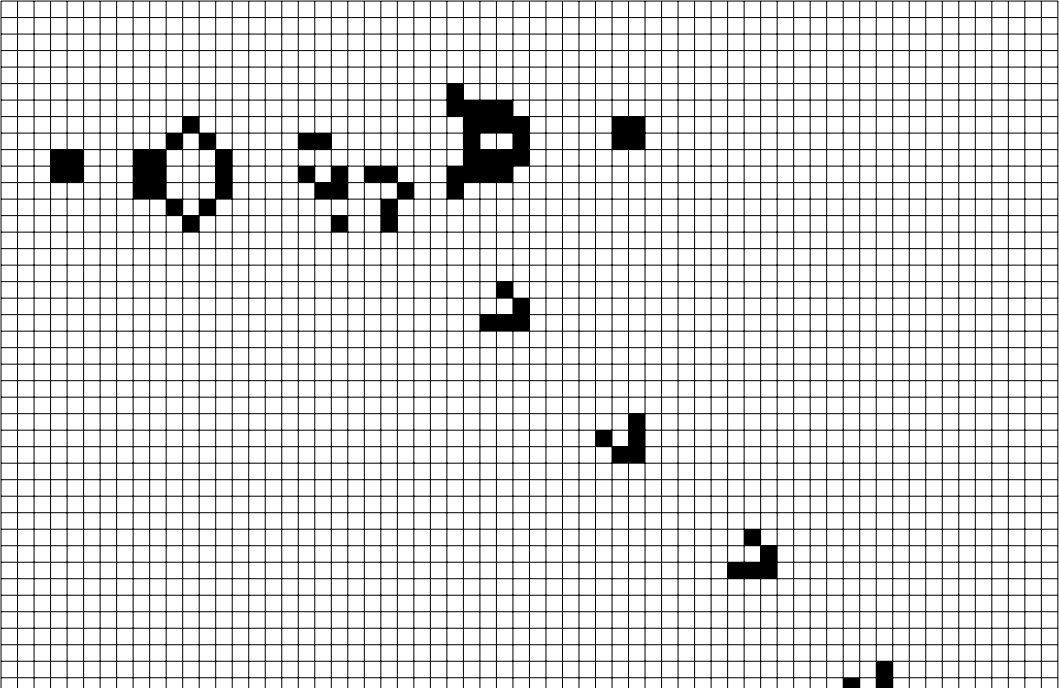


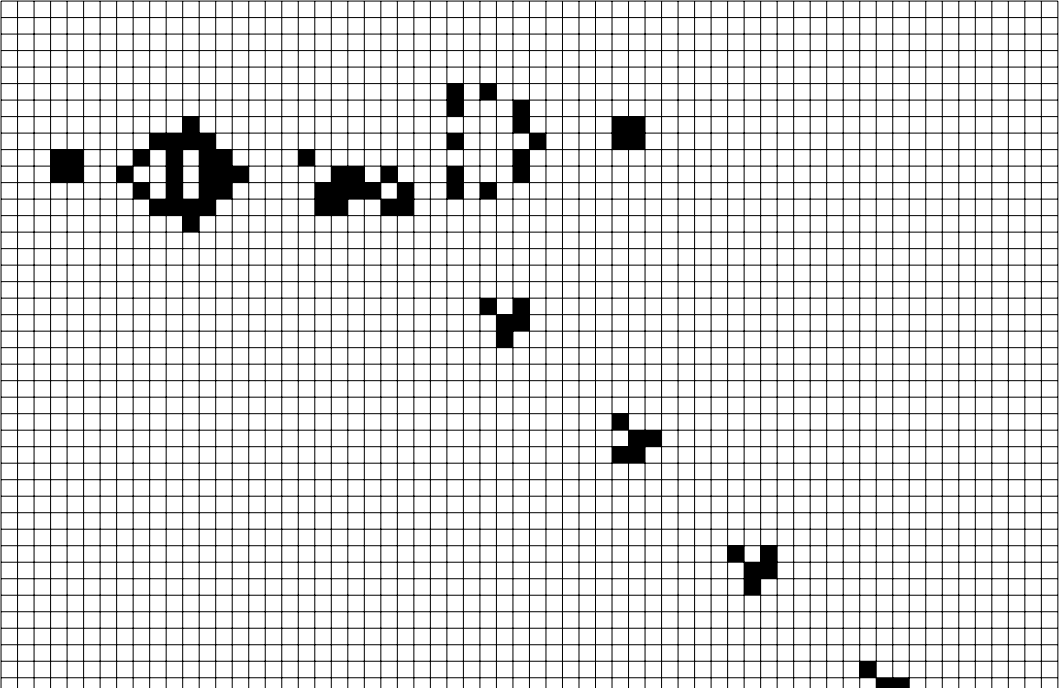


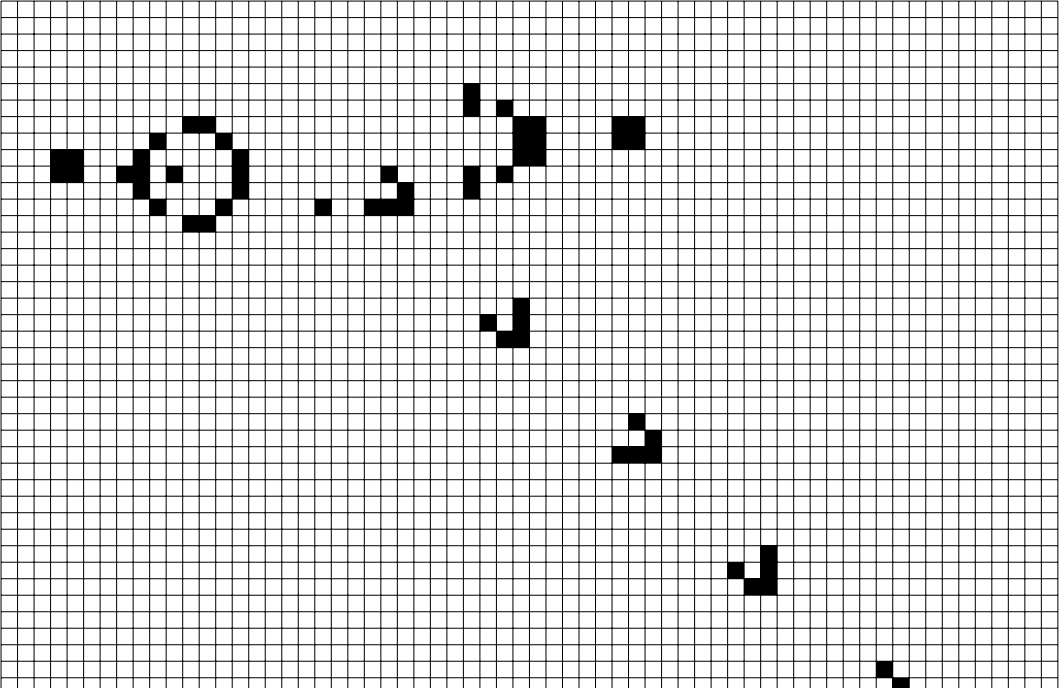


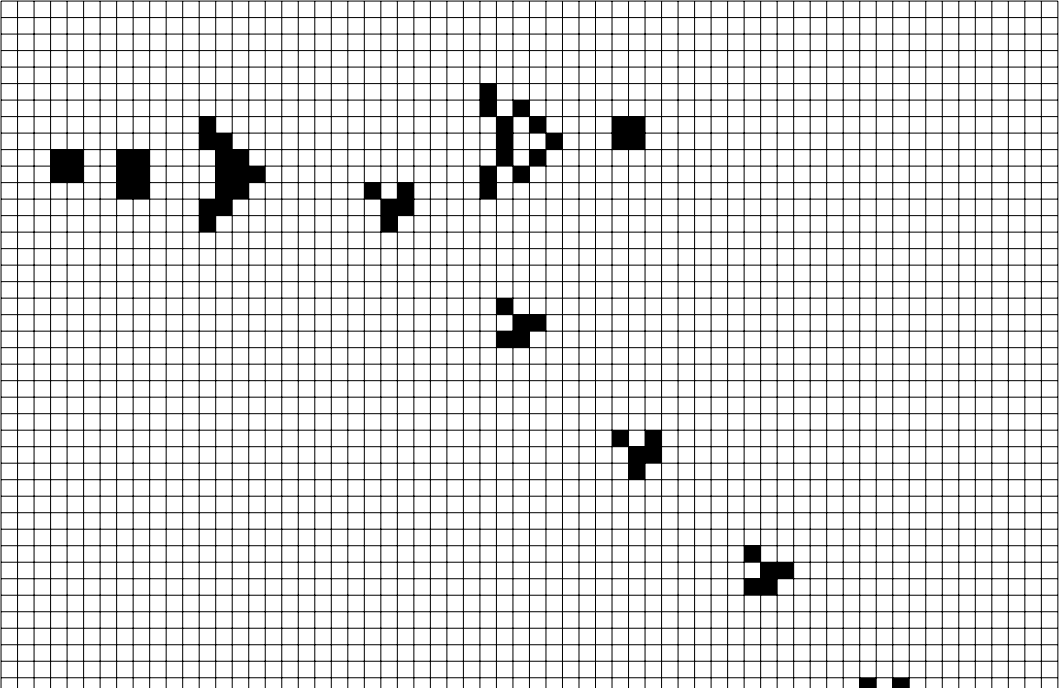


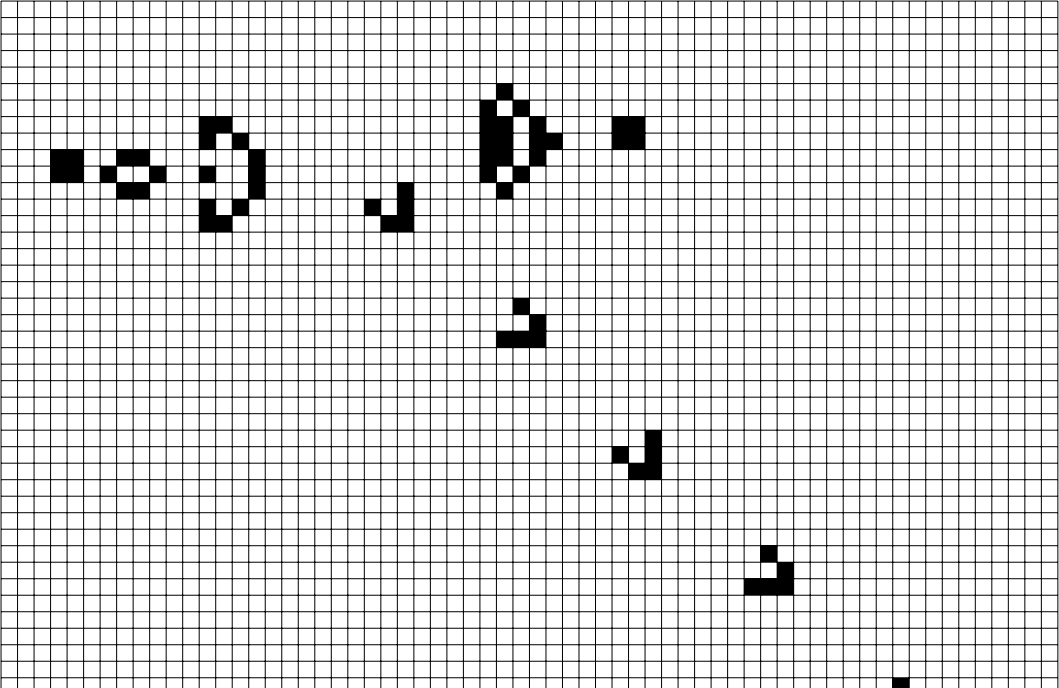




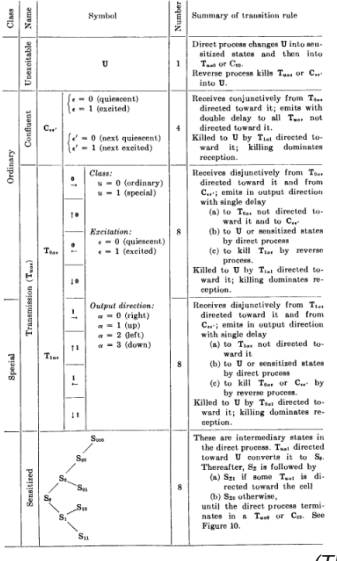






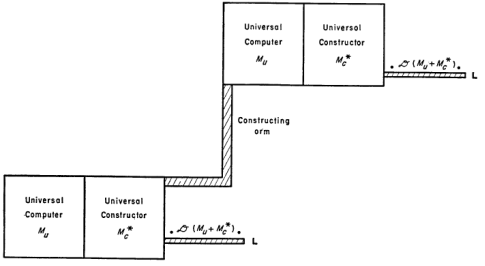


L'AC autoréplicateur de von Neumann



Class	Name	Symbol	Number	Summary of transition rule
	Unexcitable	U	1	Direct process changes U into sensitized states and then into T_{nsd} or C_{nd}. Reverse process kills T_{nsr} or C_{nr} into U.
Ordinary	Confluent	$C_{nr'}$ — $\sigma = 0$ (quiescent), $\sigma = 1$ (excited); $\sigma' = 0$ (next quiescent), $\sigma' = 1$ (next excited)	4	Receives conjunctively from T_{ksr} directed toward it; emits with double delay to all T_{nsr} not directed toward it. Killed to U by T_{isd} directed toward it; killing dominates reception.
Ordinary	Transmission (T_{nsr})	T_{nsr} — *Class:* $u = 0$ (ordinary), $u = 1$ (special); *Excitation:* $\epsilon = 0$ (quiescent), $\epsilon = 1$ (excited); *Output direction:* $\alpha = 0$ (right), $\alpha = 1$ (up), $\alpha = 2$ (left), $\alpha = 3$ (down)	8	Receives disjunctively from T_{ksr} directed toward it and from $C_{nr'}$; emits in output direction with single delay (a) to T_{nsr} not directed toward it and to $C_{nr'}$ (b) to U or sensitized states by direct process (c) to kill T_{isd} by reverse process. Killed to U by T_{isd} directed toward it; killing dominates reception.
Special	Transmission (T_{nsr})	T_{nsr} — *Output direction:* $\alpha = 0$ (right), $\alpha = 1$ (up), $\alpha = 2$ (left), $\alpha = 3$ (down)	8	Receives disjunctively from T_{ksr} directed toward it and from $C_{nr'}$; emits in output direction with single delay (a) to T_{nsr} not directed toward it (b) to U or sensitized states by direct process (c) to kill T_{nsr} or $C_{nr'}$ by reverse process. Killed to U by T_{isd} directed toward it; killing dominates reception.
	Sensitized	S_{00}, S_{01}, S_{02}, S_{03}, S_{10}, S_{11}, S_{12}, S_{13}, S_{20}, S_{21}, S_{22}, S_{23}, S_{30}, S_{31}, S_{32}, S_{33}	8	These are intermediary states in the direct process. T_{nsd} directed toward U converts it to S_0. Thereafter, S_β is followed by (a) $S_{\beta 1}$ if some T_{ksr} is directed toward the cell (b) $S_{\beta 0}$ otherwise, until the direct process terminates in a T_{nsd} or C_{nd}. See Figure 10.

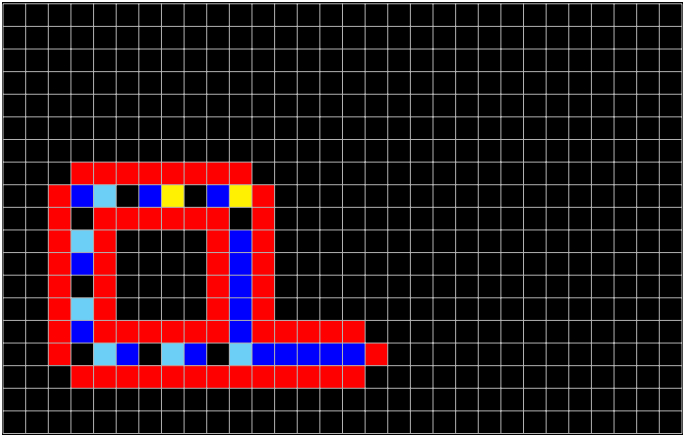
Un AC à **29 états** et voisinage de **von Neumann** réalisant signaux et capacité de construction/destruction.



Autoréplicant par **calculateur universel** + **constructeur universel**.

(Theory of Self-Reproducing Automata, edited by Burks, 1966)

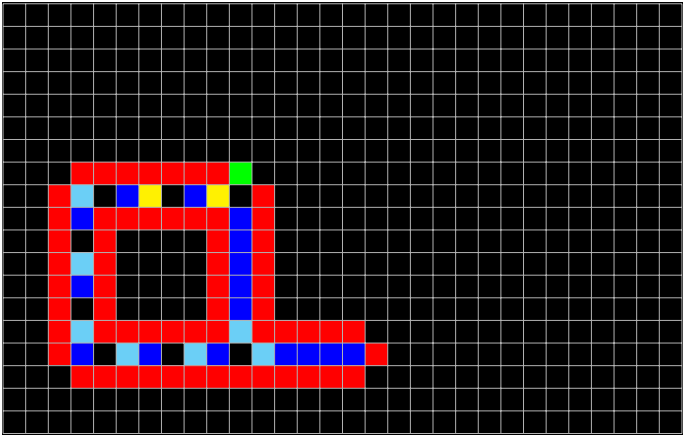
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

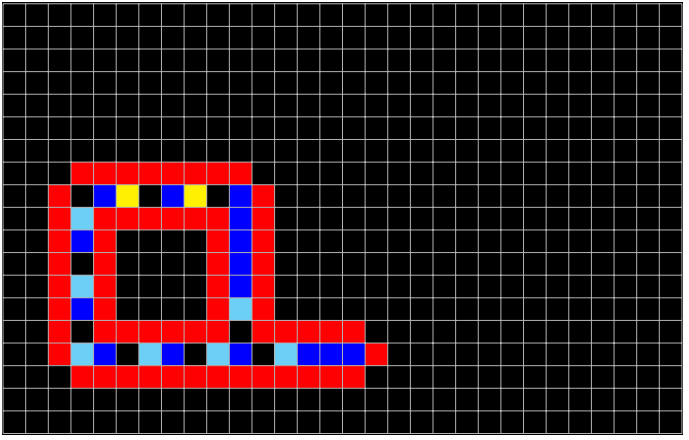
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

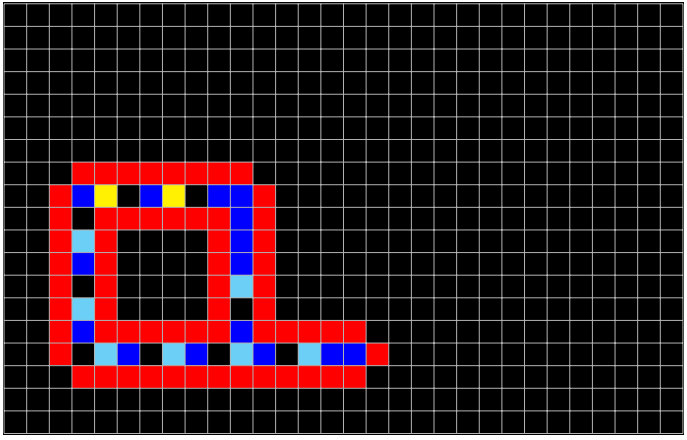
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

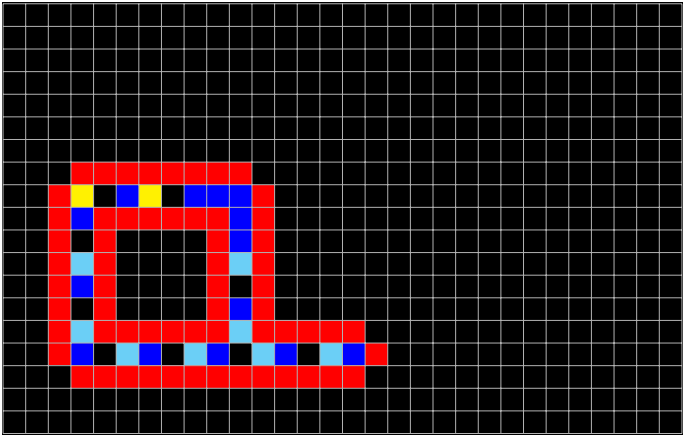
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

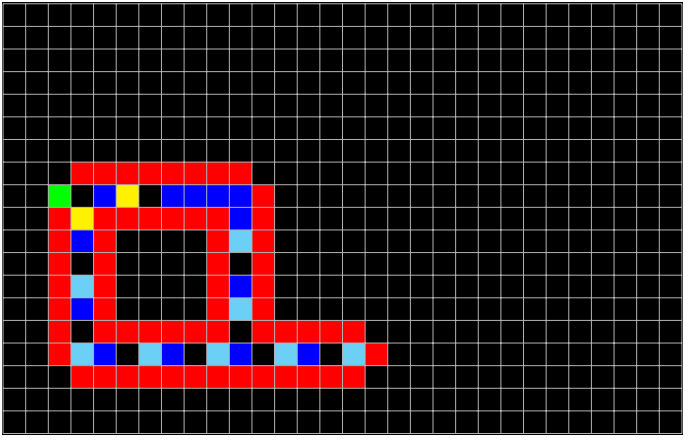
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

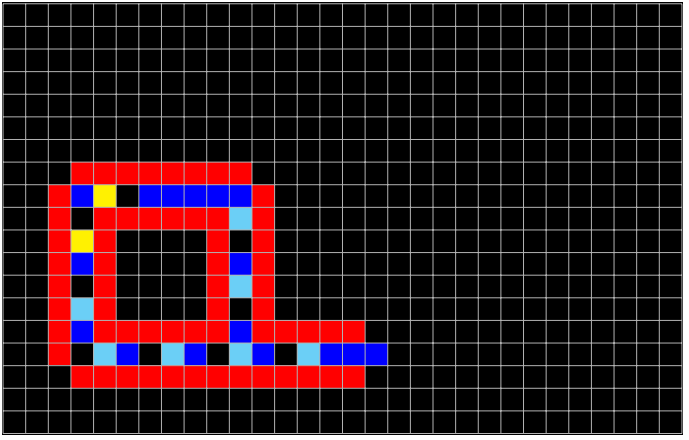
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

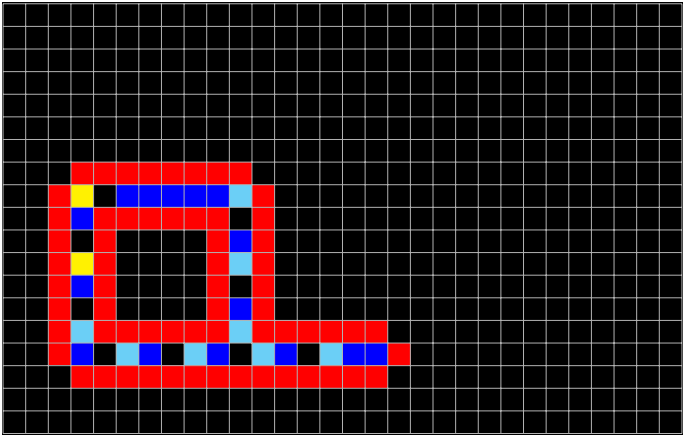
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

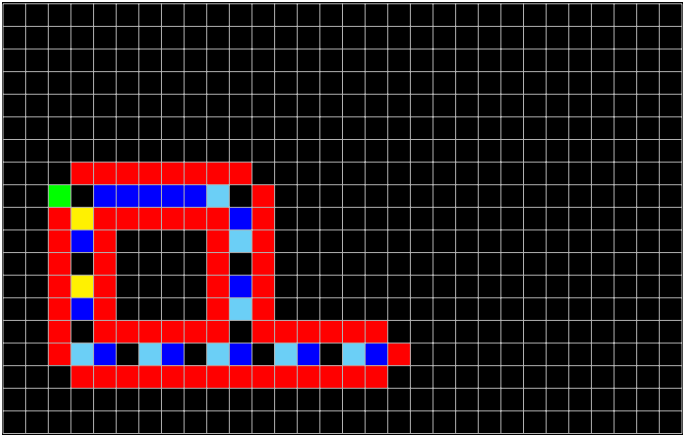
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

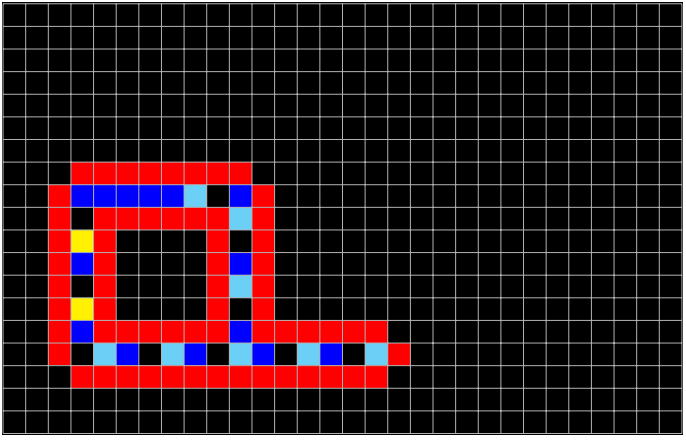
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

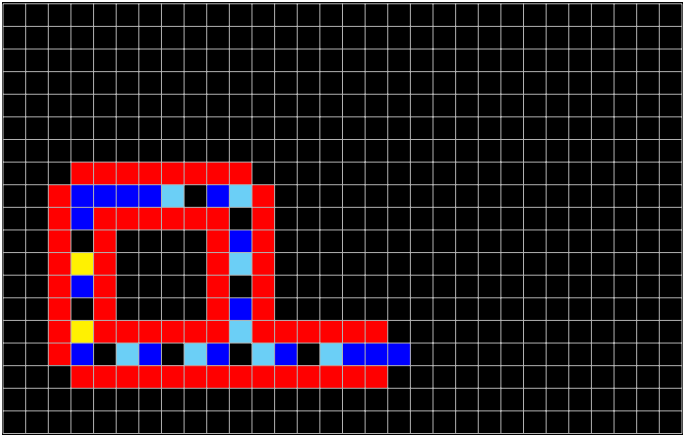
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

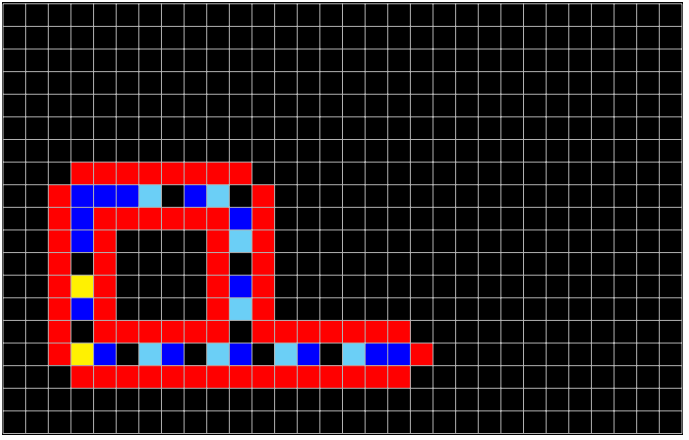
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

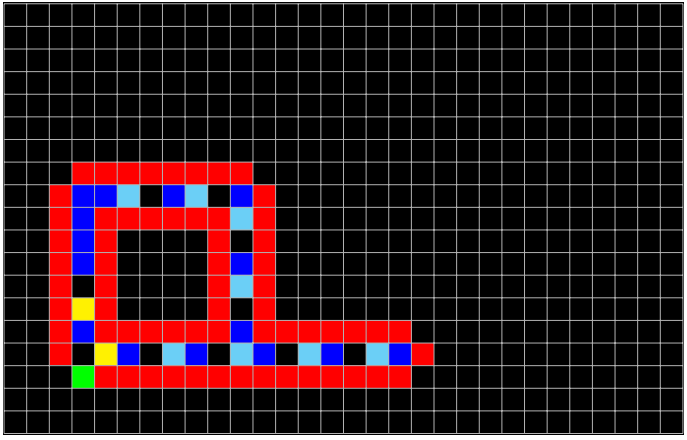
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

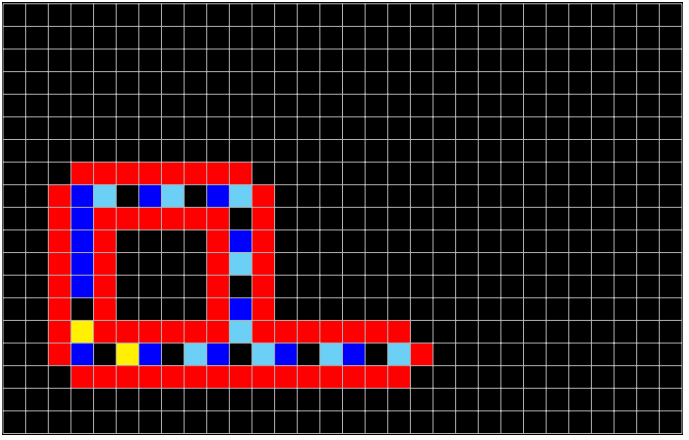
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

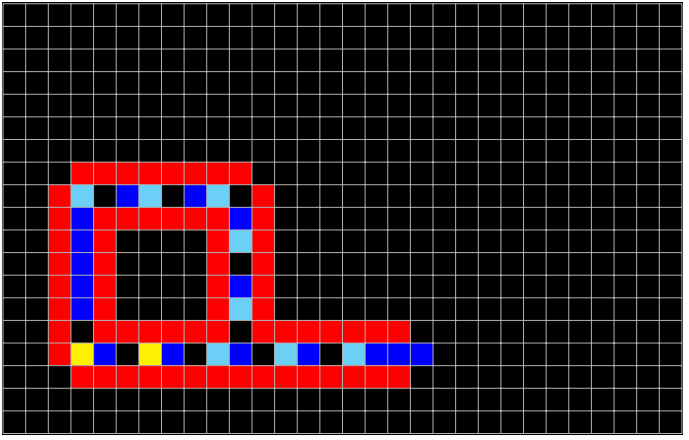
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

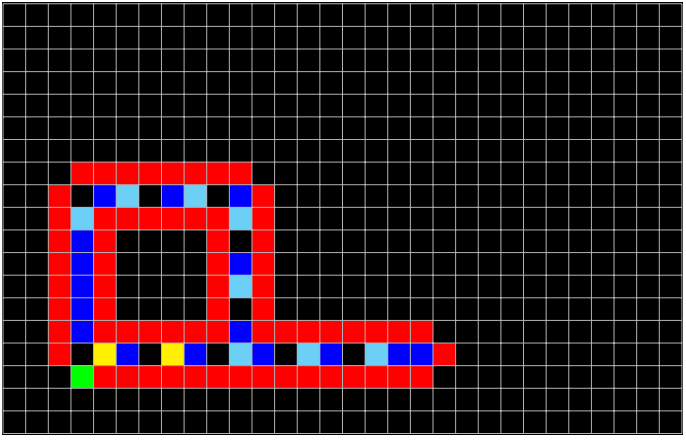
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

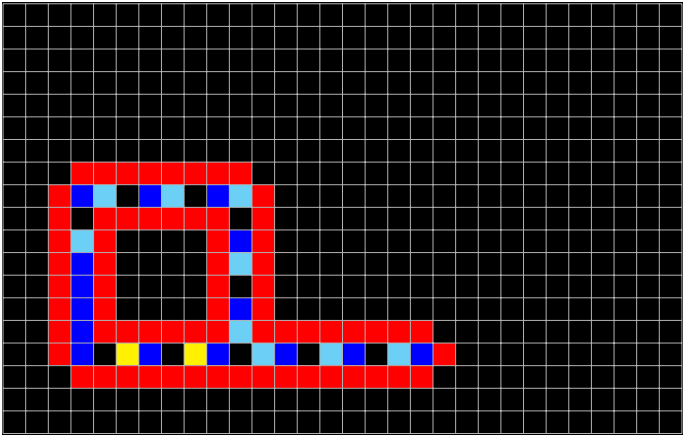
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

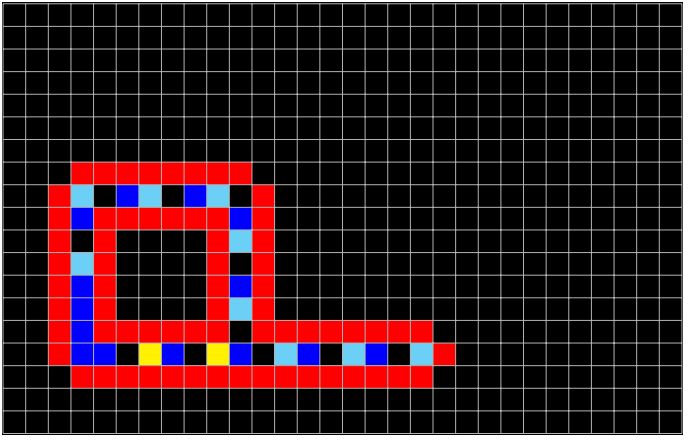
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

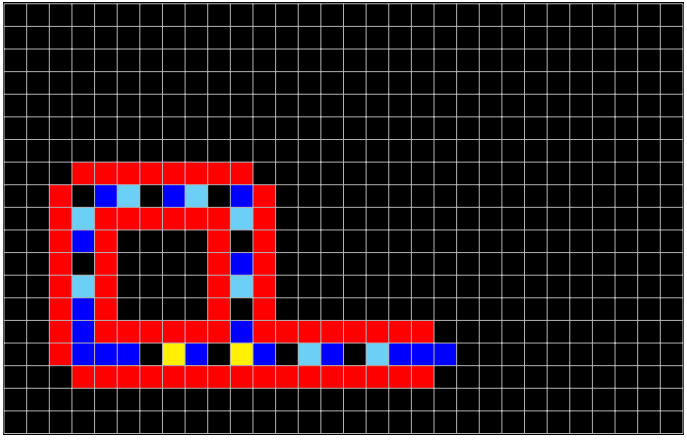
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

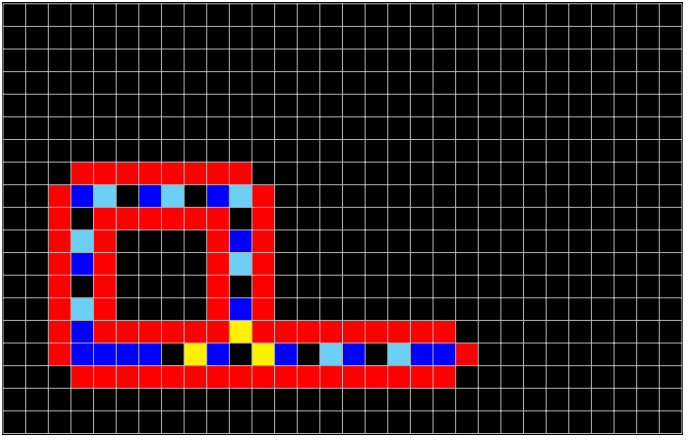
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

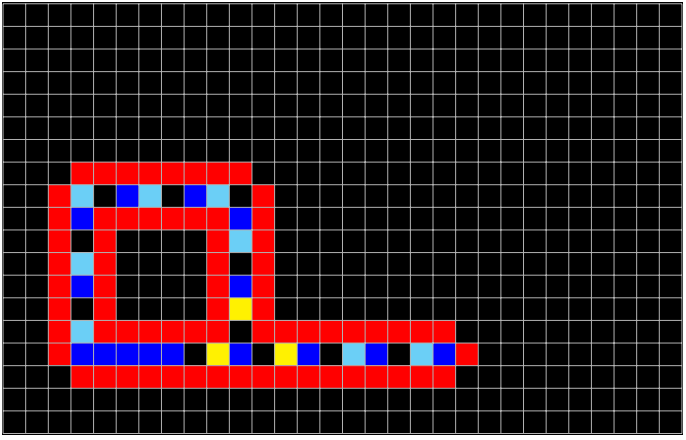
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

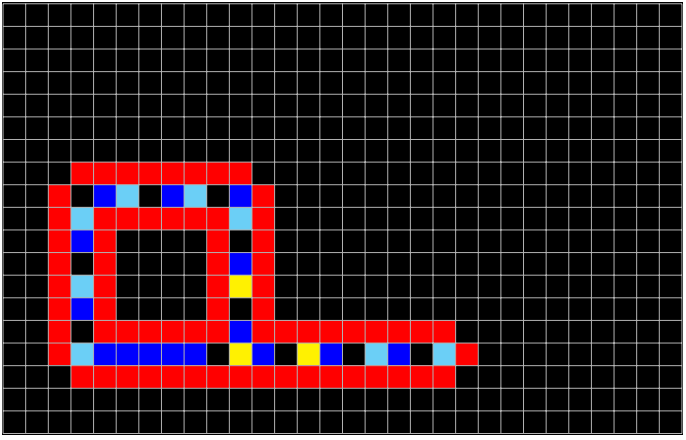
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

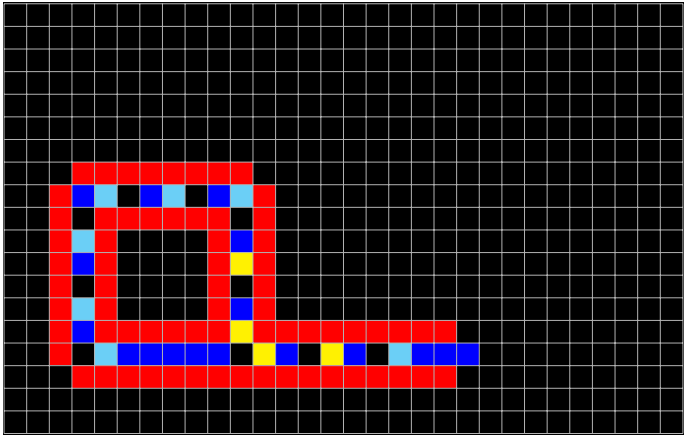
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

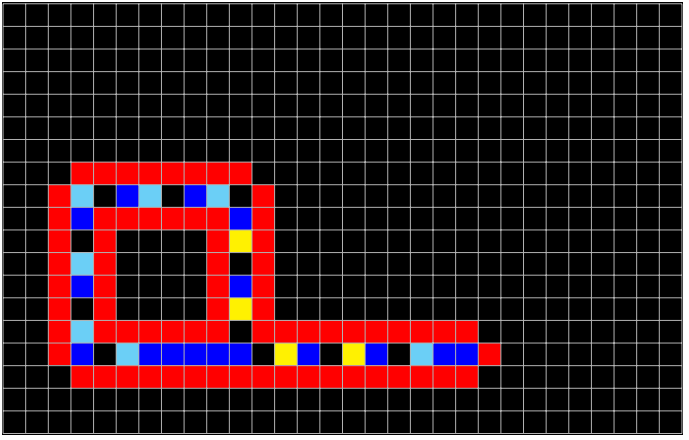
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

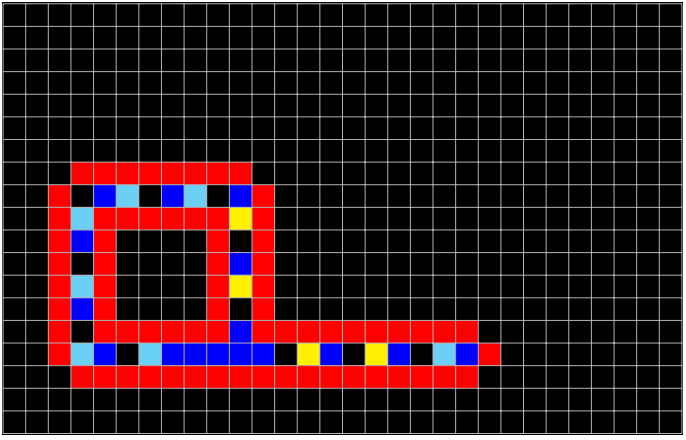
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

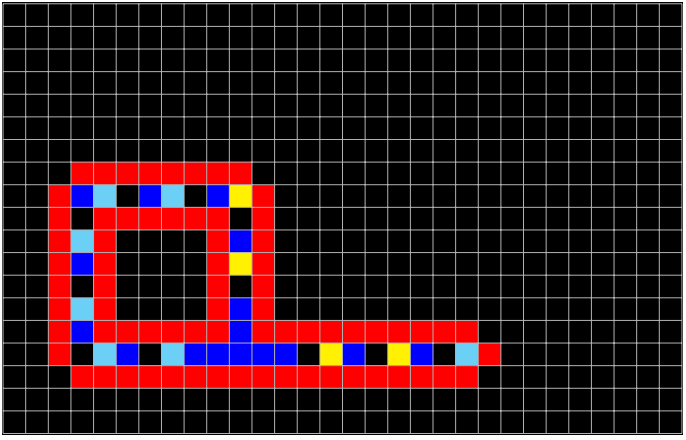
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

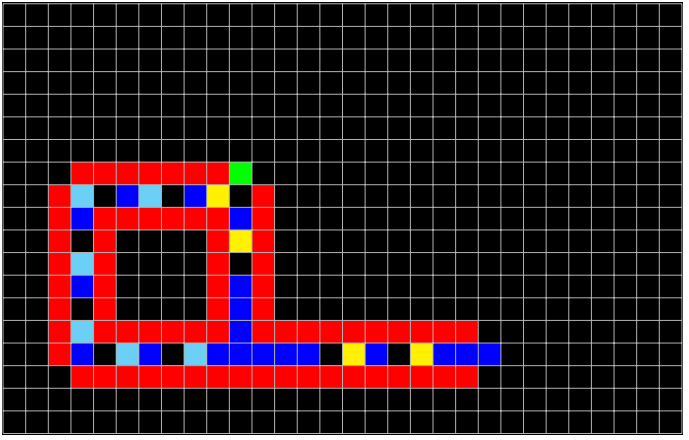
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

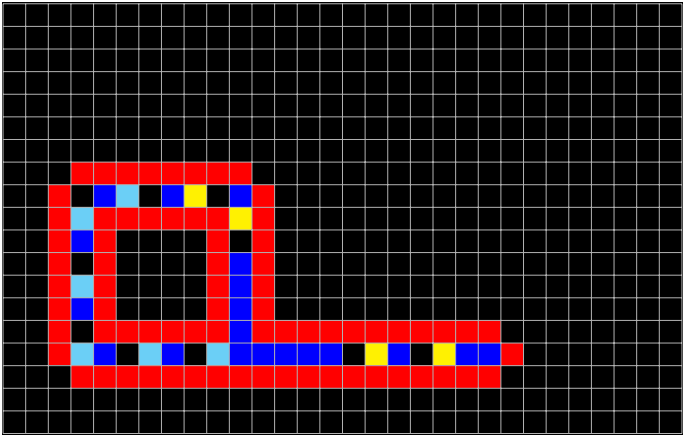
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

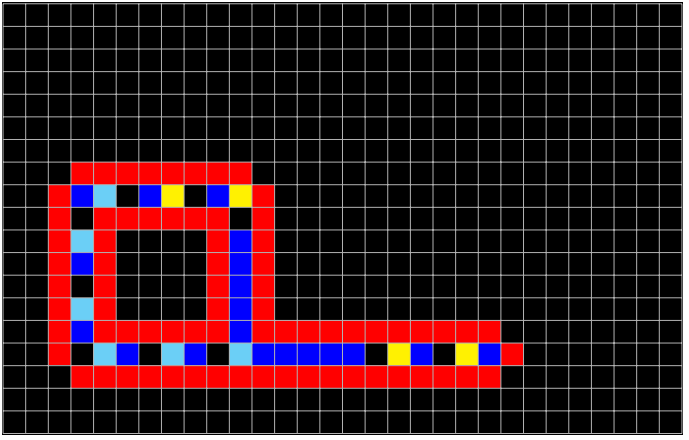
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

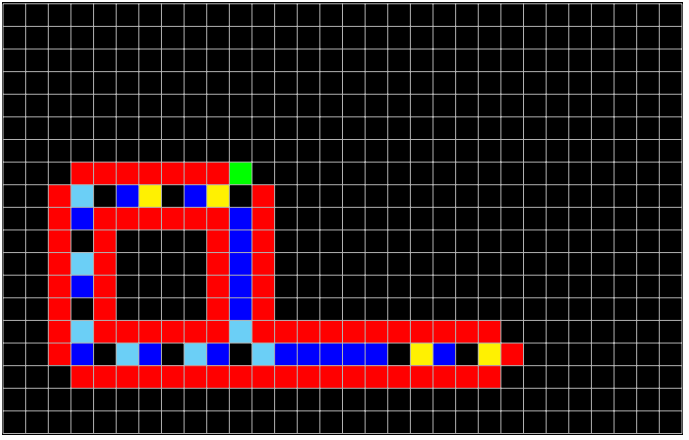
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

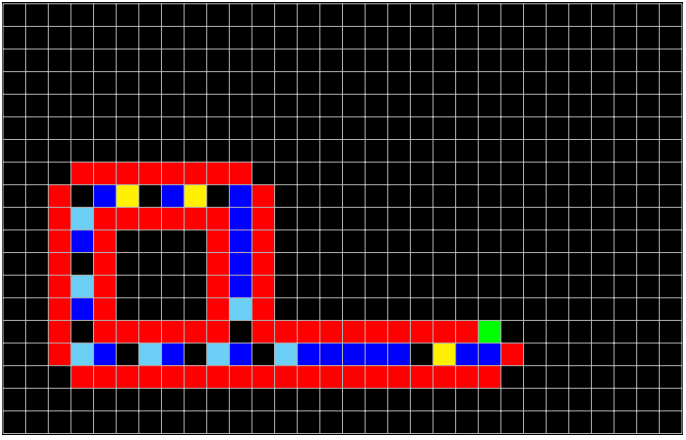
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

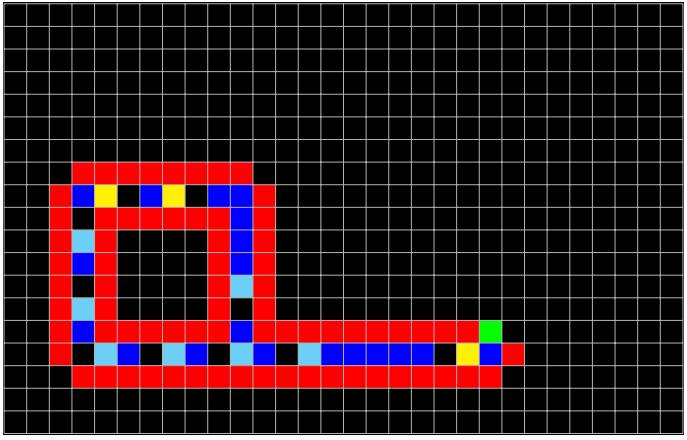
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

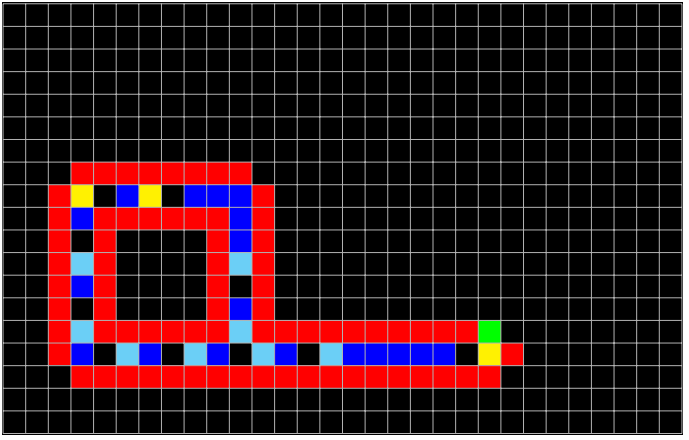
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

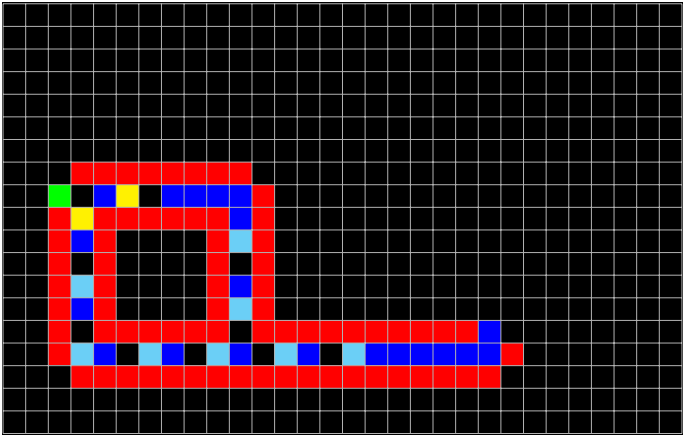
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

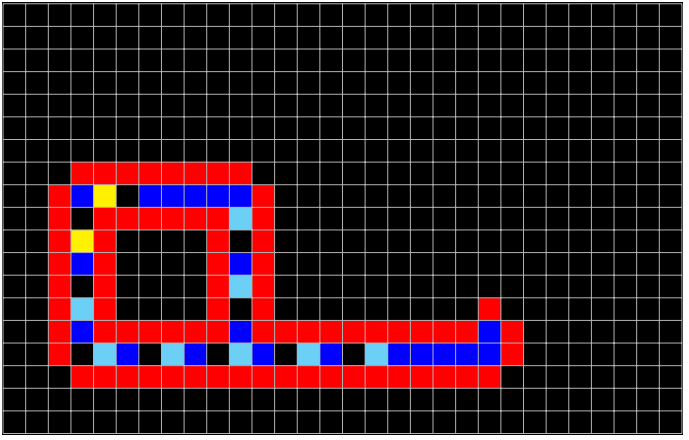
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

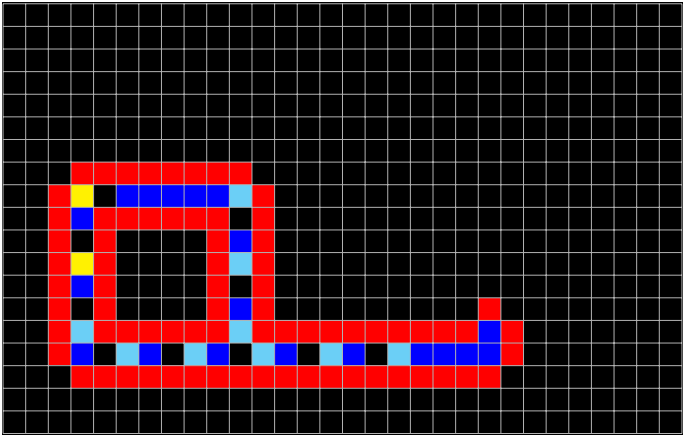
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

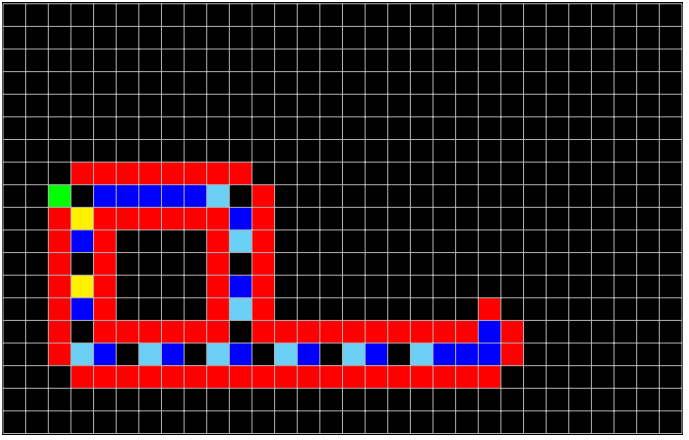
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

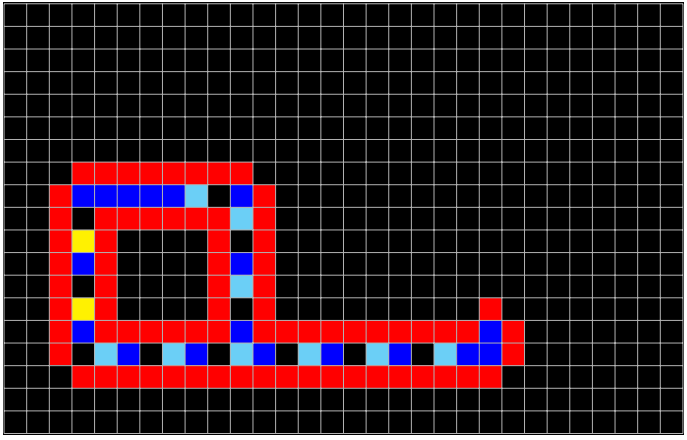
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

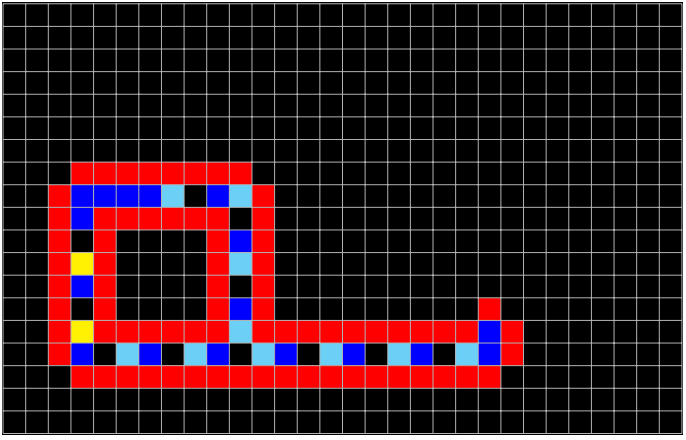
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

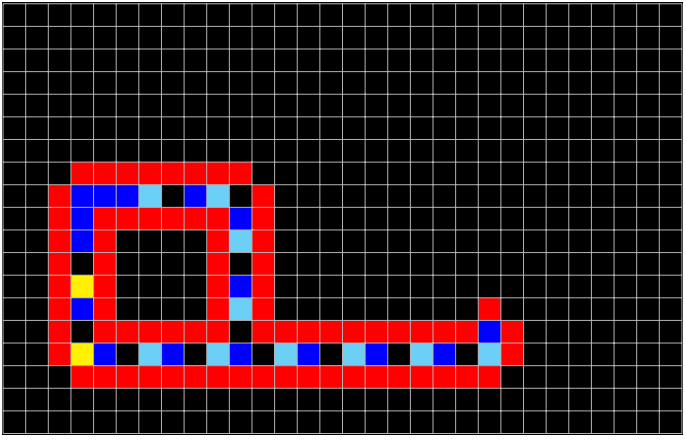
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

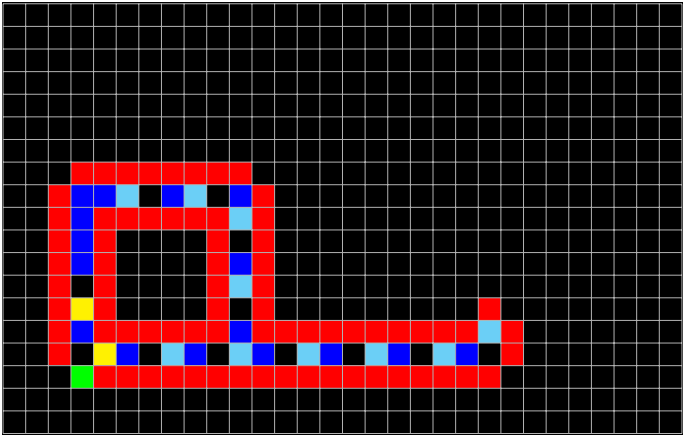
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

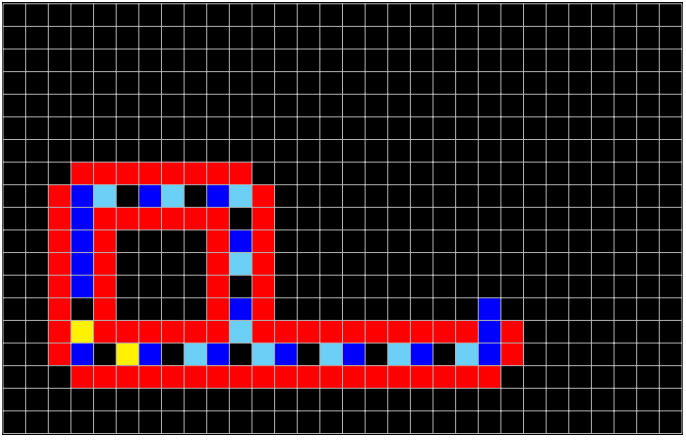
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

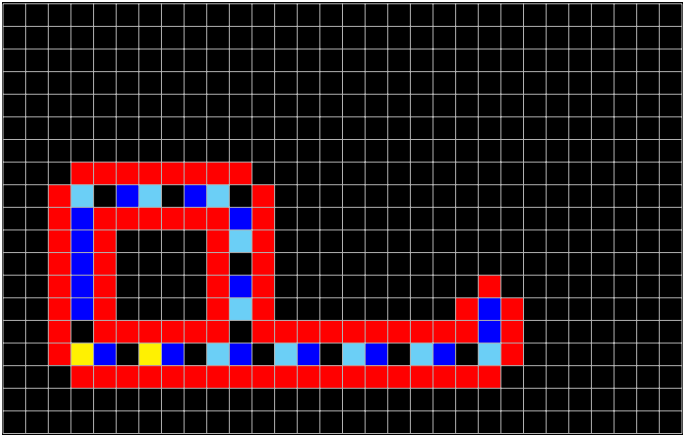
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

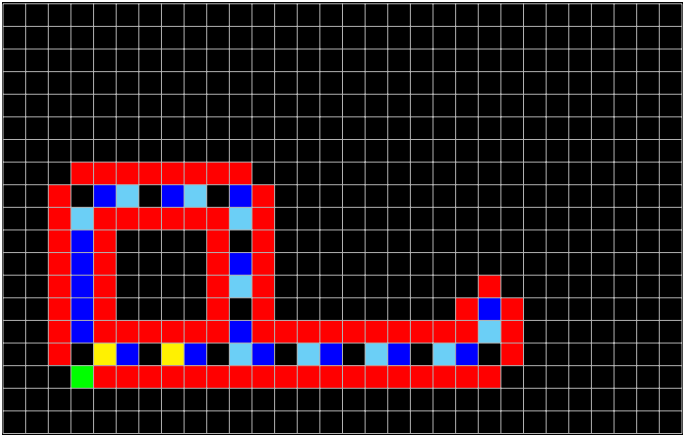
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

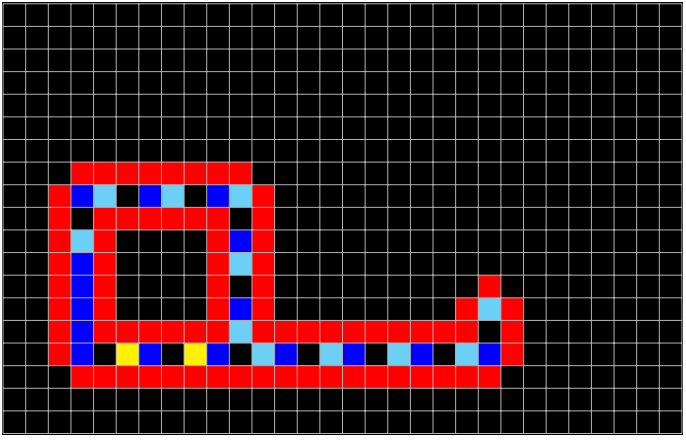
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

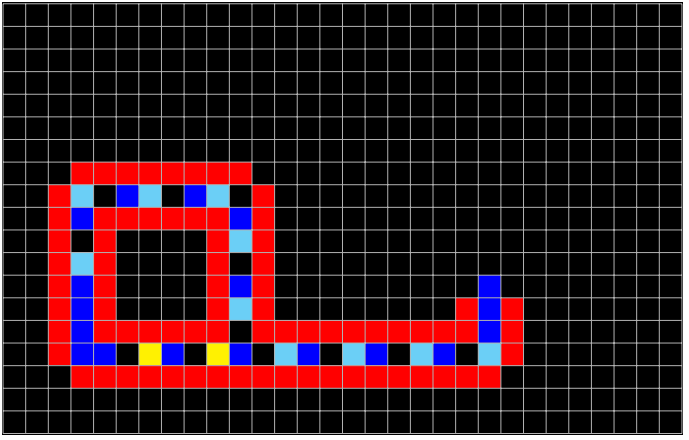
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

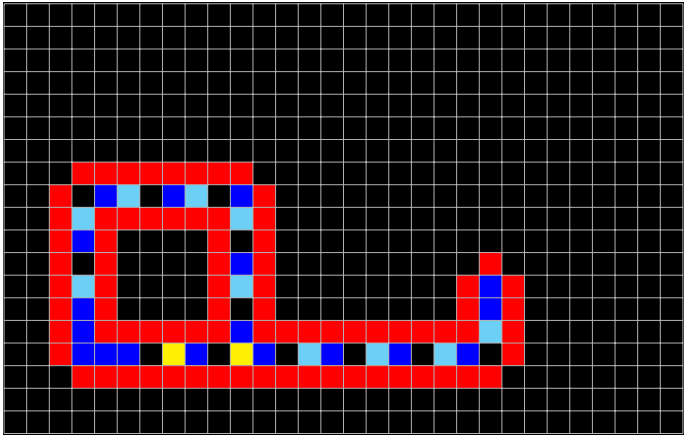
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

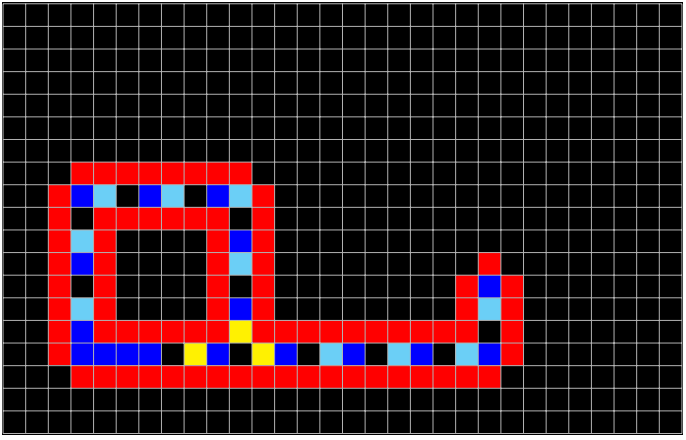
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

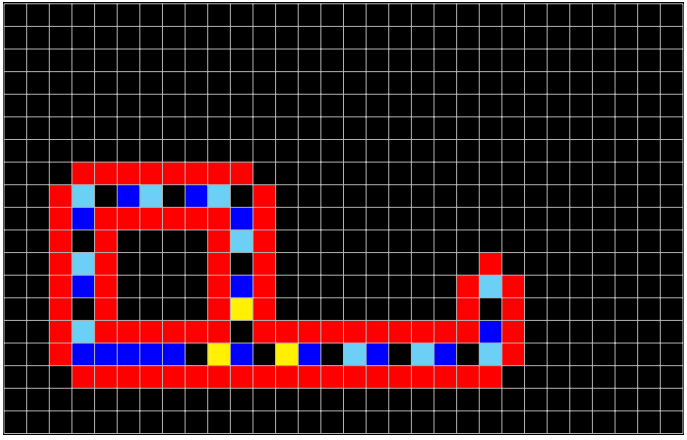
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

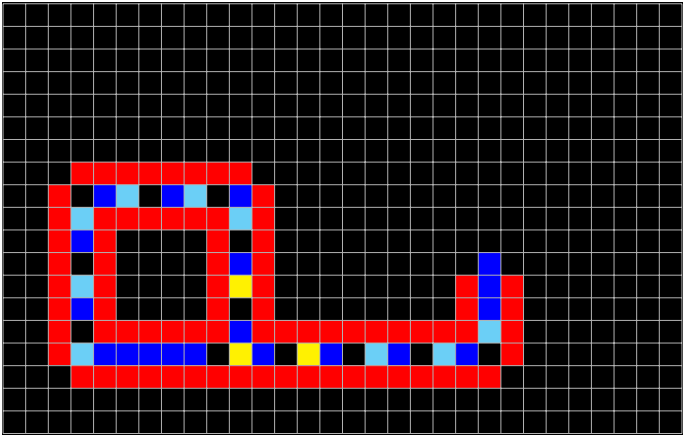
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

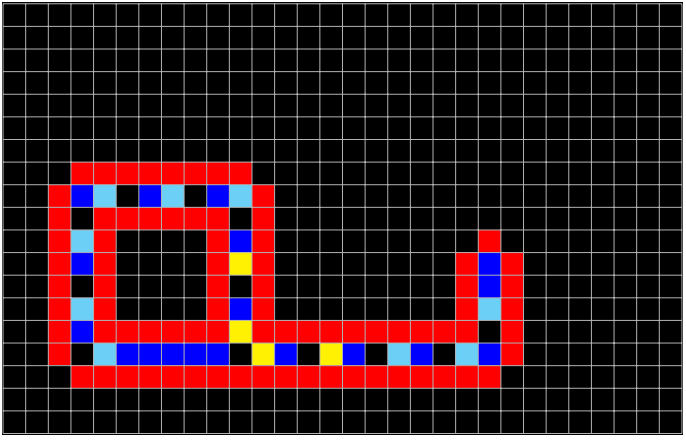
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

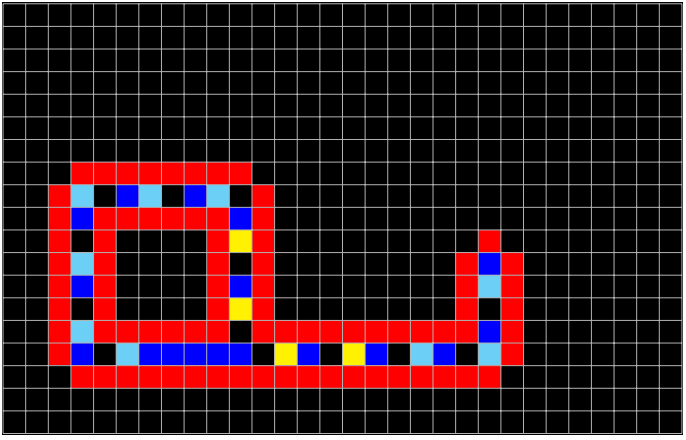
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

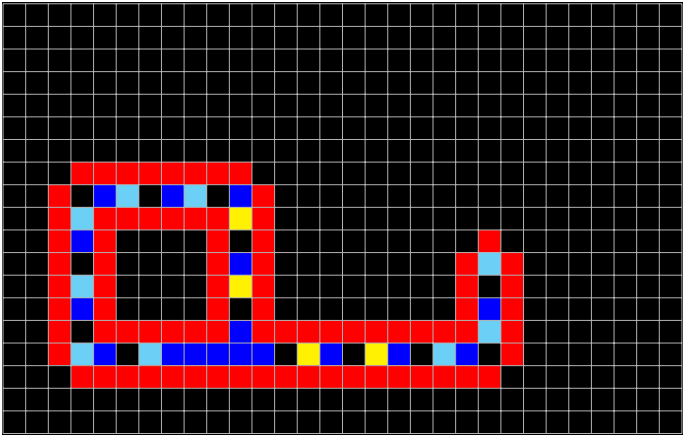
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

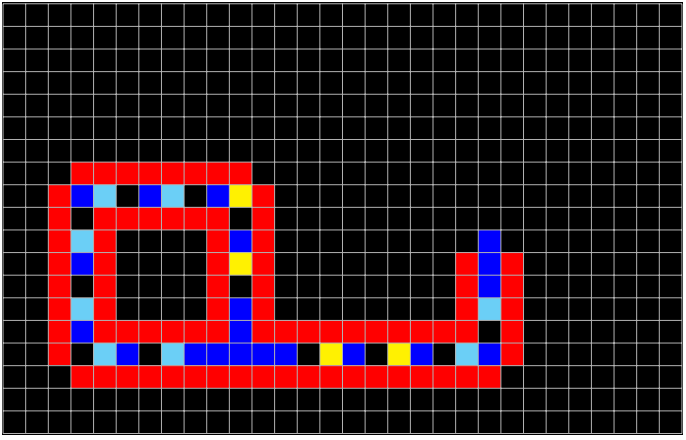
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

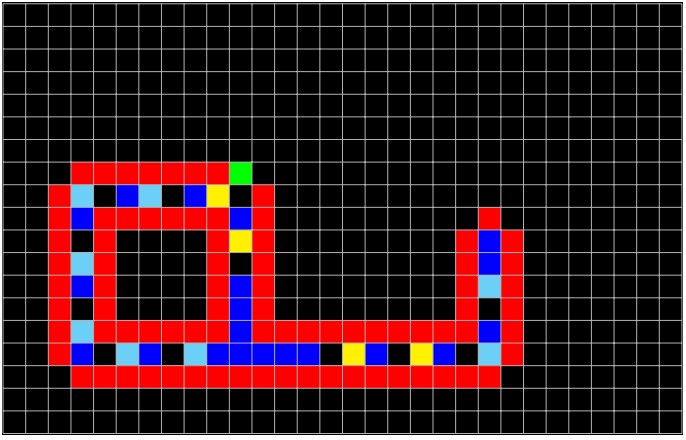
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

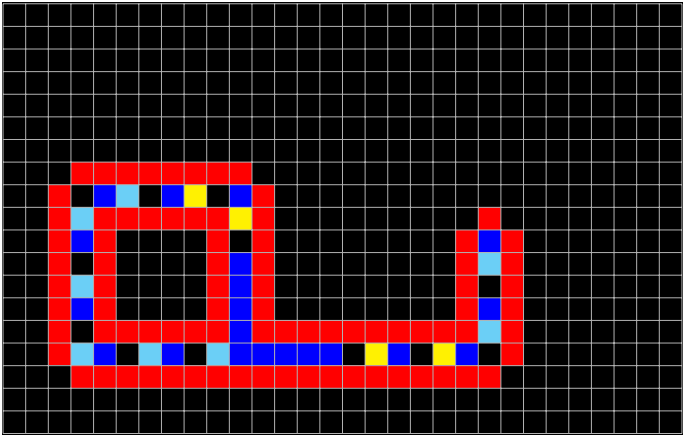
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

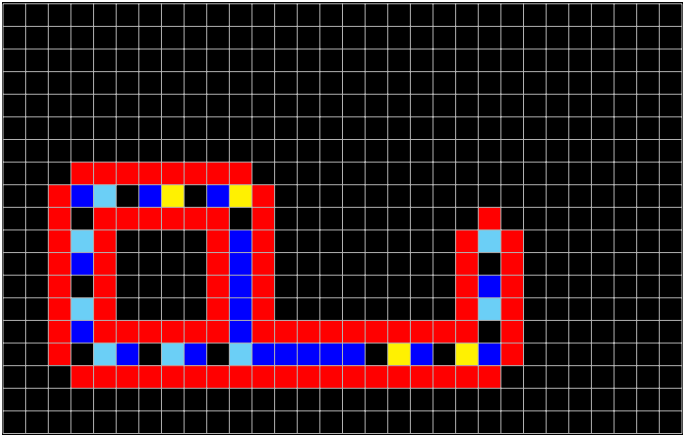
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

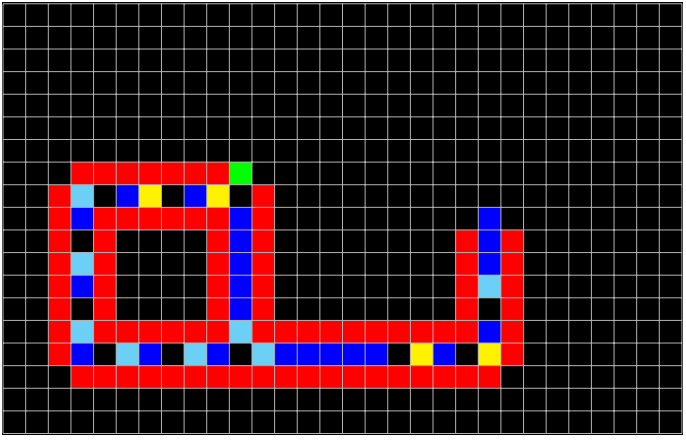
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

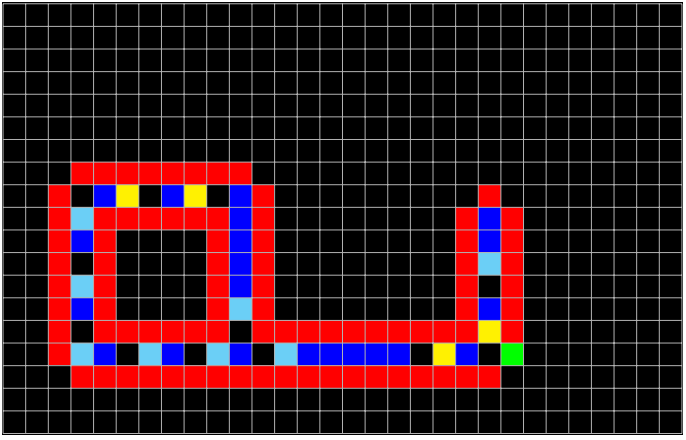
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

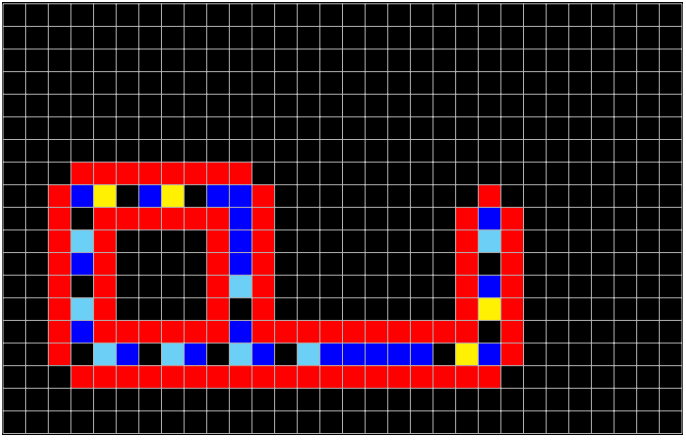
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

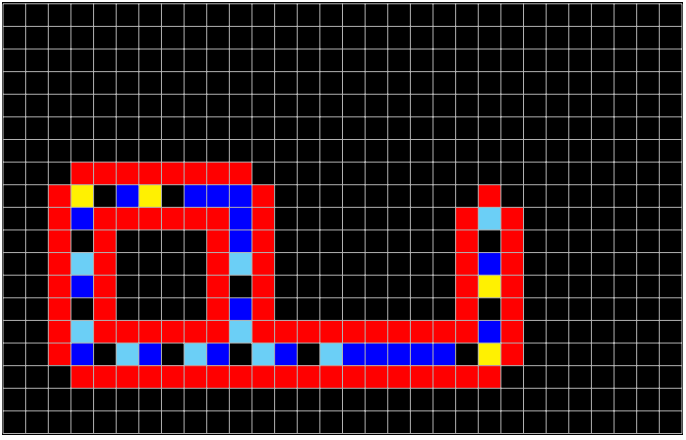
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

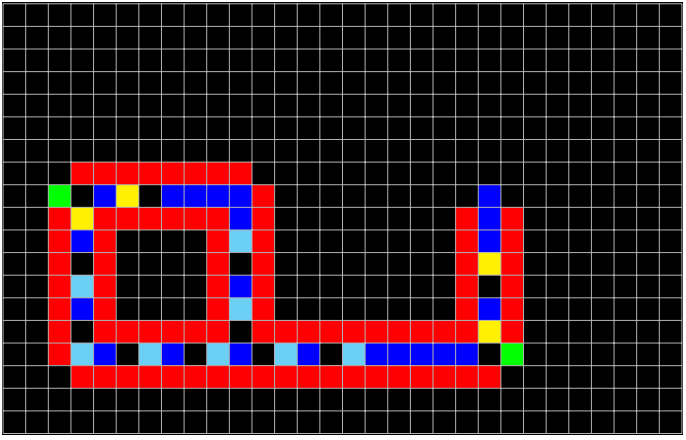
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

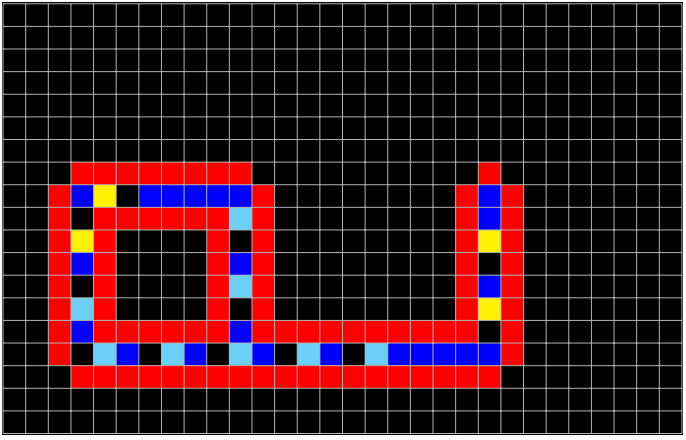
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

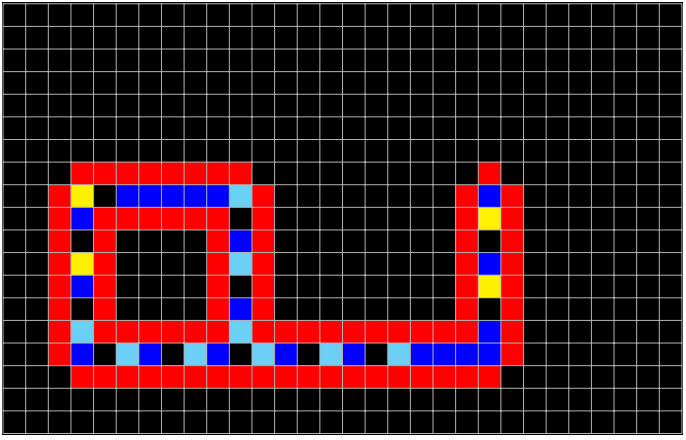
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

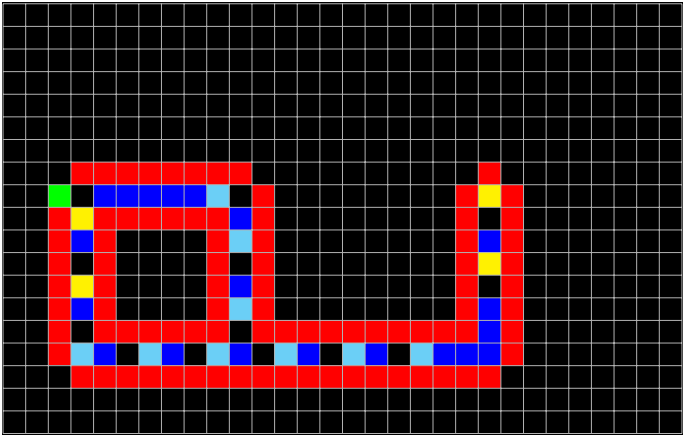
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

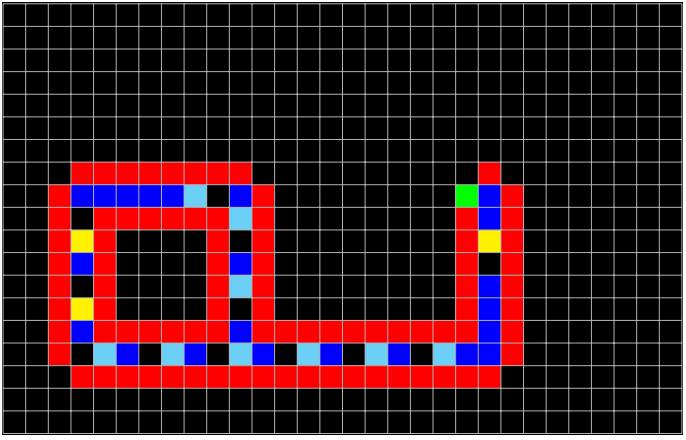
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

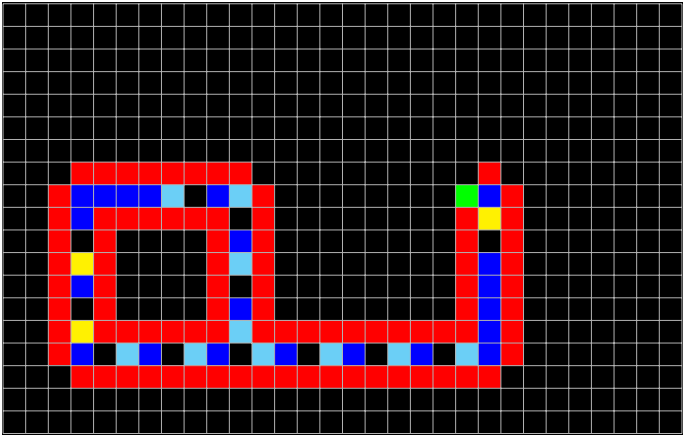
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

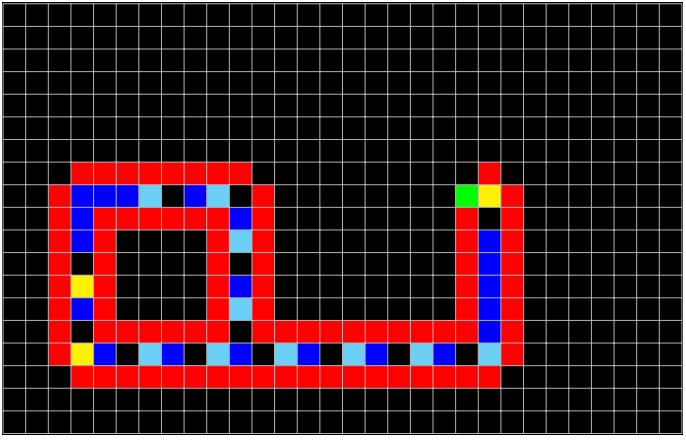
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

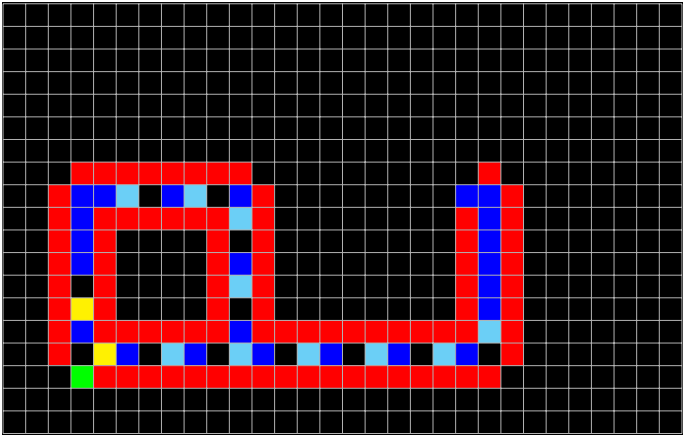
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

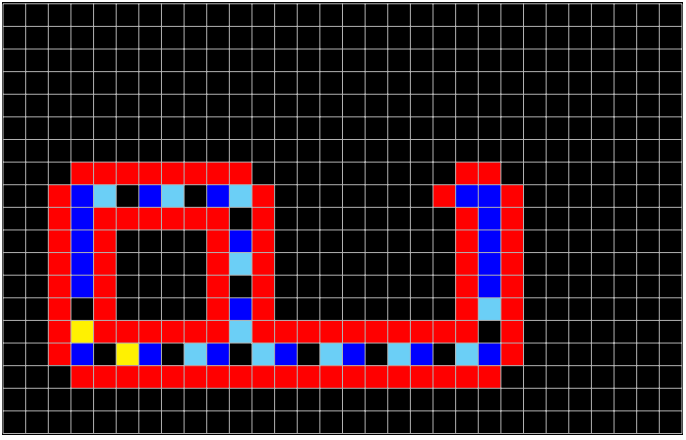
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

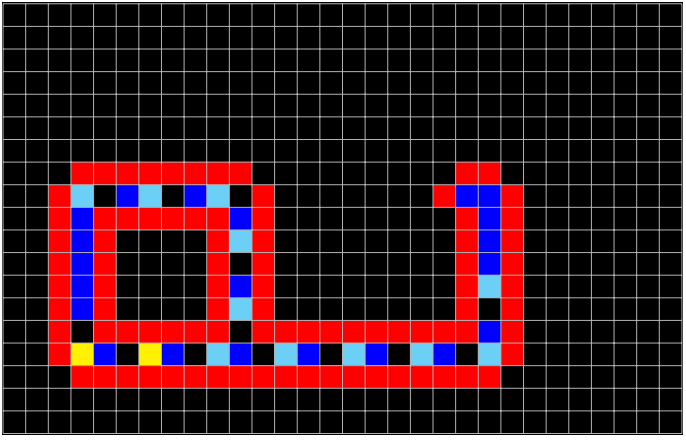
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

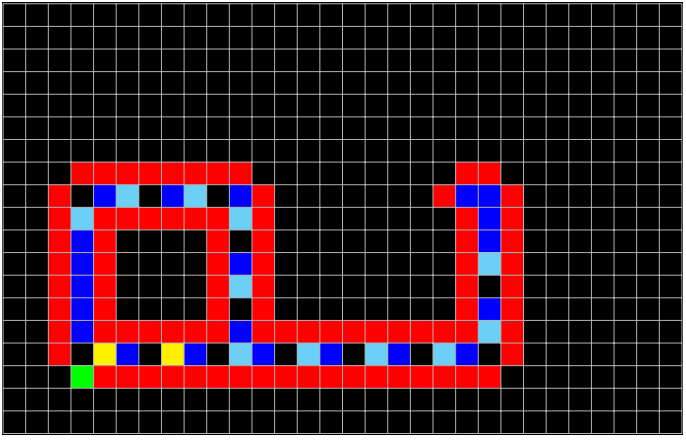
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

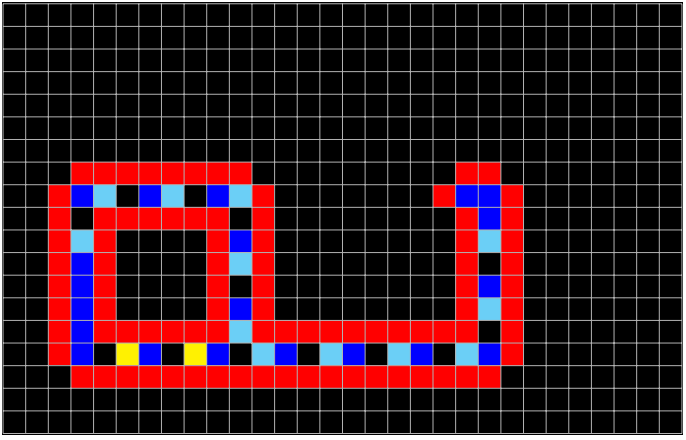
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

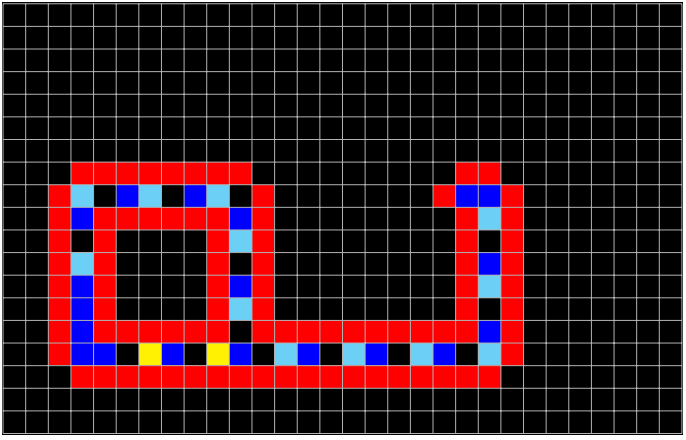
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

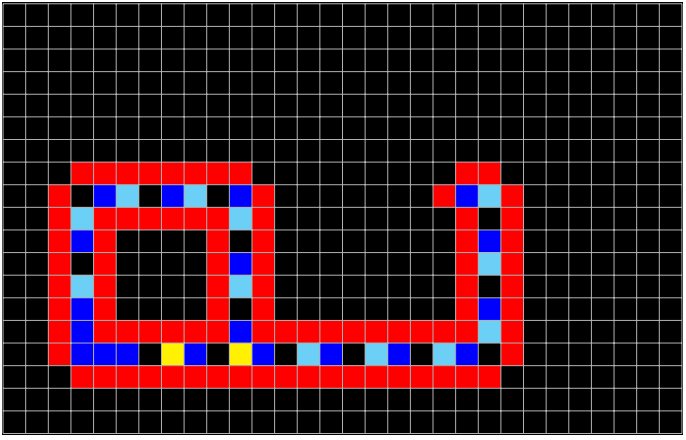
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

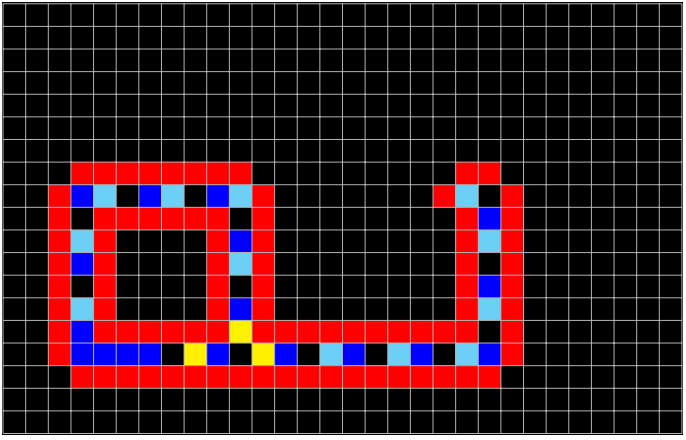
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

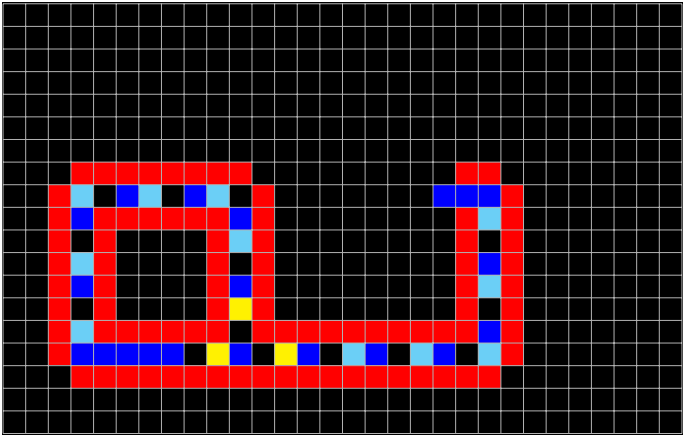
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

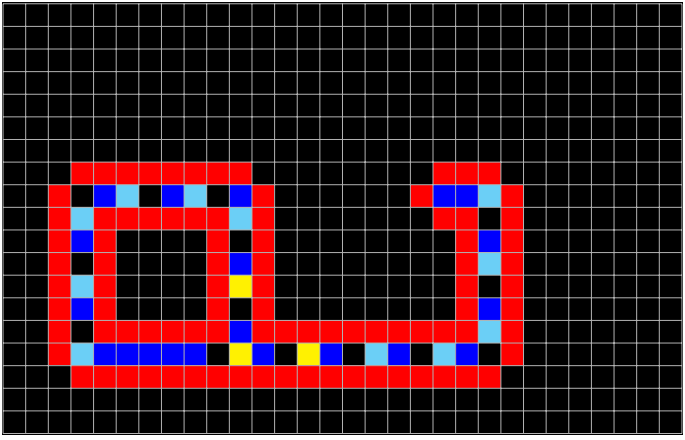
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

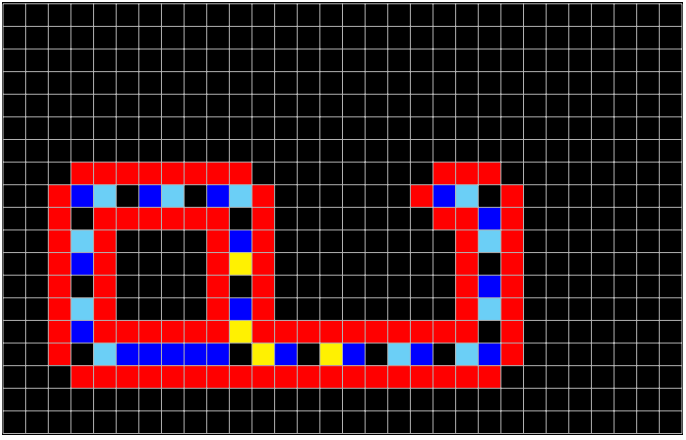
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

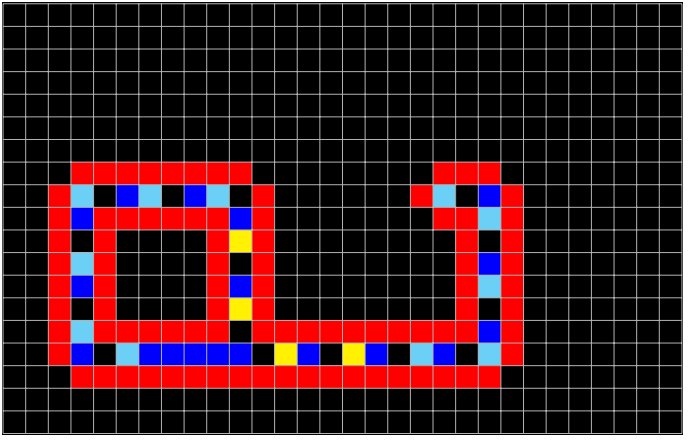
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

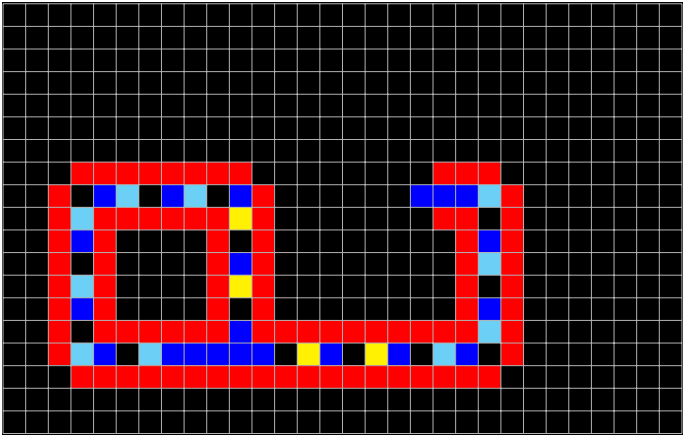
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

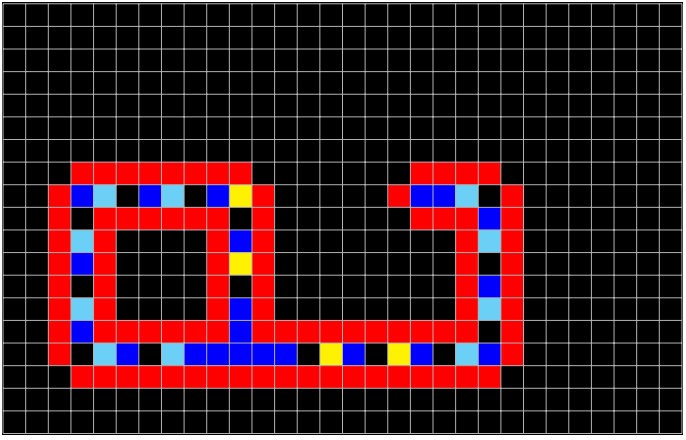
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

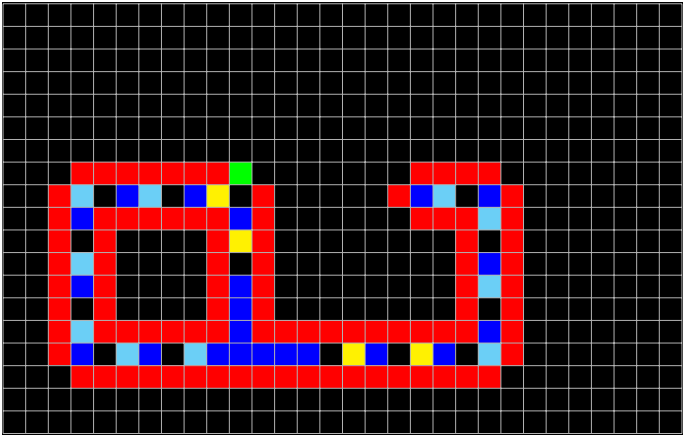
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

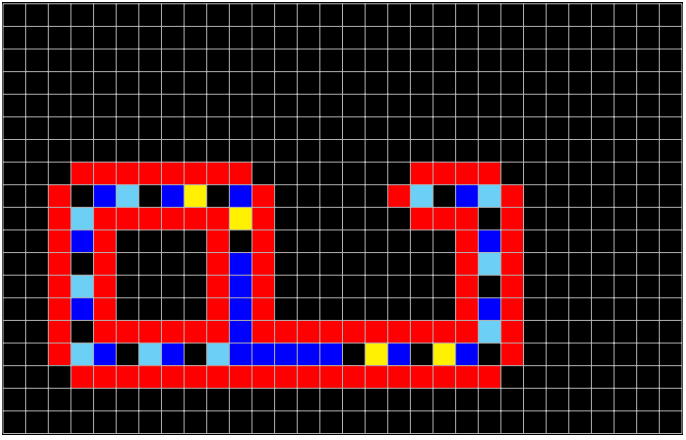
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

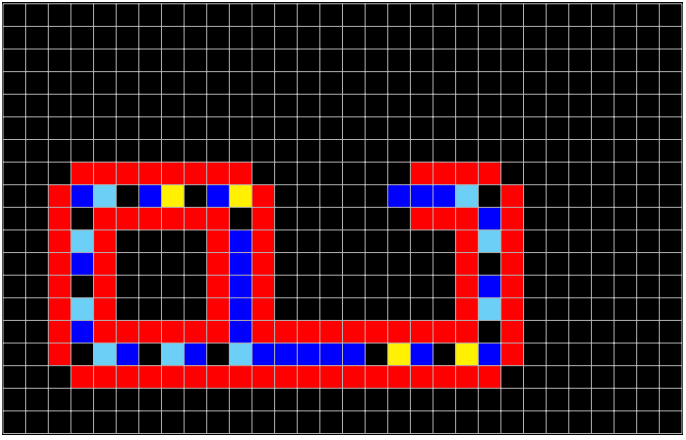
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

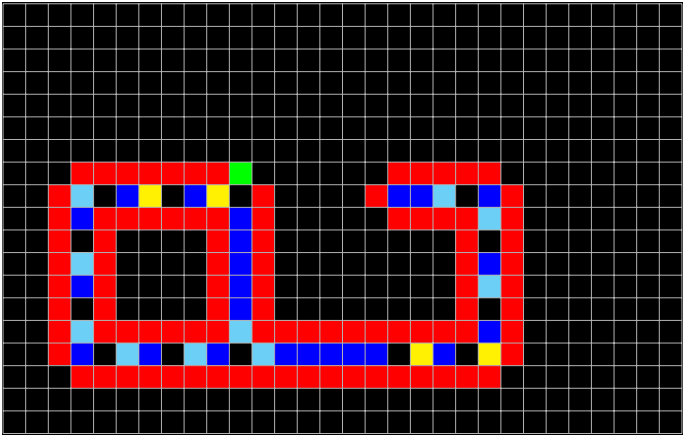
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

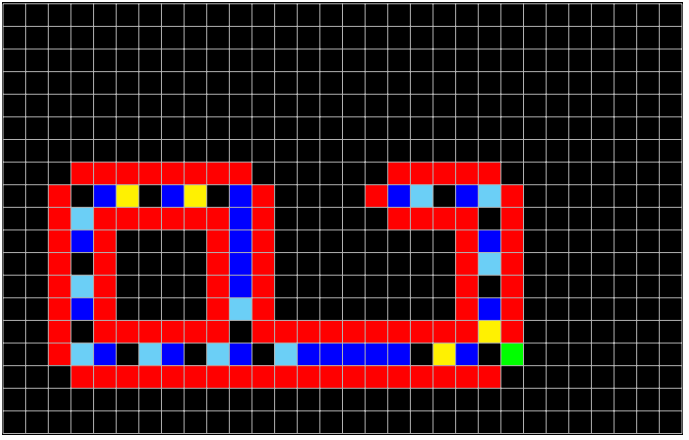
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

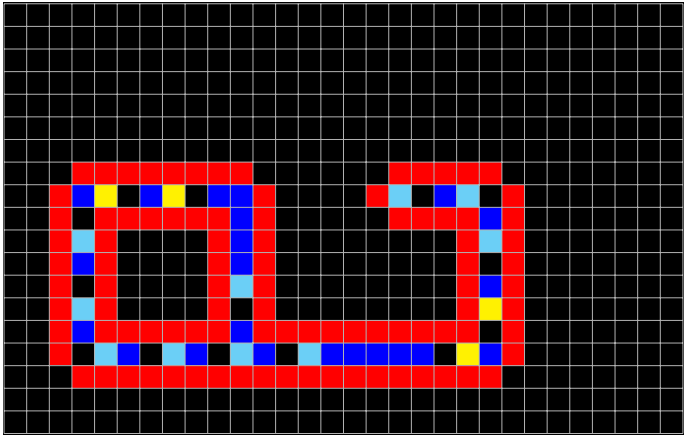
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

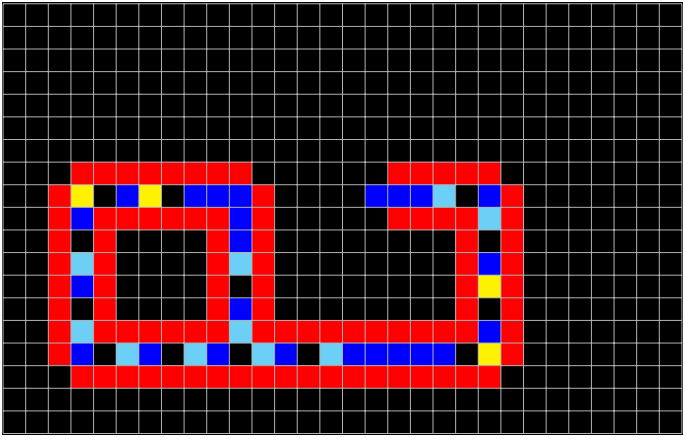
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

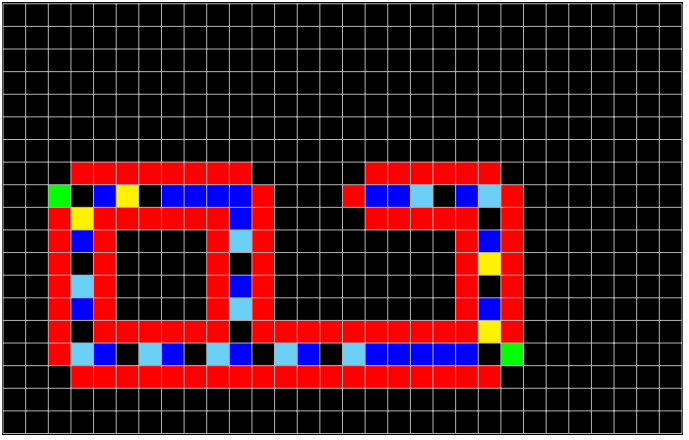
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

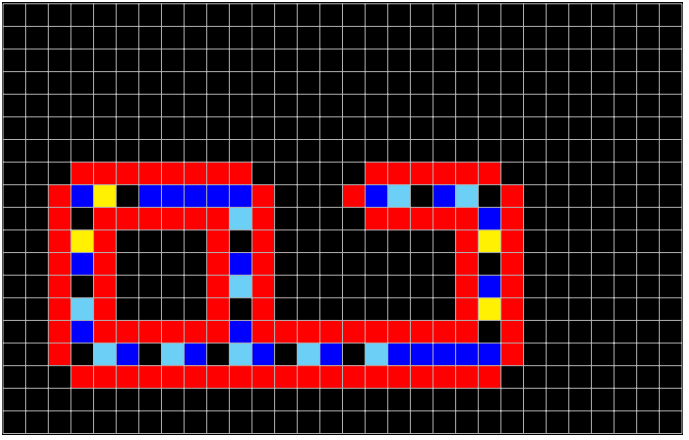
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

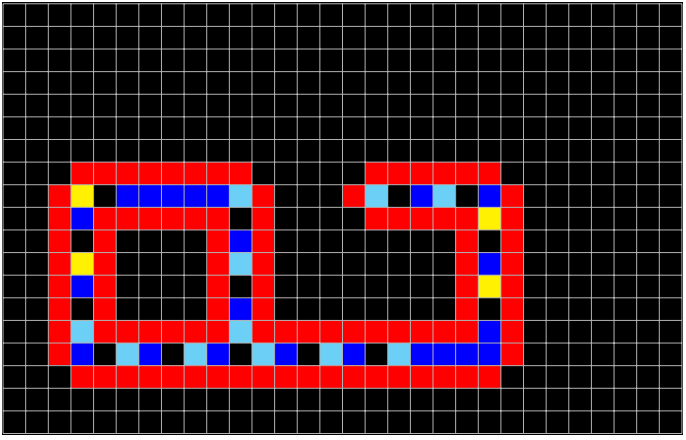
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

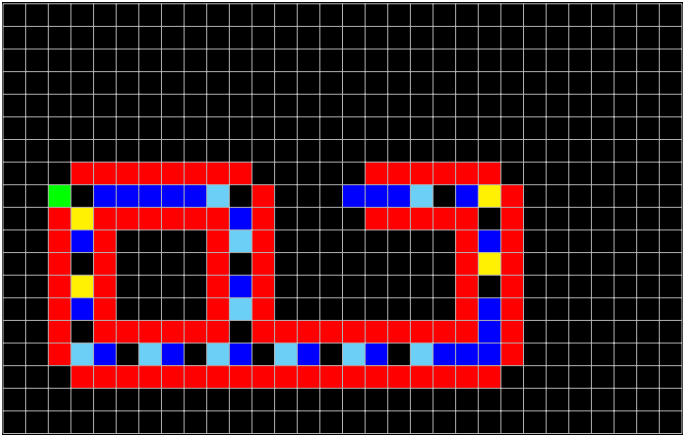
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

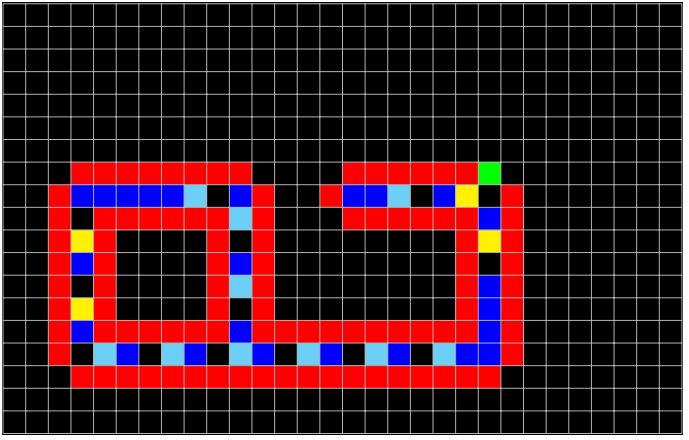
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

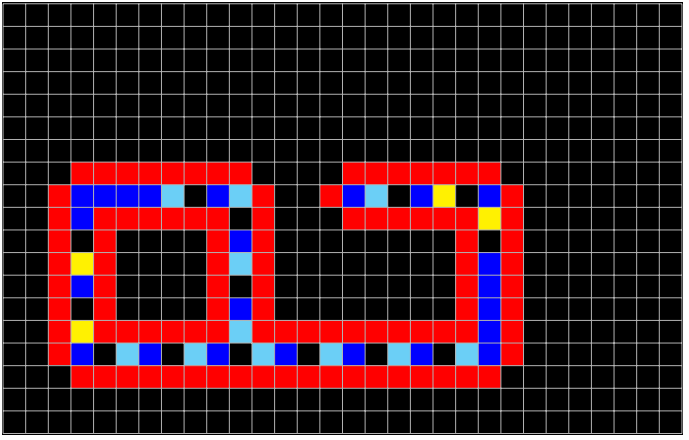
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

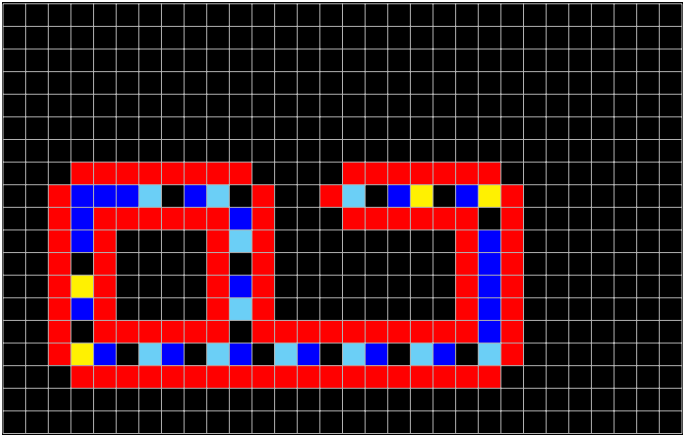
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

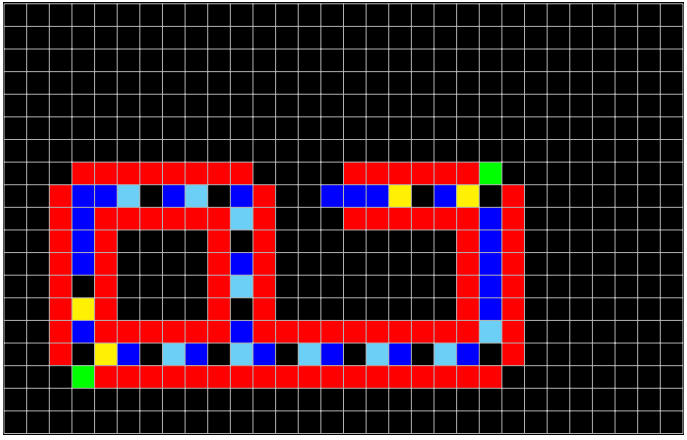
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

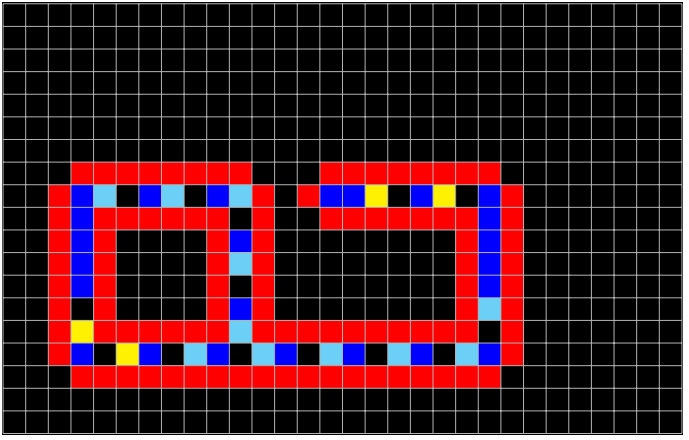
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

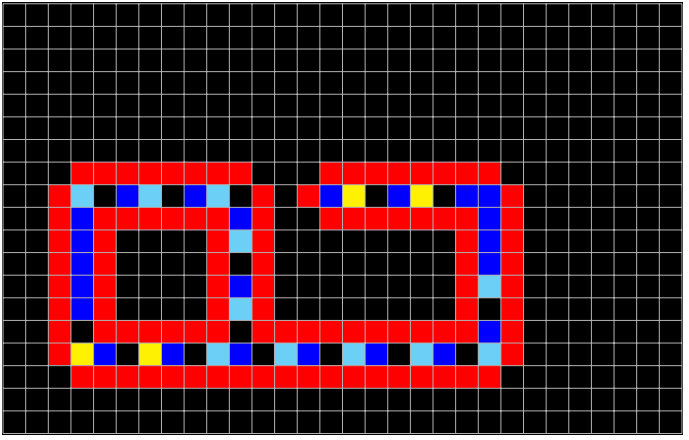
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

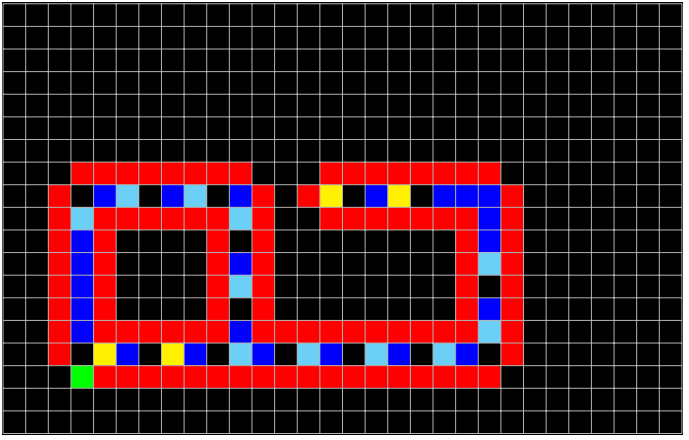
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

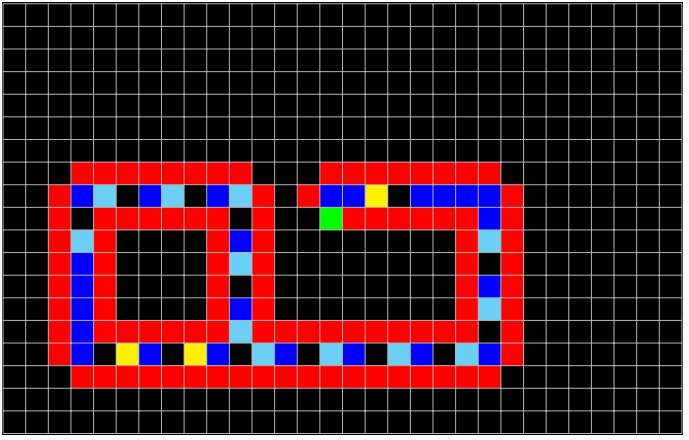
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

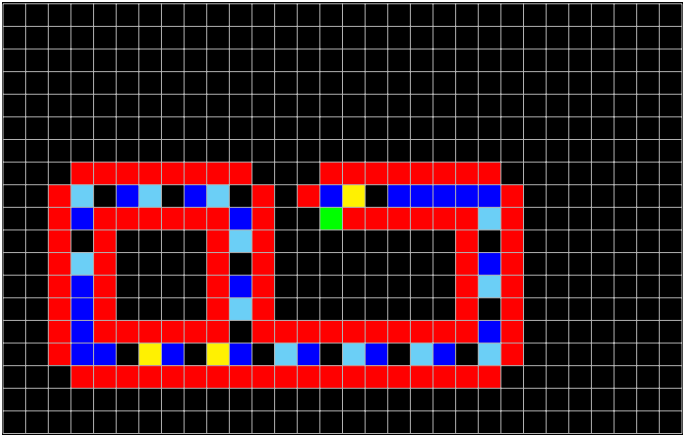
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

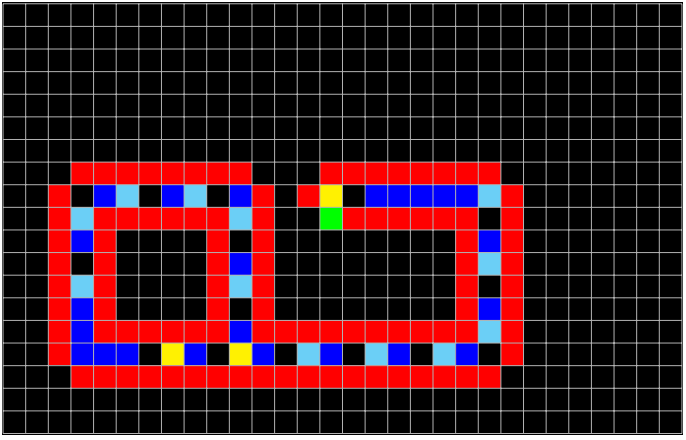
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

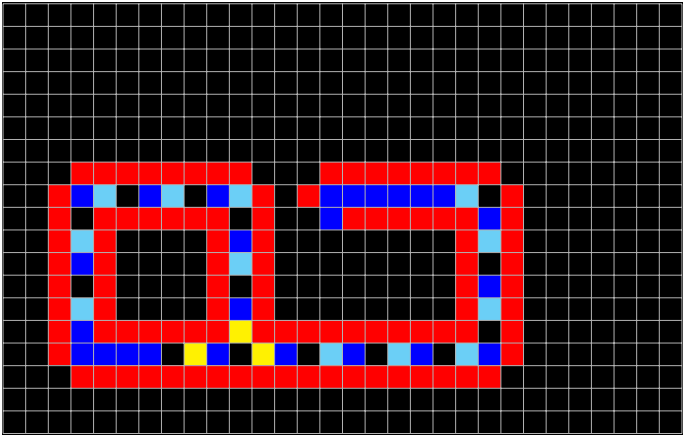
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

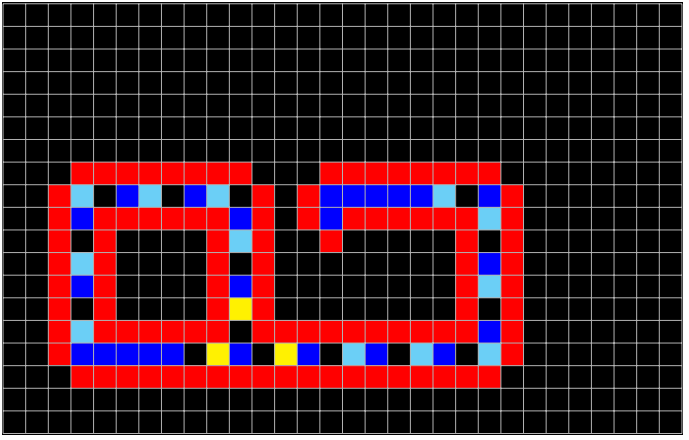
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

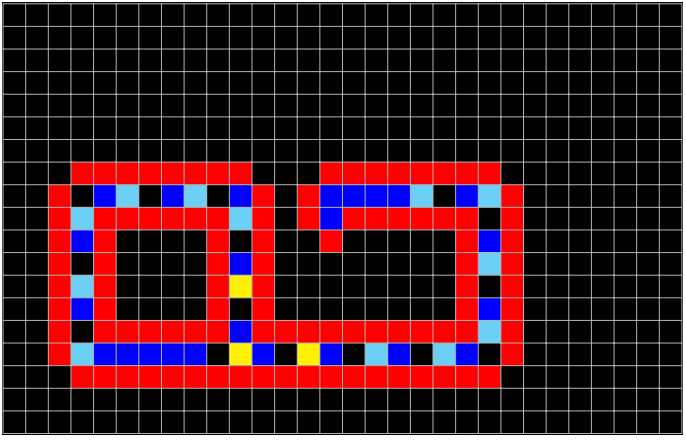
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

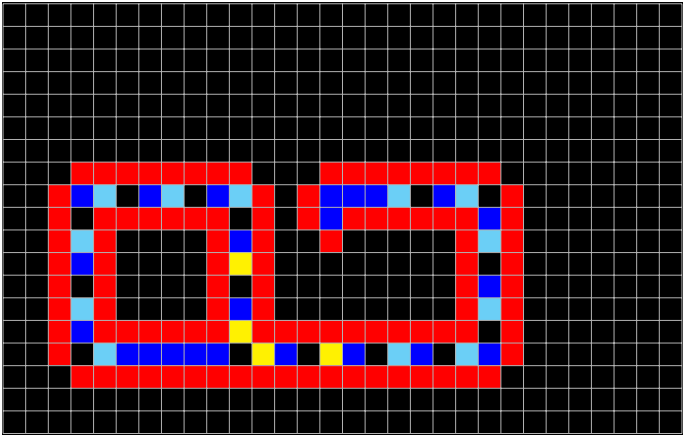
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

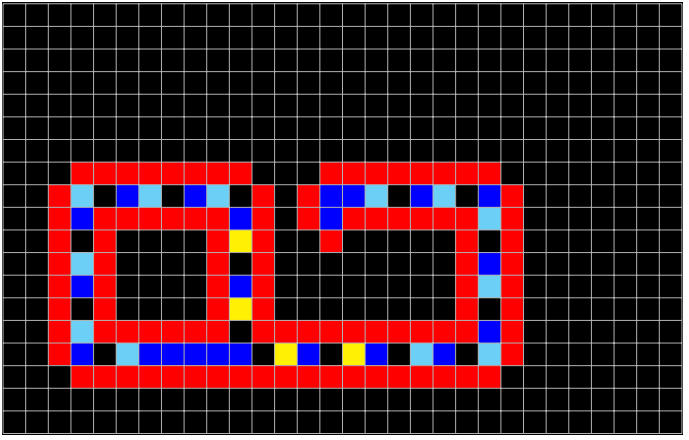
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

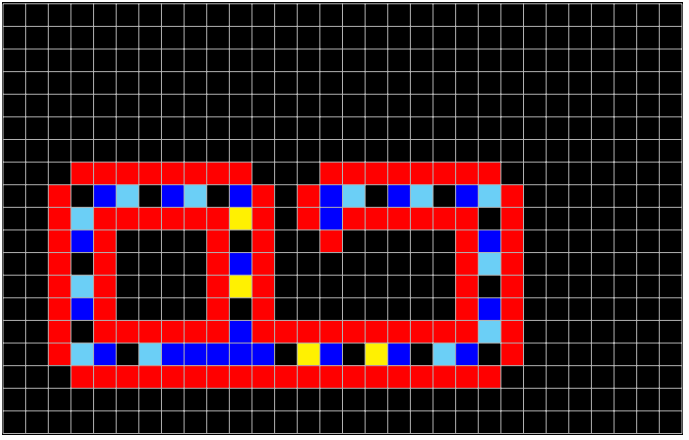
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

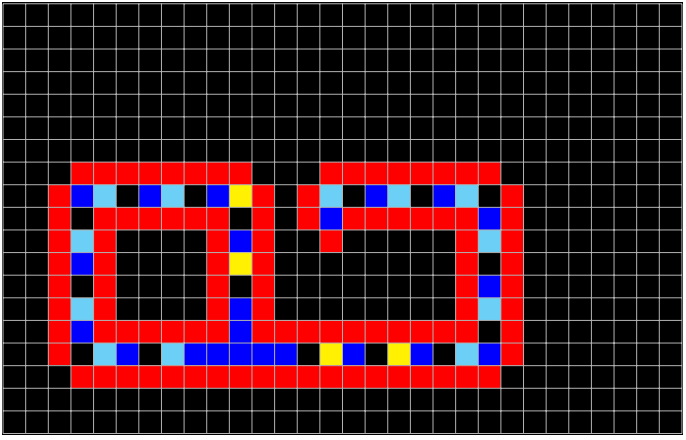
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

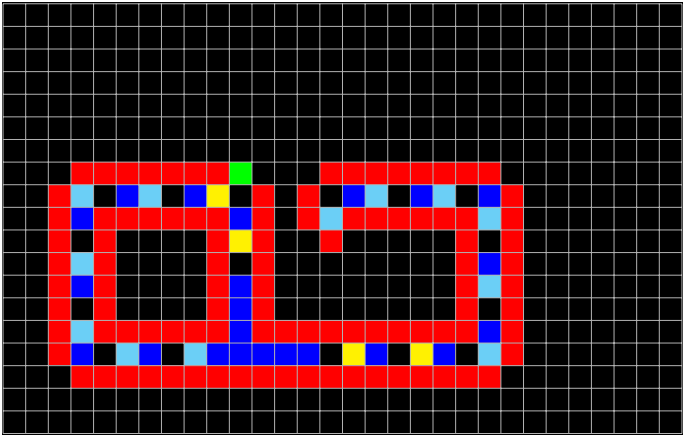
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

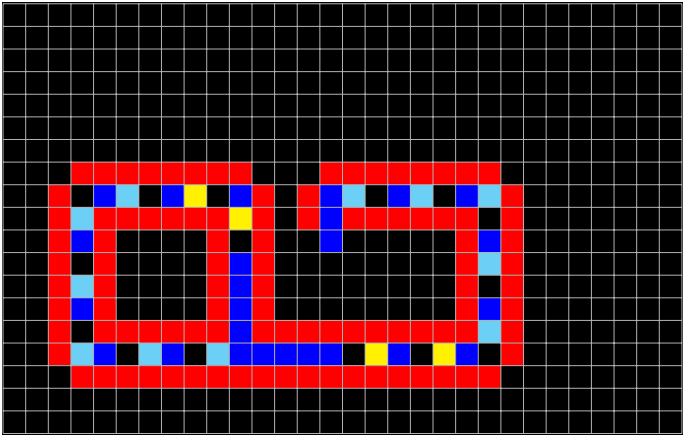
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

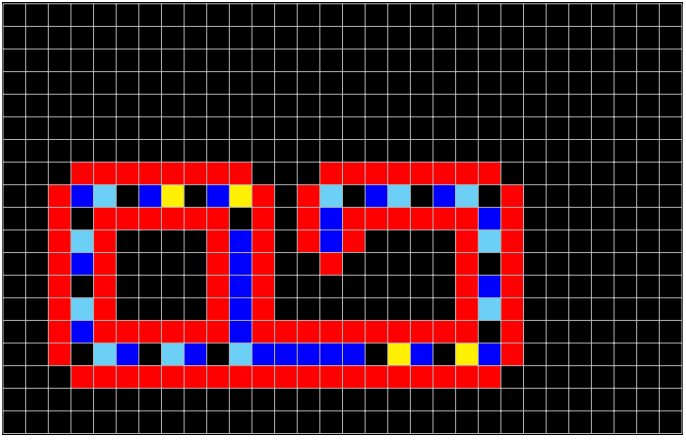
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

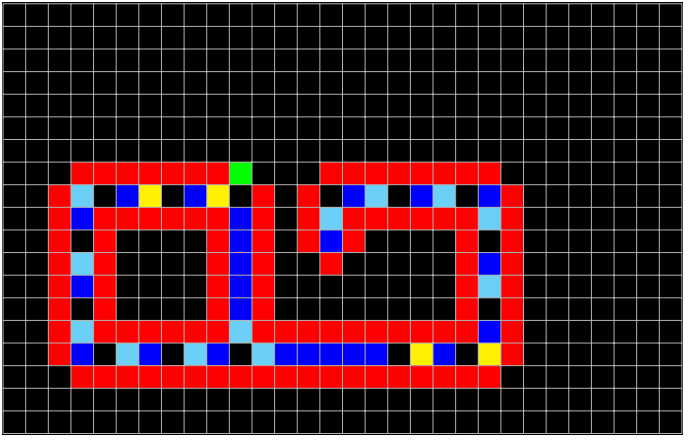
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

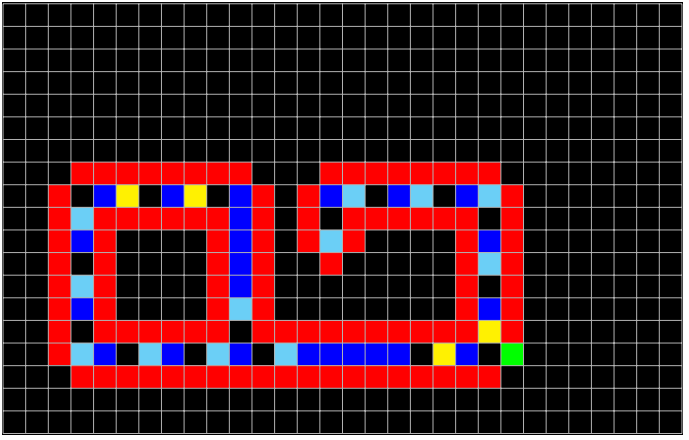
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

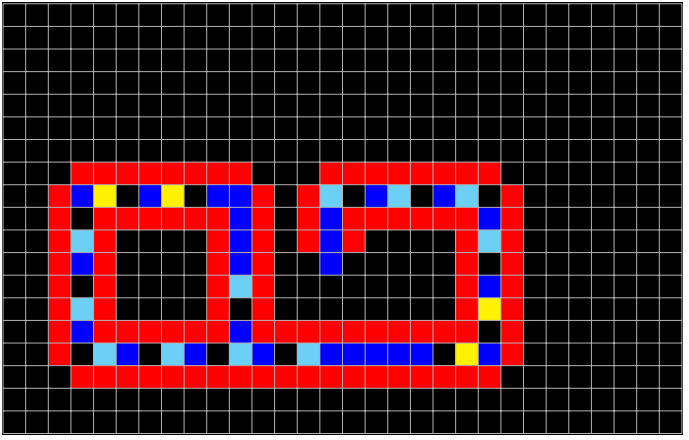
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

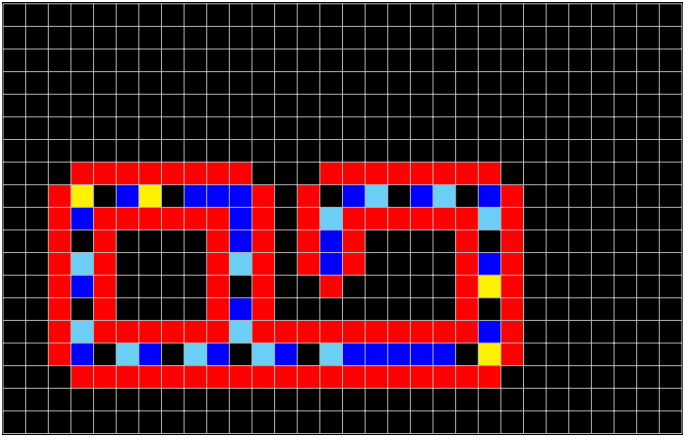
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

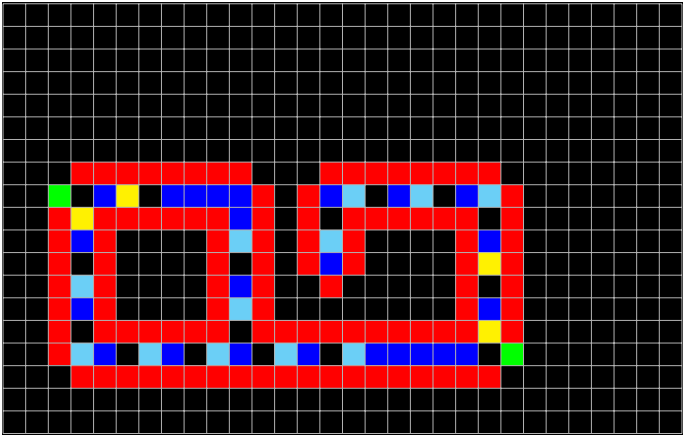
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

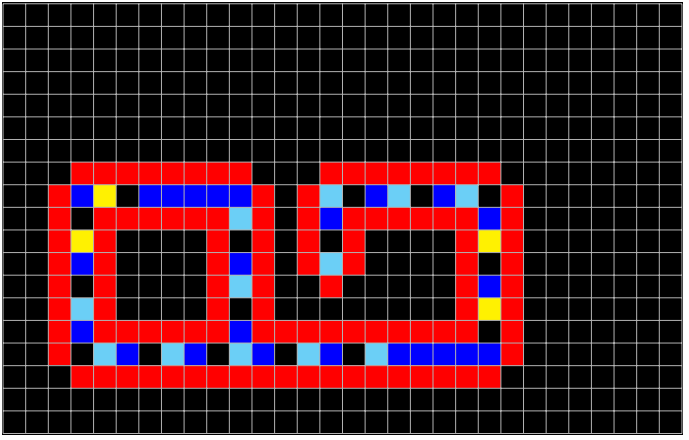
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

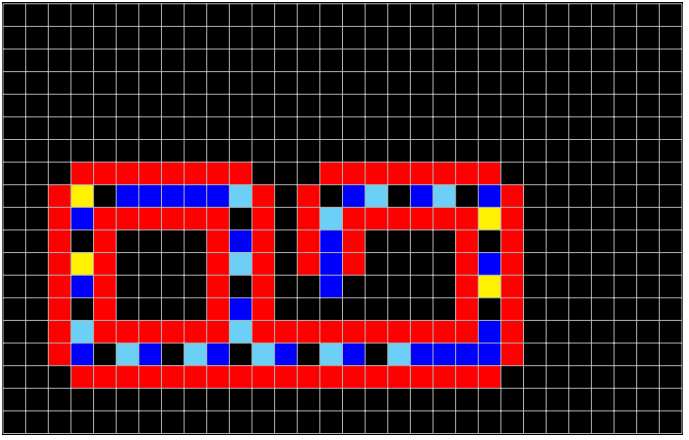
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

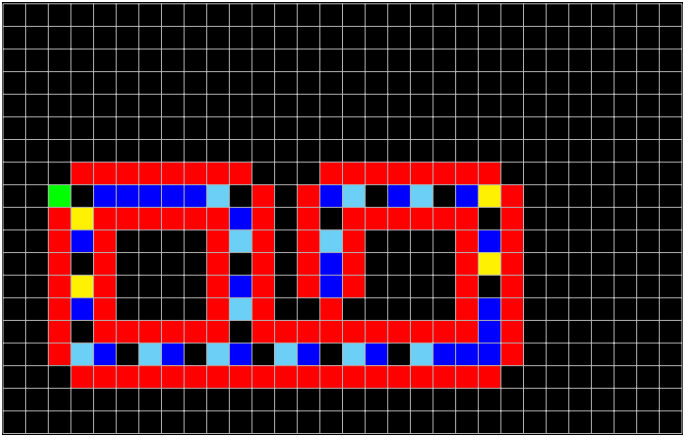
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

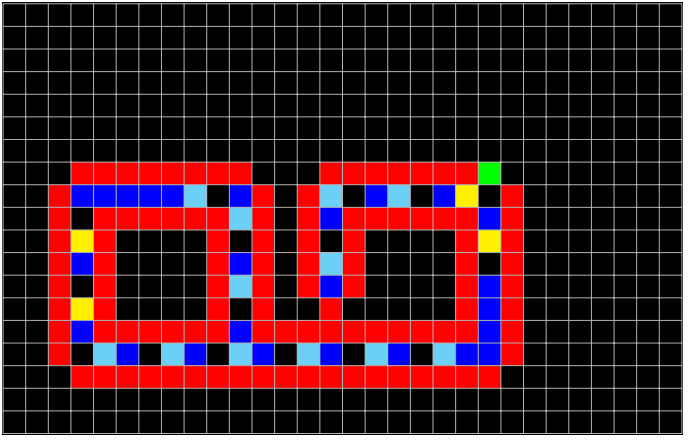
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

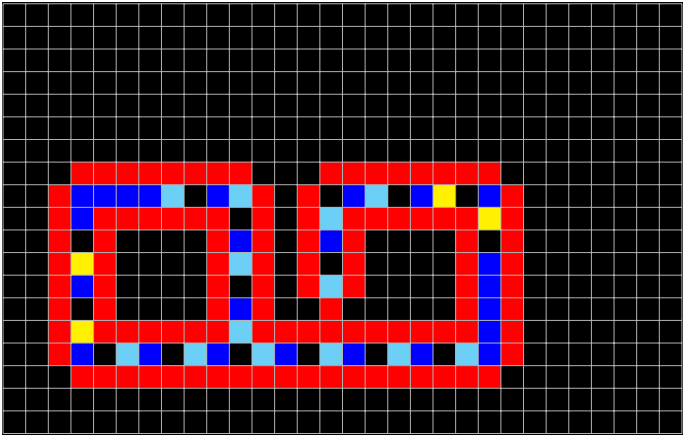
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

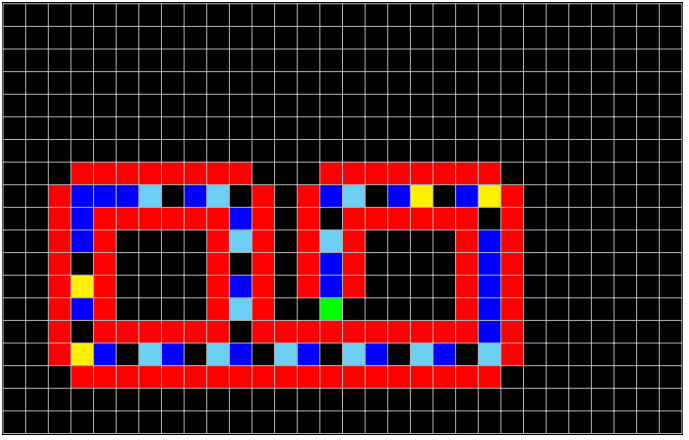
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

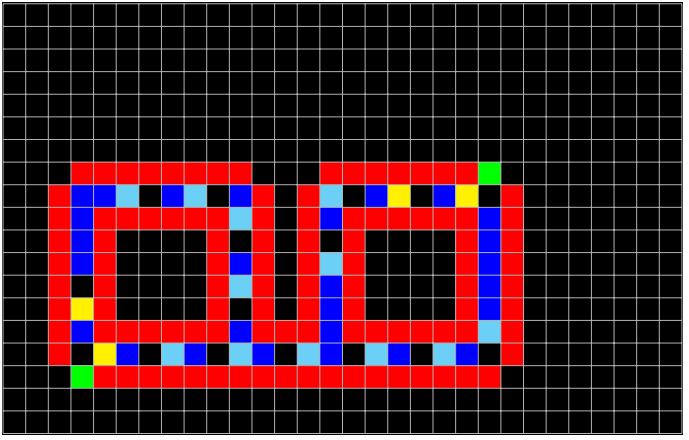
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

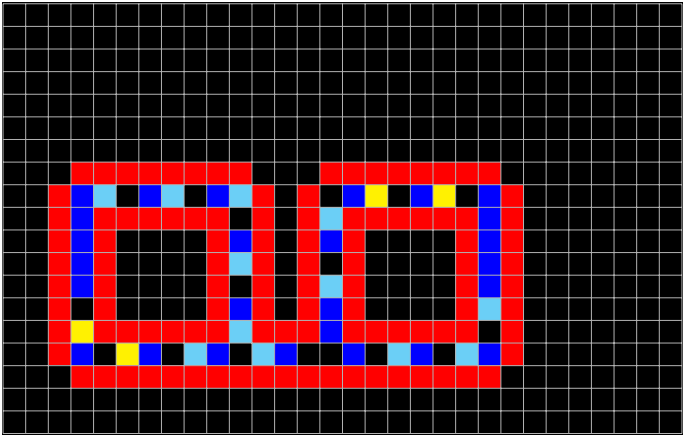
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

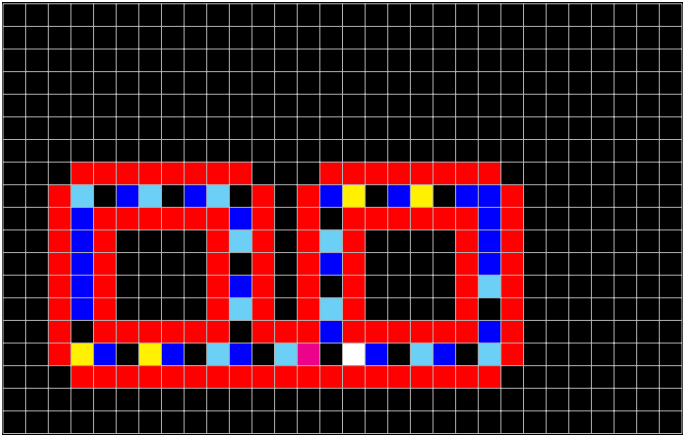
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

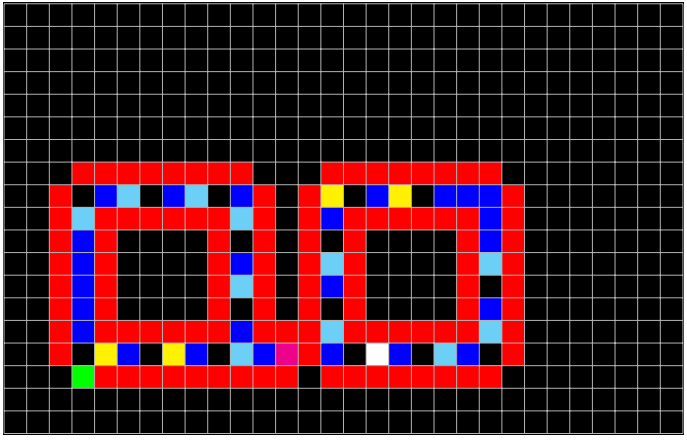
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

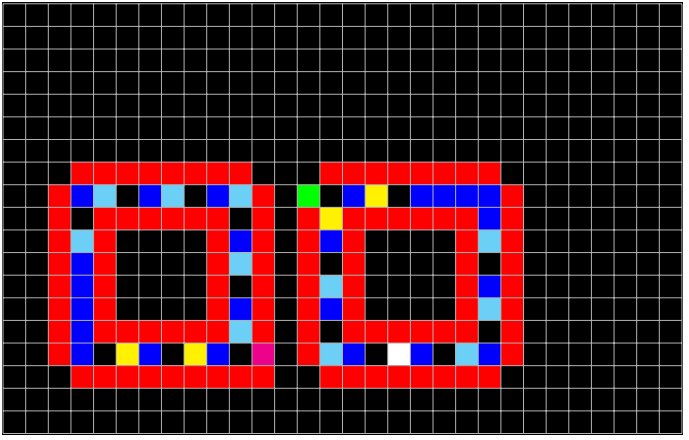
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

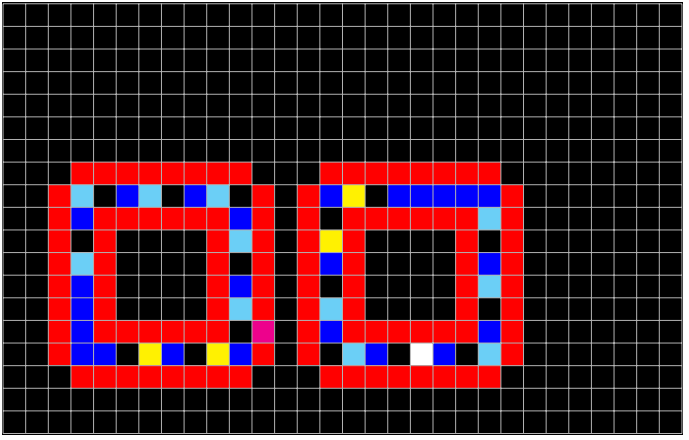
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

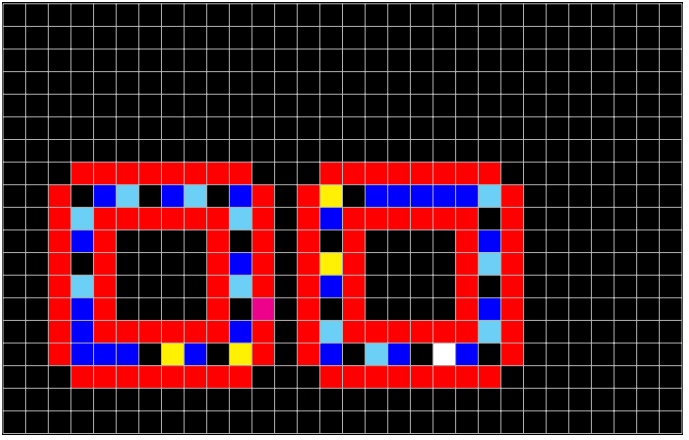
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

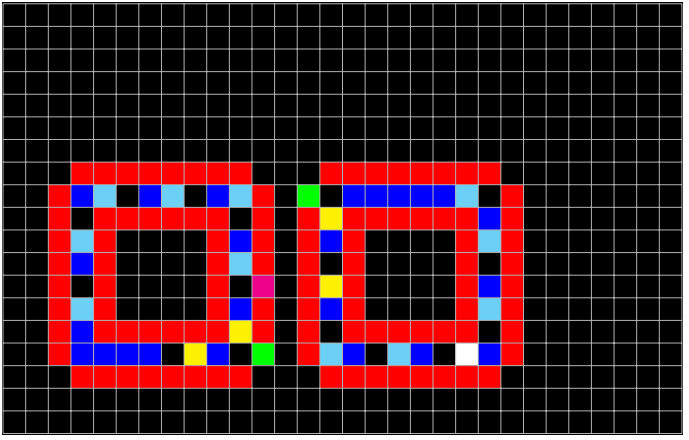
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

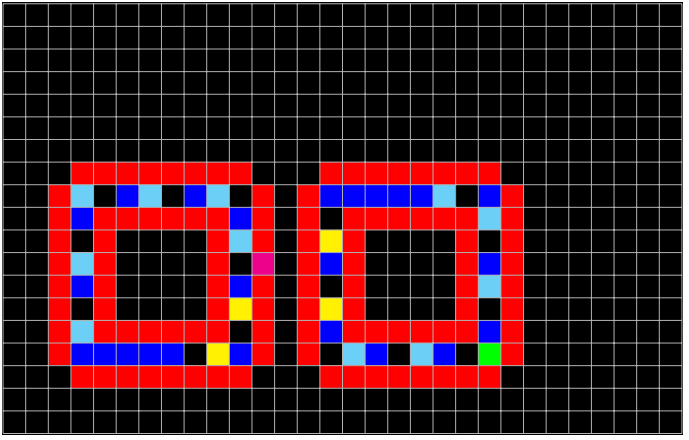
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

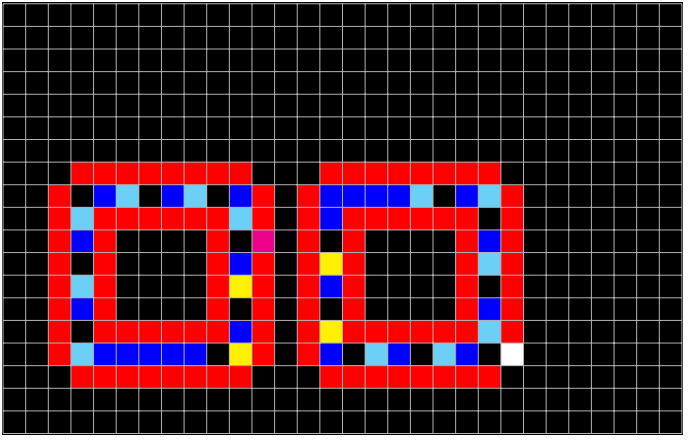
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

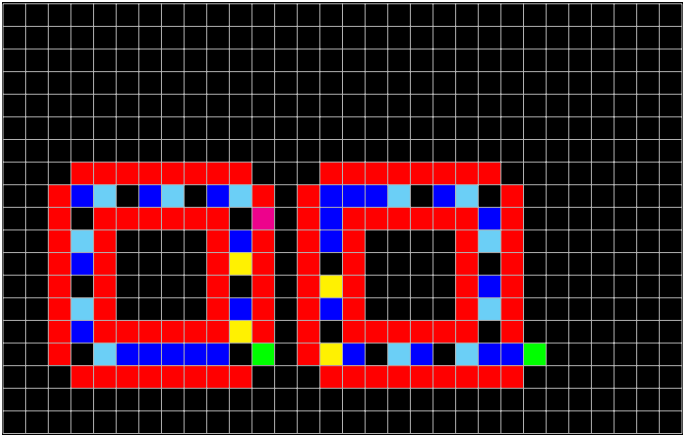
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

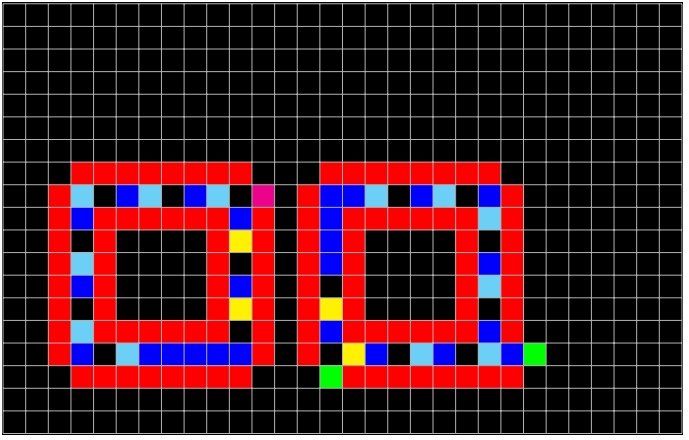
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

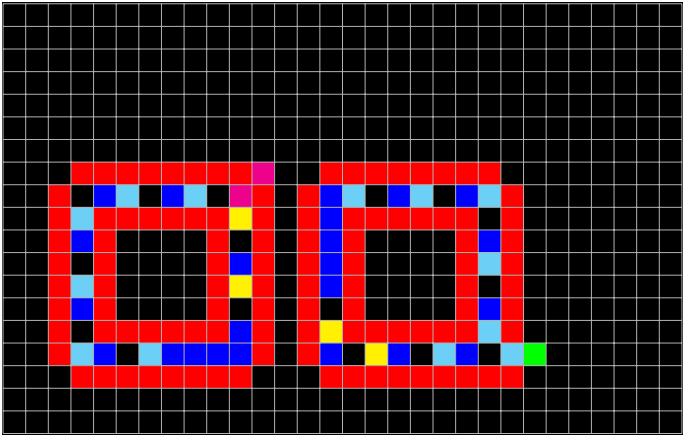
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

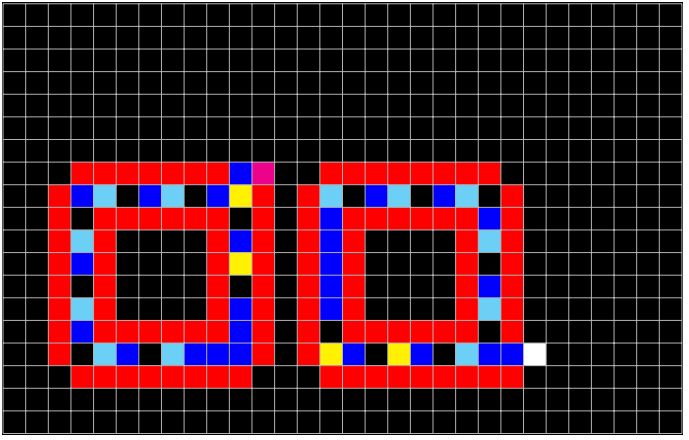
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

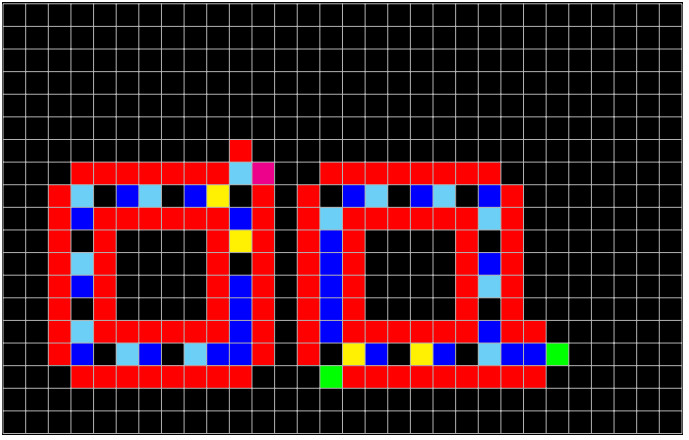
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

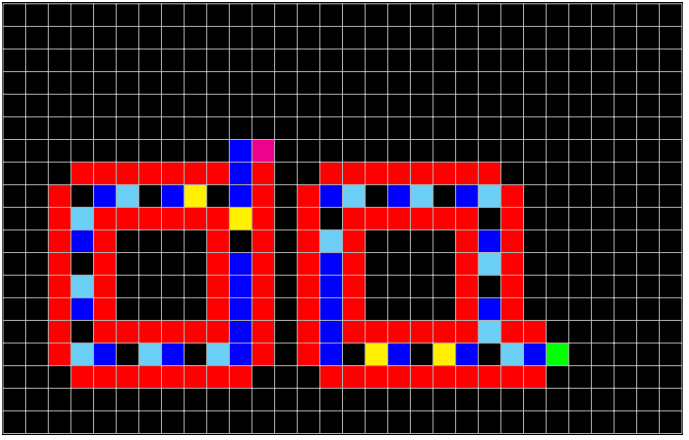
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

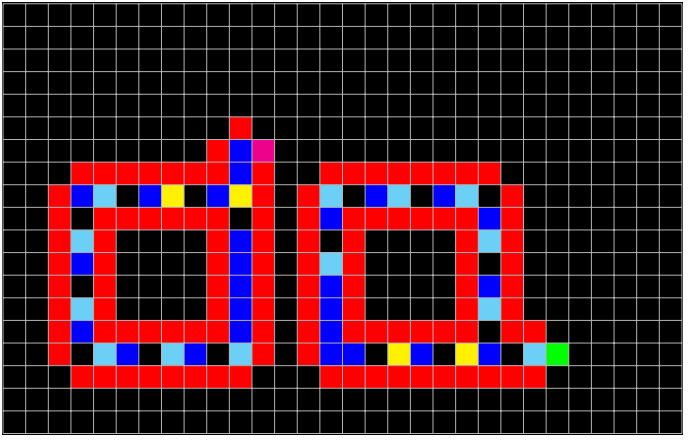
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

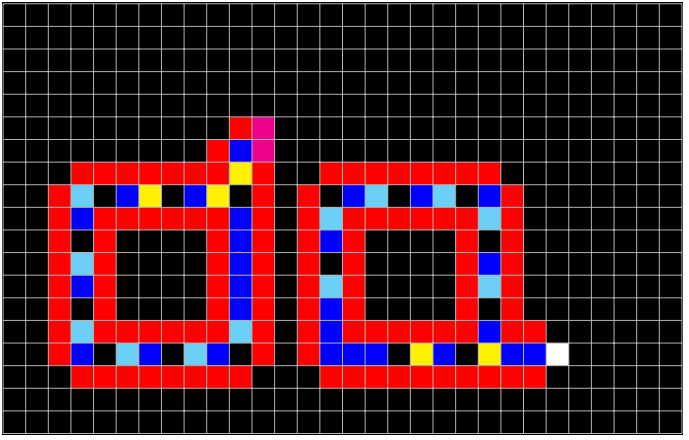
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

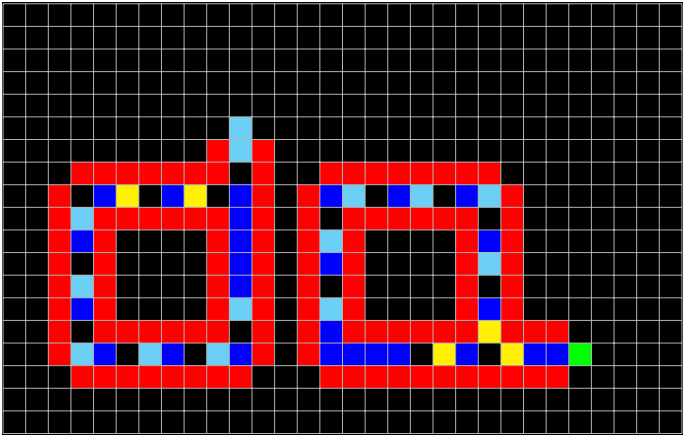
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

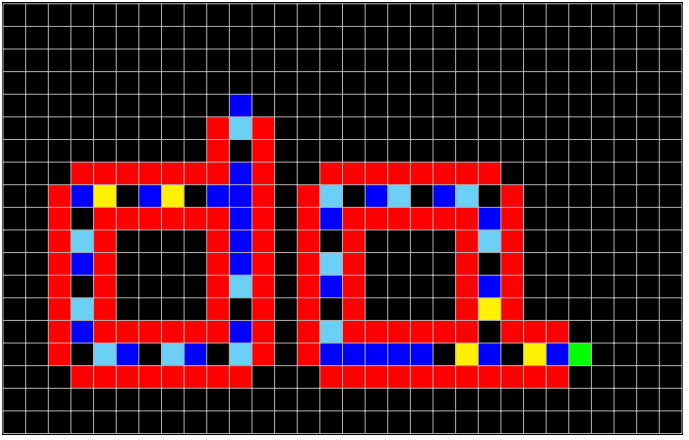
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

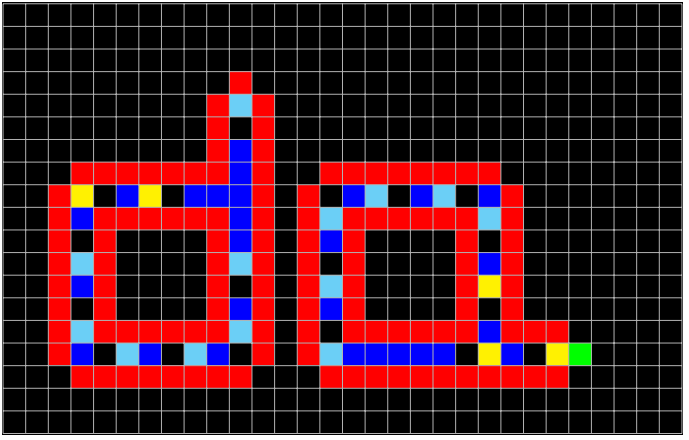
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

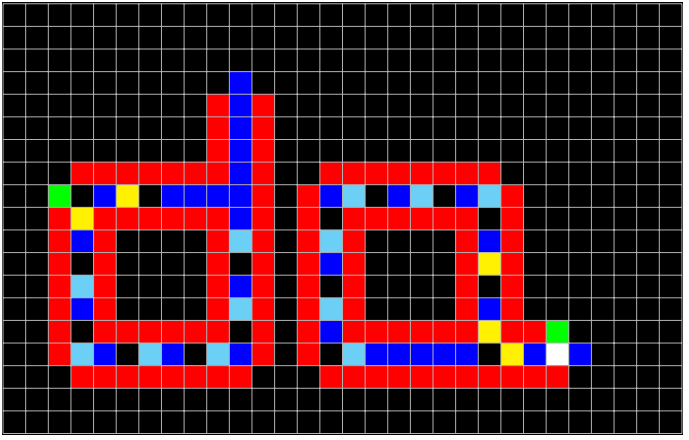
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

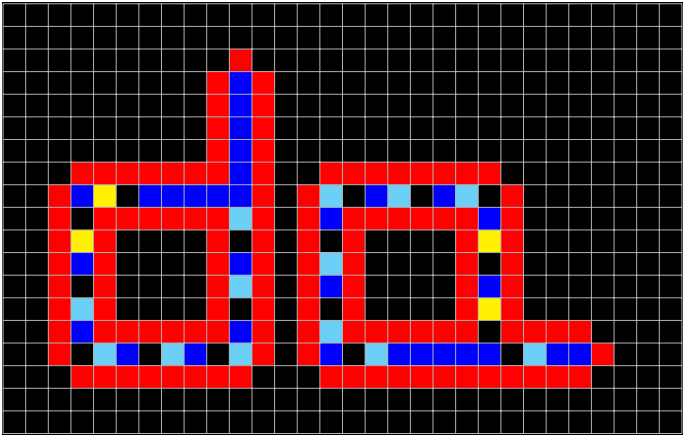
Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

(Langton, 1984)

Boucles autoréplicantes de Langton



Langton modifie un AC de Codd à **8 états** autoréplicant **non universel invariant par rotation** (86 cellules, 151 étapes).

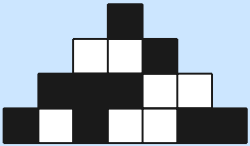
(Langton, 1984)

1. Automates cellulaires

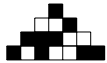
2. Universalités

3. Dynamiques indécidables

4. Conclusion

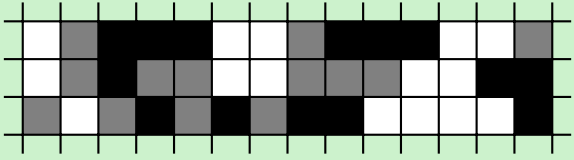


Automates cellulaires



Définition Un **AC** est un triplet (S, N, f) où S est un **ensemble fini d'états**, $N \subseteq_{\text{fini}} \mathbb{Z}^2$ est un **voisinage** fini et $f : S^N \to S$ est la **règle locale de transition** de l'automate cellulaire.

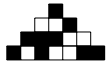
Une **configuration** $c \in S^{\mathbb{Z}^2}$ est un coloriage de $\mathbb{Z}^2$ par S.



La **fonction globale** $F : S^{\mathbb{Z}^2} \to S^{\mathbb{Z}^2}$ applique f uniformément et localement :

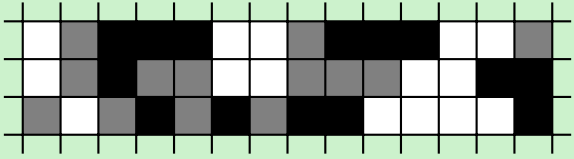
$$\forall c \in S^{\mathbb{Z}^2}, \forall z \in \mathbb{Z}^2, \qquad F(c)(z) = f(c_{|z+N}).$$

Automates cellulaires en dimension 2



Définition Un **AC** est un triplet (S, N, f) où S est un **ensemble fini d'états**, $N \subseteq_{\text{fini}} \mathbb{Z}^2$ est un **voisinage** fini et $f : S^N \to S$ est la **règle locale de transition** de l'automate cellulaire.

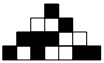
Une **configuration** $c \in S^{\mathbb{Z}^2}$ est un coloriage de $\mathbb{Z}^2$ par S.



La **fonction globale** $F : S^{\mathbb{Z}^2} \to S^{\mathbb{Z}^2}$ applique f uniformément et localement :

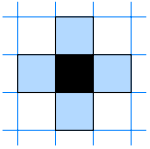
$$\forall c \in S^{\mathbb{Z}^2}, \forall z \in \mathbb{Z}^2, \qquad F(c)(z) = f(c_{|z+N}).$$

Voisinages canoniques



Voisinage de **von Neumann** :

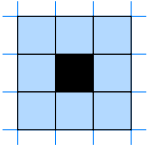
$$N_{\text{vN}} = \{0\} \times \{-1, 0, 1\} \cup \{-1, 0, 1\} \times \{0\}$$



von Neumann

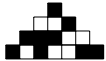
Voisinage de **Moore** :

$$N_{\text{Moore}} = \{-1, 0, 1\} \times \{-1, 0, 1\}$$



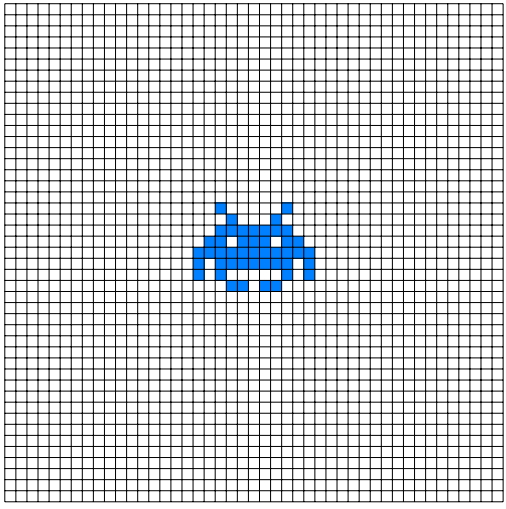
Moore

Exemple : XOR

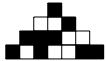


L'AC **XOR**

$$
\begin{aligned}
S &= \{0,1\} \\
N &= \\
f(x_i) &= \sum_i x_i \pmod 2
\end{aligned}
$$



Exemple : XOR

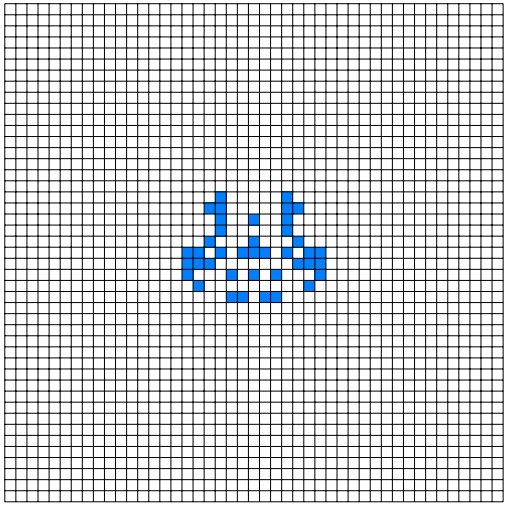


L'AC **XOR**

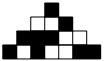
$$S = \{0,1\}$$

$$N = \quad$$ 

$$f(x_i) = \sum_i x_i \pmod 2$$



Exemple : XOR

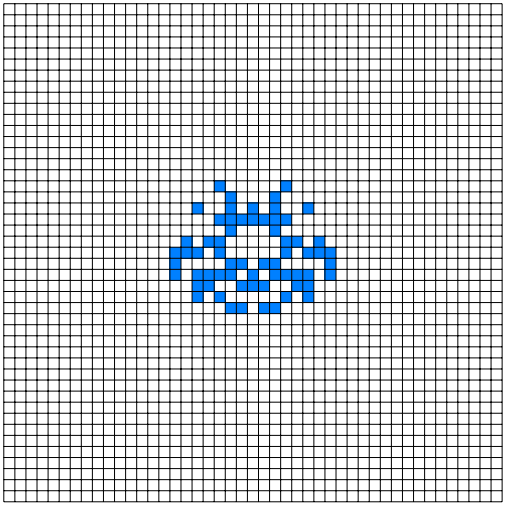


L'AC **XOR**

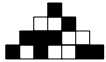
$$S = \{0,1\}$$

$$N =$$ 

$$f(x_i) = \sum_i x_i \pmod 2$$



Exemple : XOR

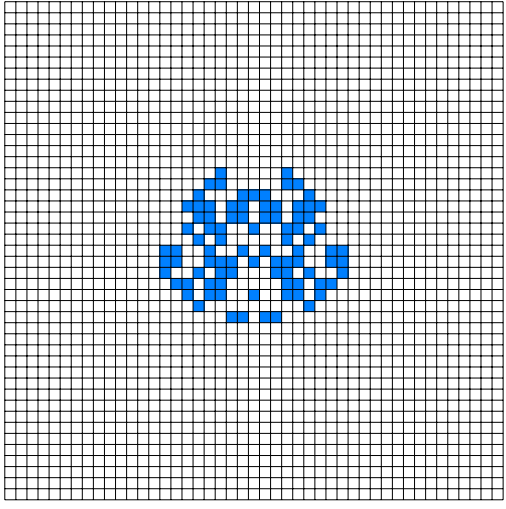


L'AC **XOR**

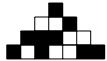
$$S = \{0,1\}$$

$$N =$$ 

$$f(x_i) = \sum_i x_i \pmod 2$$

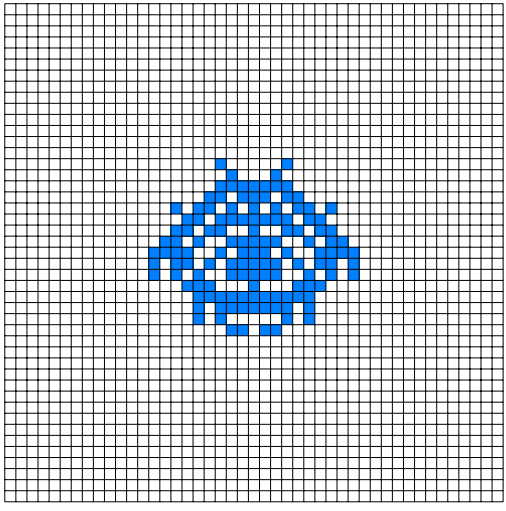


Exemple : XOR

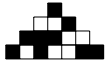


L'AC **XOR**

$$
\begin{aligned}
S &= \{0,1\} \\
N &= \\
f(x_i) &= \sum_i x_i \pmod 2
\end{aligned}
$$

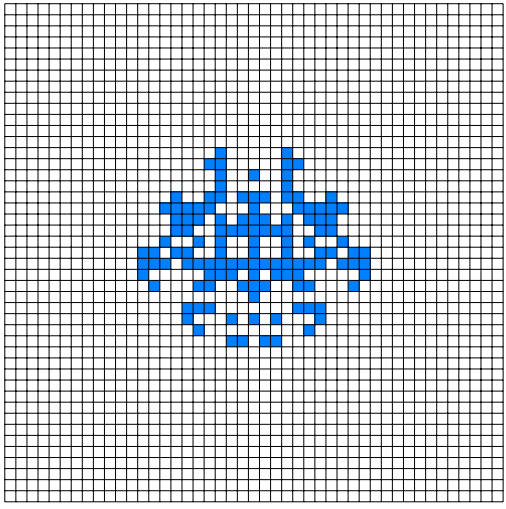


Exemple : XOR

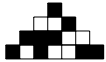


L'AC **XOR**

$$
\begin{aligned}
S &= \{0,1\} \\
N &= \\
f(x_i) &= \sum_i x_i \pmod 2
\end{aligned}
$$

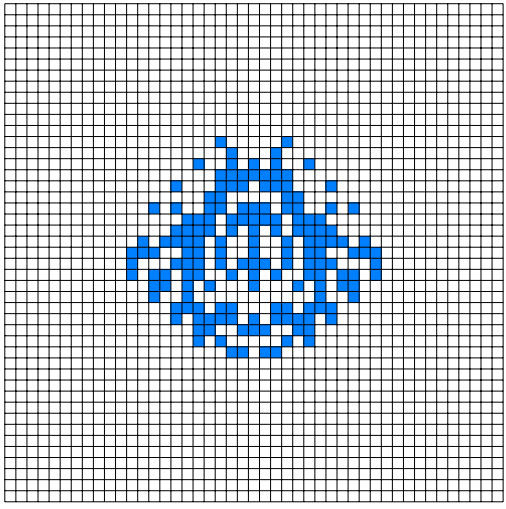


Exemple : XOR

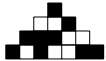


L'AC **XOR**

$$
\begin{aligned}
S &= \{0,1\} \\
N &= \\
f(x_i) &= \sum_i x_i \pmod 2
\end{aligned}
$$

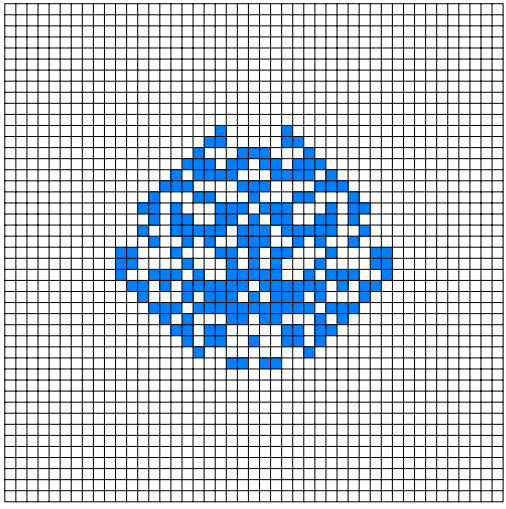


Exemple : XOR

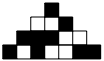


L'AC **XOR**

$$
\begin{aligned}
S &= \{0,1\} \\
N &= \\
f(x_i) &= \sum_i x_i \pmod 2
\end{aligned}
$$



Exemple : XOR

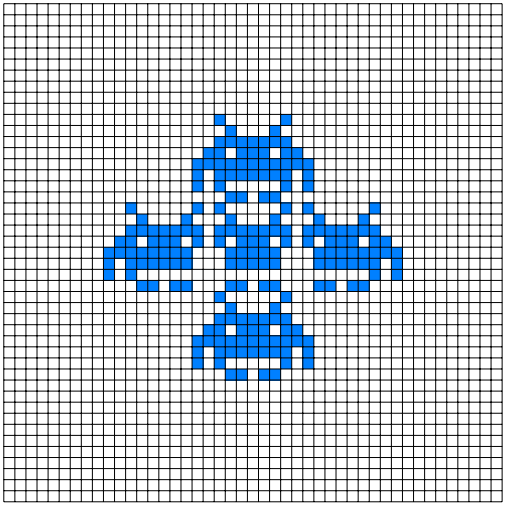


L'AC **XOR**

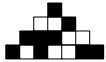
$$S = \{0,1\}$$

$$N =$$ 

$$f(x_i) = \sum_i x_i \pmod 2$$



Exemple : XOR

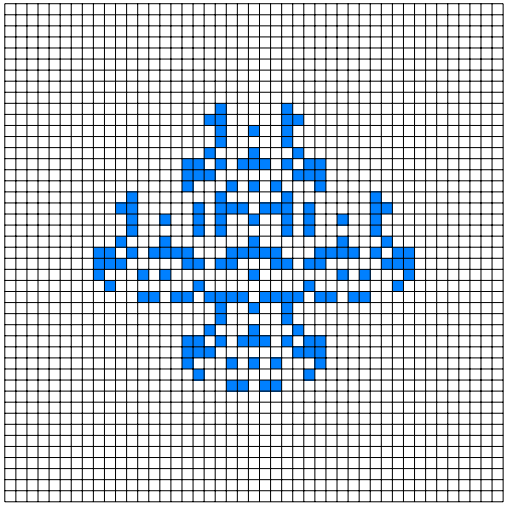


L'AC **XOR**

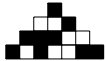
$$S = \{0,1\}$$

$$N =$$ 

$$f(x_i) = \sum_i x_i \pmod 2$$

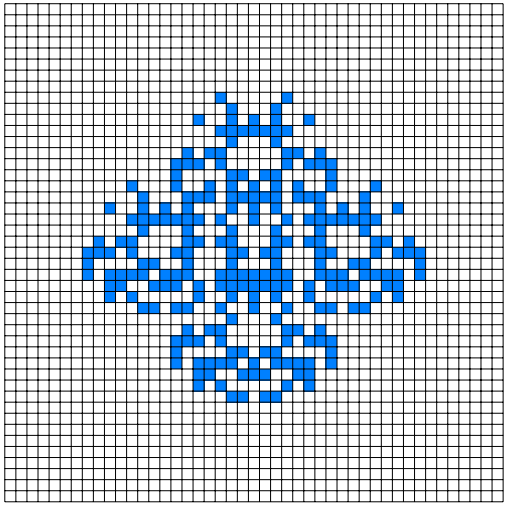


Exemple : XOR

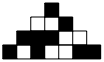


L'AC **XOR**

$$
\begin{aligned}
S &= \{0,1\} \\
N &= \\
f(x_i) &= \sum_i x_i \pmod 2
\end{aligned}
$$



Exemple : XOR

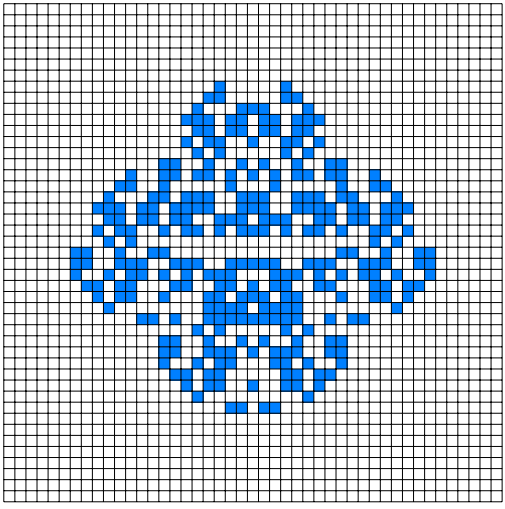


L'AC **XOR**

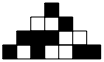
$$S = \{0,1\}$$

$$N = \quad$$ 

$$f(x_i) = \sum_i x_i \pmod 2$$

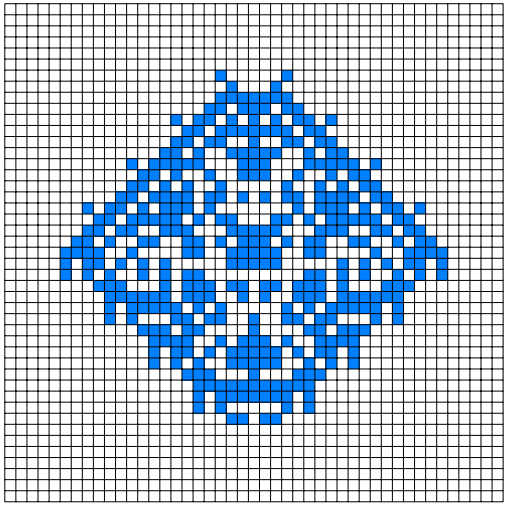


Exemple : XOR

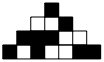


L'AC **XOR**

$$
\begin{aligned}
S &= \{0,1\} \\
N &= \\
f(x_i) &= \sum_i x_i \pmod 2
\end{aligned}
$$

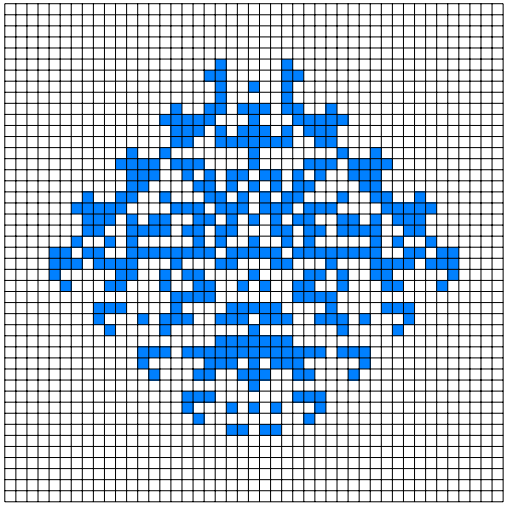


Exemple : XOR

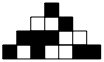


L'AC **XOR**

$$
\begin{aligned}
S &= \{0,1\} \\
N &= \\
f(x_i) &= \sum_i x_i \pmod 2
\end{aligned}
$$

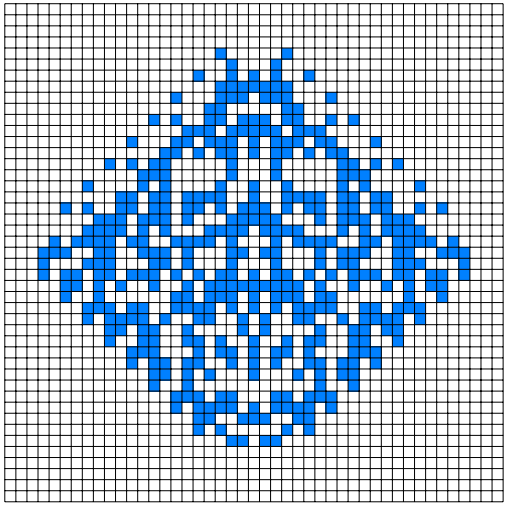


Exemple : XOR

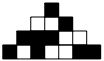


L'AC **XOR**

$$
\begin{aligned}
S &= \{0,1\} \\
N &= \\
f(x_i) &= \sum_i x_i \pmod 2
\end{aligned}
$$



Exemple : XOR

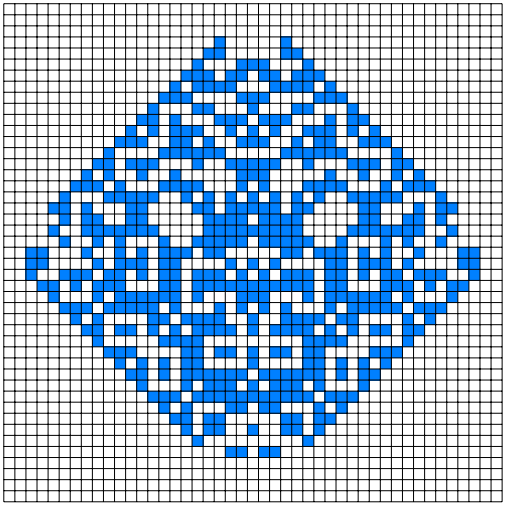


L'AC **XOR**

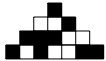
$$S = \{0,1\}$$

$$N =$$ 

$$f(x_i) = \sum_i x_i \pmod 2$$

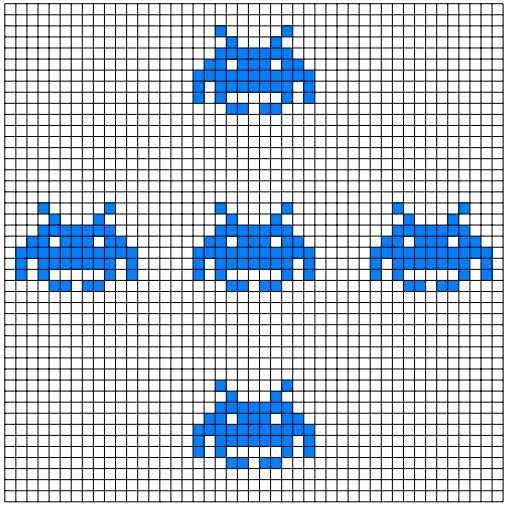


Exemple : XOR

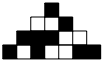


L'AC **XOR**

$$
\begin{aligned}
S &= \{0,1\} \\
N &= \\
f(x_i) &= \sum_i x_i \pmod 2
\end{aligned}
$$



Configurations



L'ensemble des configurations $S^{\mathbb{Z}^2}$ est **indénombrable**. Quel **sous-ensemble dénombrable** raisonnable considérer ?

L'ensemble des **configurations récursives** n'est pas très utile, l'**indécidabilité** est partout (théorème de Rice).

Ensemble des **configurations finies** pour un état quiescent.

Ensemble des **configurations périodiques** qui sont ultimement périodiques dans le temps.

Un compromis : les **configurations ultimement périodiques**.

Diagramme espace-temps (en 1D)

$S = \{0, 1, 2\}$, $r = 1$, $f(x, y, z) = \lfloor 6450288690466/3^{9x+3y+z} \rfloor \pmod 3$

1. Automates cellulaires

2. Universalités

3. Dynamiques indécidables

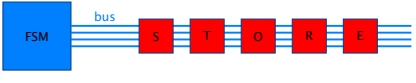
4. Conclusion

Universalité en 2D+

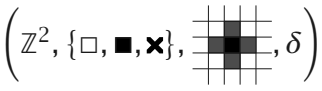
La construction d'AC **universels** apparaît avec les AC comme un outil pour embarquer du calcul. D'abord pour des **AC 2D**.

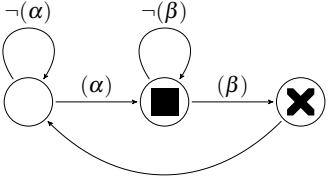
1966	von Neumann	5	29
1968	Codd	5	8
1970	Conway	8	2
1970	Banks	5	2

En 2D, une idée efficace consiste à simuler des **circuits booléens universels** à l'aide de divers ingrédients : **signaux**, **câbles**, **virages**, **fan-outs**, **portes**, **délais**, **horloges**, *etc.*



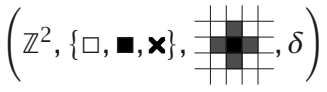
Copper

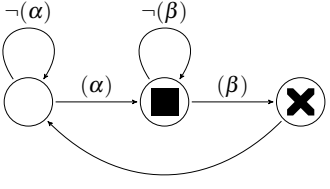




- (α) la paire de voisins nord/sud ou est/ouest est dans l'état ✖ ou dans l'état ■ ;
- (β) au moins deux voisins dans l'état ✖ ou l'état ■ et ou bien exactement un voisin dans l'état ✖ ou bien exactement un voisin dans l'état ■.

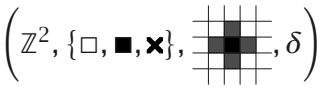
Copper

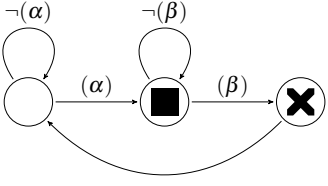




- (α) la paire de voisins nord/sud ou est/ouest est dans l'état ✖ ou dans l'état ■ ;
- (β) au moins deux voisins dans l'état ✖ ou l'état ■ et ou bien exactement un voisin dans l'état ✖ ou bien exactement un voisin dans l'état ■.

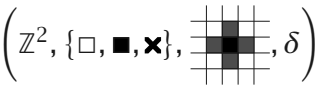
Copper

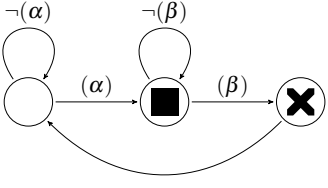




- (α) la paire de voisins nord/sud ou est/ouest est dans l'état × ou dans l'état ■ ;
- (β) au moins deux voisins dans l'état × ou l'état ■ et ou bien exactement un voisin dans l'état × ou bien exactement un voisin dans l'état ■.

Copper

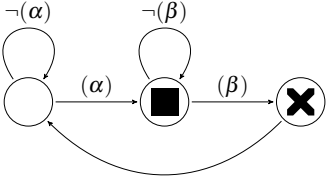




- (α) la paire de voisins nord/sud ou est/ouest est dans l'état × ou dans l'état ■ ;
- (β) au moins deux voisins dans l'état × ou l'état ■ et ou bien exactement un voisin dans l'état × ou bien exactement un voisin dans l'état ■.

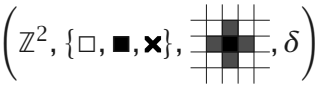
Copper

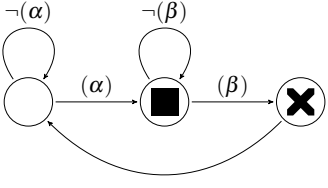
$$\left(\mathbb{Z}^2, \{\square, \blacksquare, \times\}, \; , \delta\right)$$



- (α) la paire de voisins nord/sud ou est/ouest est dans l'état × ou dans l'état ■ ;
- (β) au moins deux voisins dans l'état × ou l'état ■ et ou bien exactement un voisin dans l'état × ou bien exactement un voisin dans l'état ■.

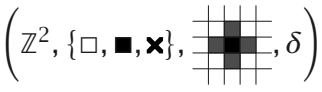
Copper

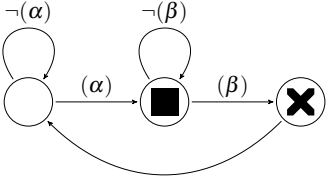




- (α) la paire de voisins nord/sud ou est/ouest est dans l'état × ou dans l'état ■ ;
- (β) au moins deux voisins dans l'état × ou l'état ■ et ou bien exactement un voisin dans l'état × ou bien exactement un voisin dans l'état ■.

Copper

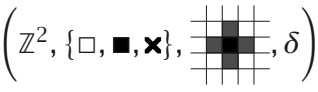


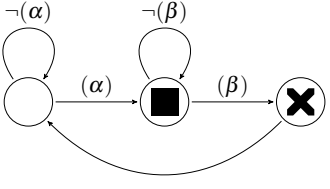


(α) la paire de voisins nord/sud ou est/ouest est dans l'état ✖ ou dans l'état ■ ;

(β) au moins deux voisins dans l'état ✖ ou l'état ■ et ou bien exactement un voisin dans l'état ✖ ou bien exactement un voisin dans l'état ■.

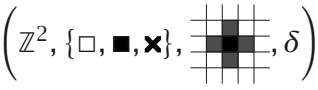
Copper

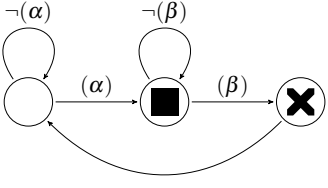




- (α) la paire de voisins nord/sud ou est/ouest est dans l'état ✖ ou dans l'état ■ ;
- (β) au moins deux voisins dans l'état ✖ ou l'état ■ et ou bien exactement un voisin dans l'état ✖ ou bien exactement un voisin dans l'état ■.

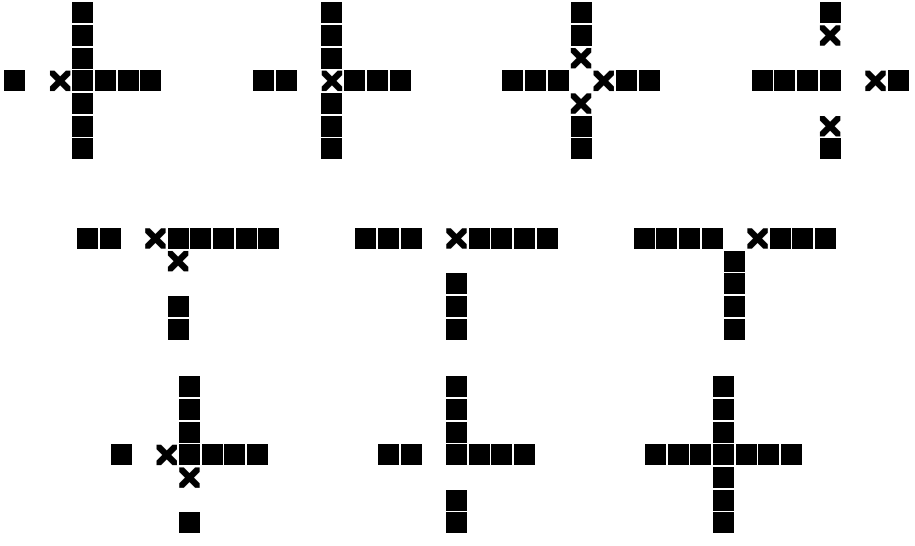
Copper



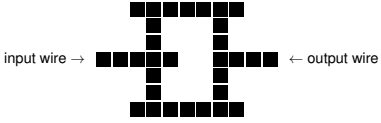


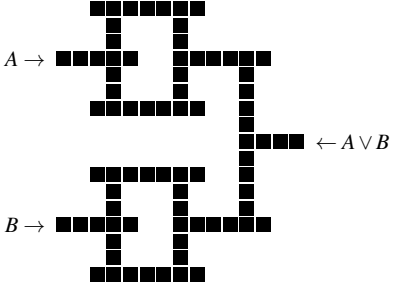
- (α) la paire de voisins nord/sud ou est/ouest est dans l'état ✖ ou dans l'état ■ ;
- (β) au moins deux voisins dans l'état ✖ ou l'état ■ et ou bien exactement un voisin dans l'état ✖ ou bien exactement un voisin dans l'état ■.

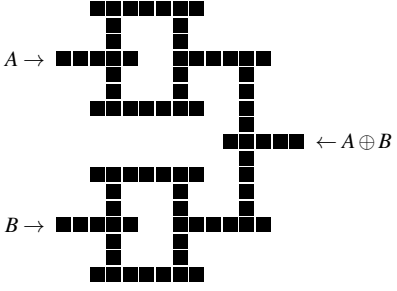
Copper : intersections



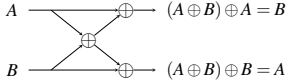
Copper : portes

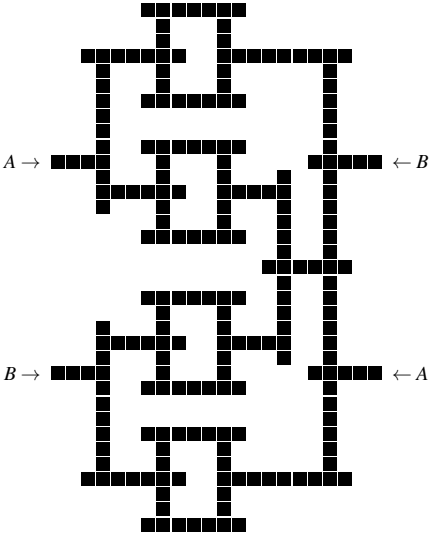




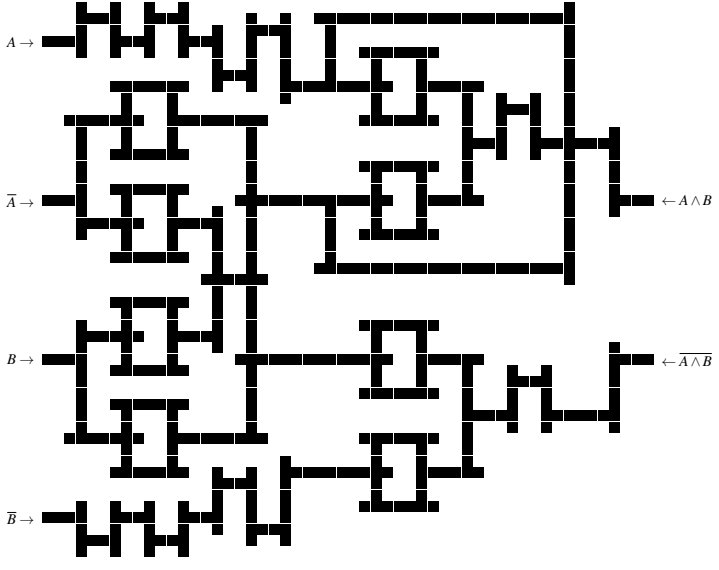


Copper : croisement

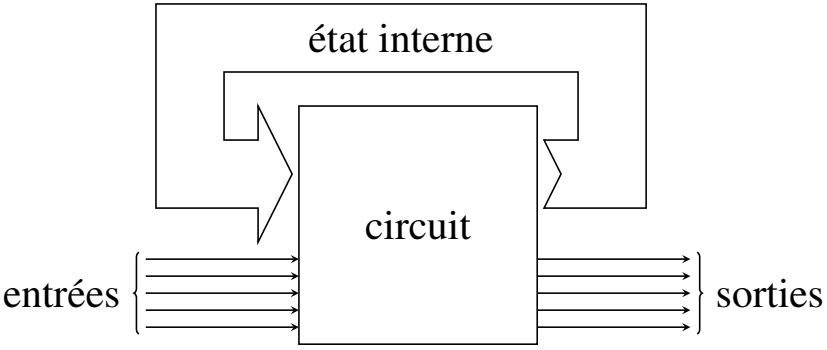




Copper : AND


$A \rightarrow$
$\overline{A} \rightarrow$
$\leftarrow A \wedge B$
$B \rightarrow$
$\leftarrow \overline{A \wedge B}$
$\overline{B} \rightarrow$

Copper : automates finis



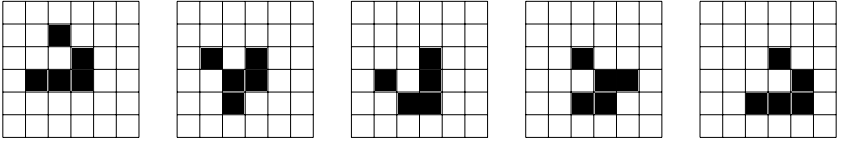
Théorème Copper est **universel pour les circuits booléens**.

La puissance Turing nécessite ici une **configuration ultimement périodique** contenant une **infinité** de cellules non quiescentes.

The Game of Life

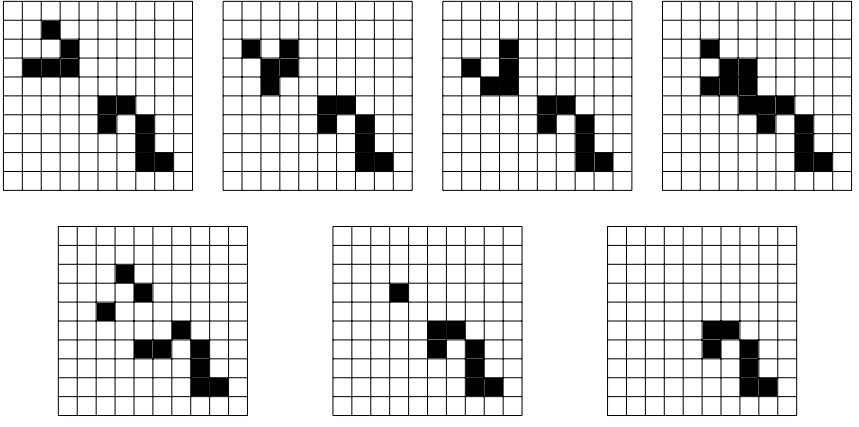
Théorème GoL est **universel pour les circuits booléens**.

La construction utilise des **gliders** comme signaux.

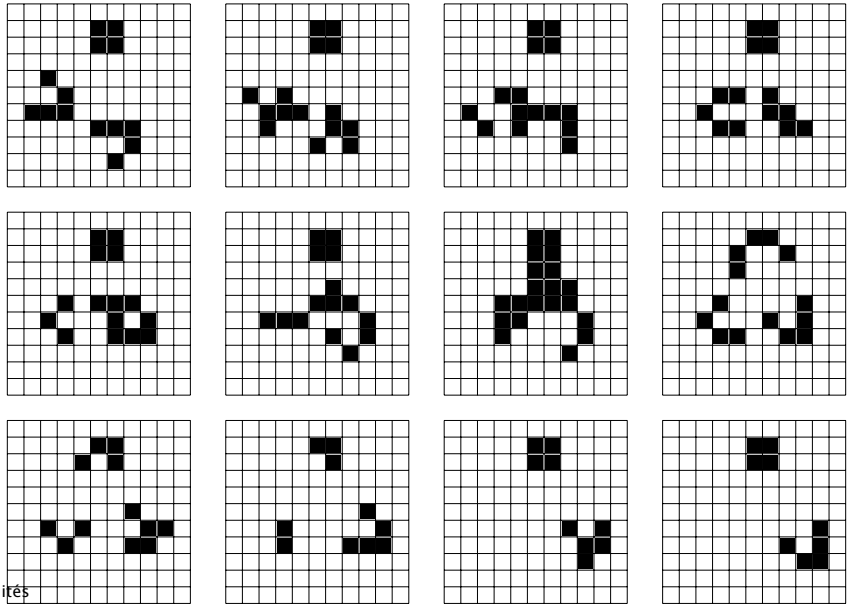


(Conway *et al.*, Winning Ways Vol. 2., 1971)

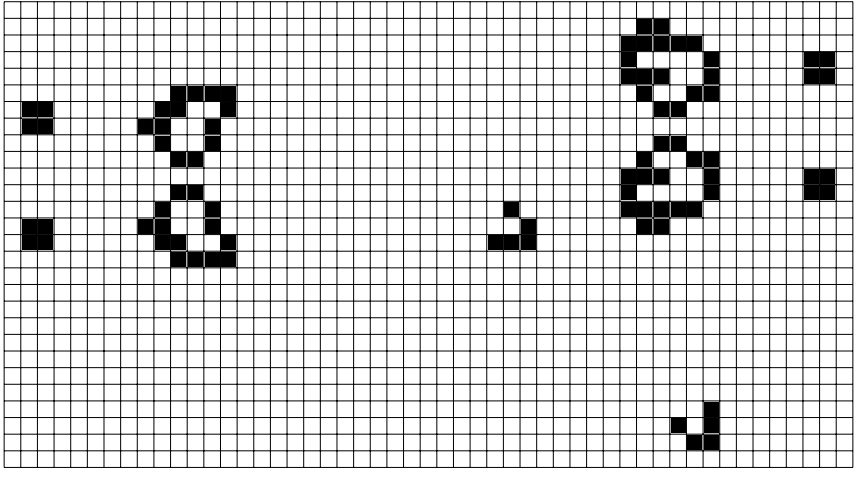
GoL : Eater



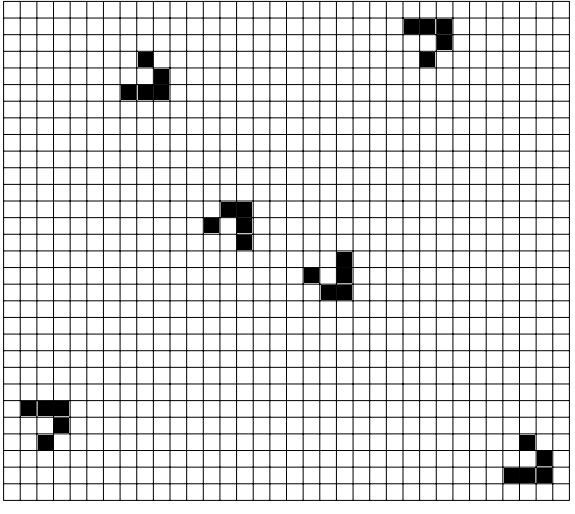
GoL : Duplicator



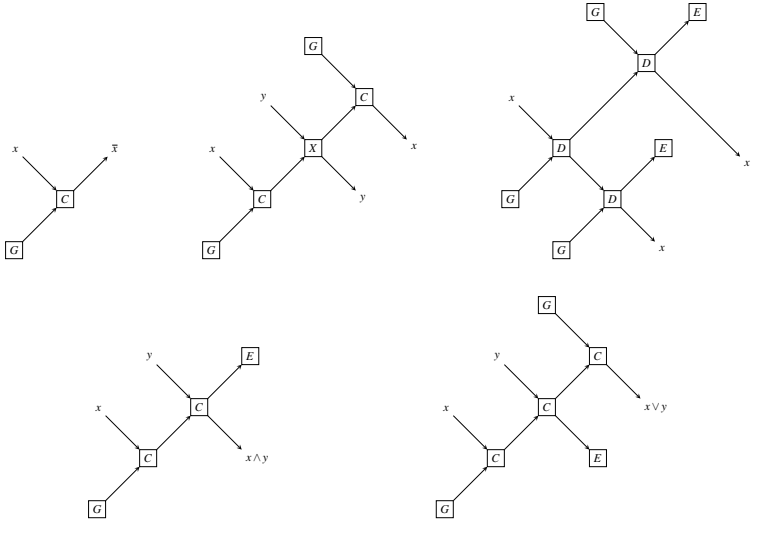
GoL : Gosper's p46 Gun



GoL : Xing



GoL : circuits



Universalité en 1D

Remarque La simulation de circuits booléens est **moins intuitive** en 1D mais il est facile de simuler des **modèles de calcul séquentiels** comme les machines de Turing.

(A. R. Smith III, Simple computation-universal cellular spaces, 1971)

1971	Smith III	18
1987	Albert & Culik II	14
1990	Lindgren & Nordhal	7
2004	Cook	2

Un AC est **Turing-universel** si

Universalité en 1D

Remarque La simulation de circuits booléens est **moins intuitive** en 1D mais il est facile de simuler des **modèles de calcul séquentiels** comme les machines de Turing.

(A. R. Smith III, Simple computation-universal cellular spaces, 1971)

1971	Smith III	18
1987	Albert & Culik II	14
1990	Lindgren & Nordhal	7
2004	Cook	2

Un AC est **Turing-universel** si... Quelle est la bonne **définition formelle** ?

Universalité en 1D

Remarque La simulation de circuits booléens est **moins intuitive** en 1D mais il est facile de simuler des **modèles de calcul séquentiels** comme les machines de Turing.

(A. R. Smith III, Simple computation-universal cellular spaces, 1971)

1971	Smith III	18
1987	Albert & Culik II	14
1990	Lindgren & Nordhal	7
2004	Cook	2

Un AC est **Turing-universel** si... Quelle est la bonne **définition formelle** ? C'est quoi un AC **non universel** ?

Universalité en 1D

Remarque La simulation de circuits booléens est **moins intuitive** en 1D mais il est facile de simuler des **modèles de calcul séquentiels** comme les machines de Turing.

(A. R. Smith III, Simple computation-universal cellular spaces, 1971)

1971	Smith III	18
1987	Albert & Culik II	14
1990	Lindgren & Nordhal	7
2004	Cook	2

Un AC est **Turing-universel** si... Quelle est la bonne **définition formelle** ? C'est quoi un AC **non universel** ?

Pas de **définition formelle** qui fasse **consensus**. Convergence improbable vers un consensus. (Durand & Roka, 1999)

MT *à la* Smith III

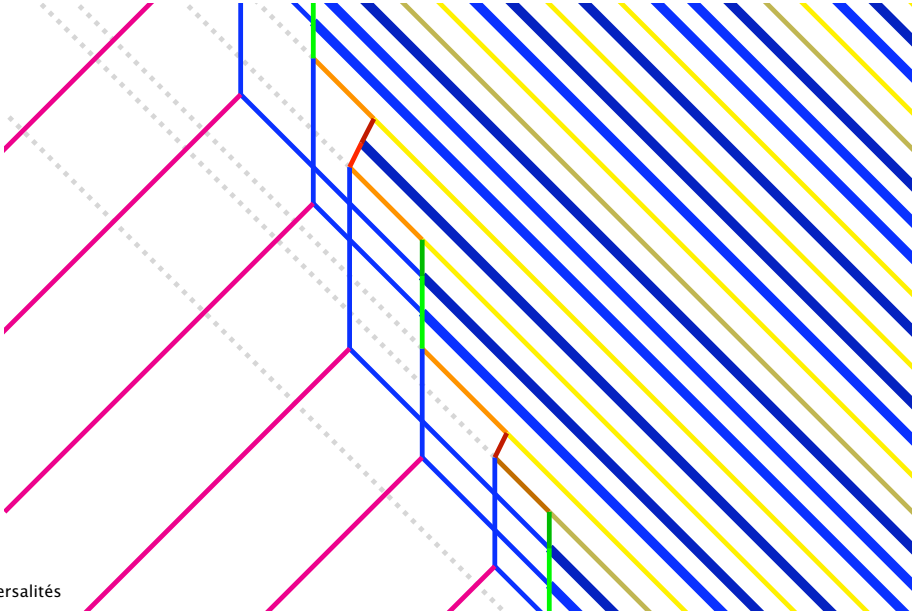
B	B	(q_0, a)	a	B
B	B	a	(q_1, B)	B
B	B	(q_1, b)	B	B
B	(q_0, B)	b	B	B
B	B	(q_0, a)	B	B

Le temps va toujours vers le haut !

MT *à la* Lindgren & Nordhal

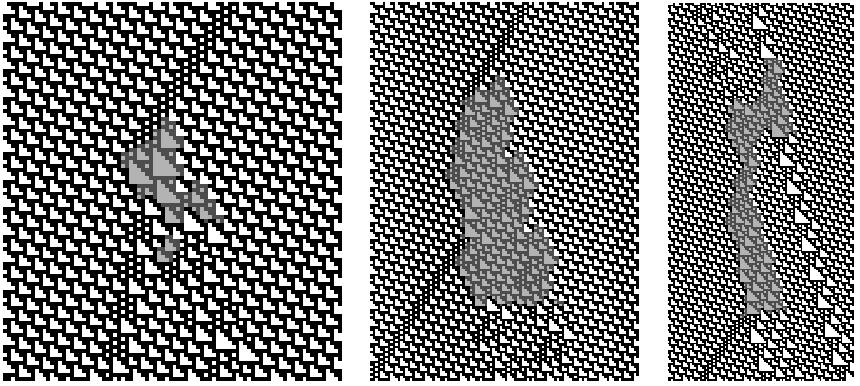
B	$\bullet$	B	$\bullet$	a	q_0	$\bullet$	a	$\bullet$	B
B	$\leftrightarrow$	B	$\leftrightarrow$	a	$\leftrightarrow$	q_0	a	$\leftrightarrow$	B
B	$\bullet$	B	$\bullet$	a	$\bullet$	q_1	B	$\bullet$	B
B	$\leftrightarrow$	B	$\leftrightarrow$	a	q_1	$\leftrightarrow$	B	$\leftrightarrow$	B
B	$\bullet$	B	$\bullet$	q_1	b	$\bullet$	B	$\bullet$	B
B	$\leftrightarrow$	B	q_1	$\leftrightarrow$	b	$\leftrightarrow$	B	$\leftrightarrow$	B
B	$\bullet$	B	q_0	$\bullet$	b	$\bullet$	B	$\bullet$	B
B	$\leftrightarrow$	B	$\leftrightarrow$	q_0	b	$\leftrightarrow$	B	$\leftrightarrow$	B
B	$\bullet$	B	$\bullet$	q_0	a	$\bullet$	B	$\bullet$	B

Universalité de la règle 110 *à la* Cook



Cook : détails...

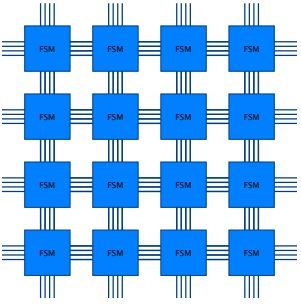
Utilise d'**énormes** particules et collisions...



Théorème La règle 110 est **Turing-universelle**.

Une autre voie vers l'universalité

Remarque Les circuits booléens forment un **modèle de calcul massivement parallèle**.



C'est la clé vers une notion plus forte d'**universalité intrinsèque**, la capacité d'un AC à simuler tous les AC.

Classification par groupage

Idée définir un **préordre** sur les AC, préordre dont les **classes d'équivalence** capturent des comportements des AC.

Définition Un AC $\mathcal{A}$ est **algorithmiquement plus simple** qu'un AC $\mathcal{B}$ si tous les diagrammes espace-temps de $\mathcal{A}$ sont des diagrammes espace-temps de $\mathcal{B}$ *(à renommage uniforme des états près)*.

Formellement, $\mathcal{A} \subseteq \mathcal{B}$ s'il existe $\varphi : S_{\mathcal{A}} \to S_{\mathcal{B}}$ injective telle que $\overline{\varphi} \circ G_{\mathcal{A}} = G_{\mathcal{B}} \circ \overline{\varphi}$. C'est-à-dire si le diagramme suivant commute :

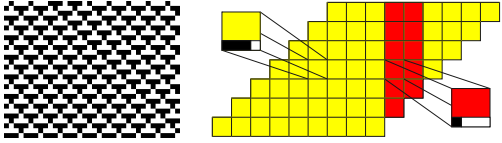
$$\begin{array}{ccc} C & \xrightarrow{\varphi} & \overline{\varphi}(C) \\ {\scriptstyle G_{\mathcal{A}}}\downarrow & & \downarrow{\scriptstyle G_{\mathcal{B}}} \\ G_{\mathcal{A}}(C) & \xrightarrow[\varphi]{} & \overline{\varphi}(G_{\mathcal{A}}(C)) \end{array}$$

Groupage

Quotienter l'ensemble des AC par les **transformations affines discrètes**, les seules transformations géométriques qui préservent les AC.

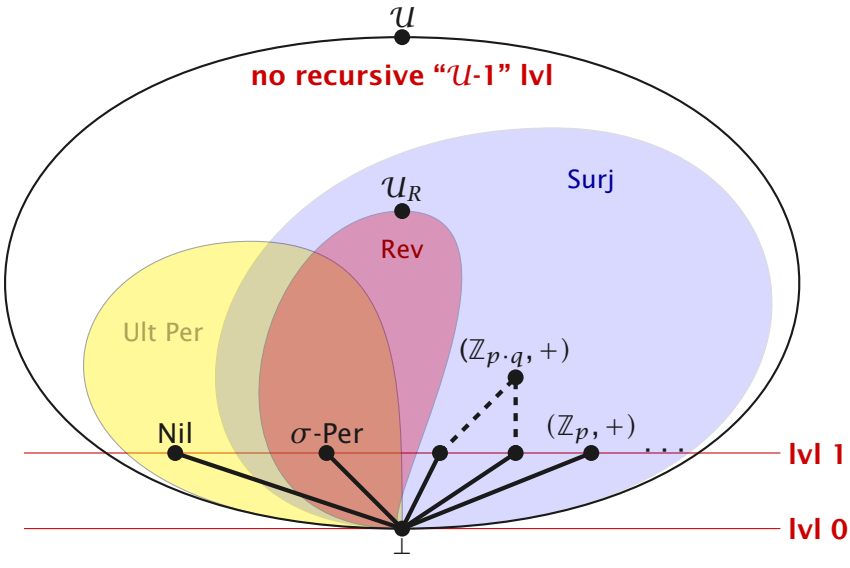
Le transformé $\langle m, n, k\rangle$ de $\mathcal{A}$ vérifie :

$$G_{\mathcal{A}^{\langle m,n,k\rangle}} = \sigma^k \circ o^m \circ G_{\mathcal{A}}^n \circ o^{-m} \ .$$



Définition Le **groupage** est défini par $\mathcal{A} \preccurlyeq \mathcal{B}$ s'il existe $\langle m, n, k\rangle$ et $\langle m', n', k'\rangle$ tels que $\mathcal{A}^{\langle m,n,k\rangle} \subseteq \mathcal{B}^{\langle m',n',k'\rangle}$.

The big picture



Universalité intrinsèque

Définition Un AC $\mathcal{U}$ est **intrinséquement universel** s'il est maximal pour $\preccurlyeq$, *i.e.* pour tout AC $\mathcal{A}$, il existe α telle que $\mathcal{A} \subseteq \mathcal{U}^\alpha$.

Théorème Il existe des AC **Turing universels** qui ne sont pas intrinsèquement universels.

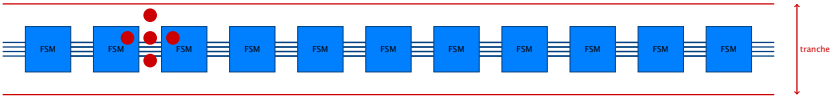
où l'universalité Turing est obtenue de manière très classique afin de satisfaire toute définition raisonnable.

Théorème Les AC 2D **universels pour les circuits** sont aussi **intrinséquement universels**.

(Delorme et al., 2011)

Avec des circuits booléens

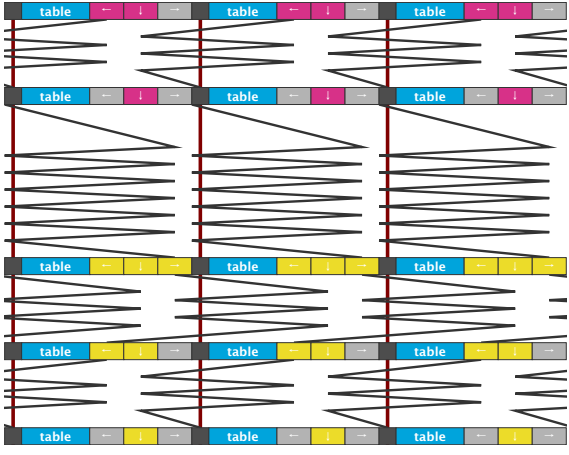
Tout AC **2D universel pour les circuits** peut être converti en un AC **1D intrinséquement universel** **[Banks 1970]**.



Découper des **tranches** de configuration périodique, les concaténer **horizontalement**, utiliser un **voisinage adéquat**.

Quitte à **augmenter le nombre d'états**, on peut toujours se ramener à un voisinage de rayon 1.

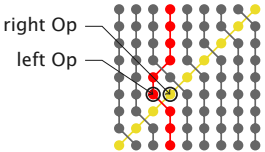
Avec des MT massivement parallèles

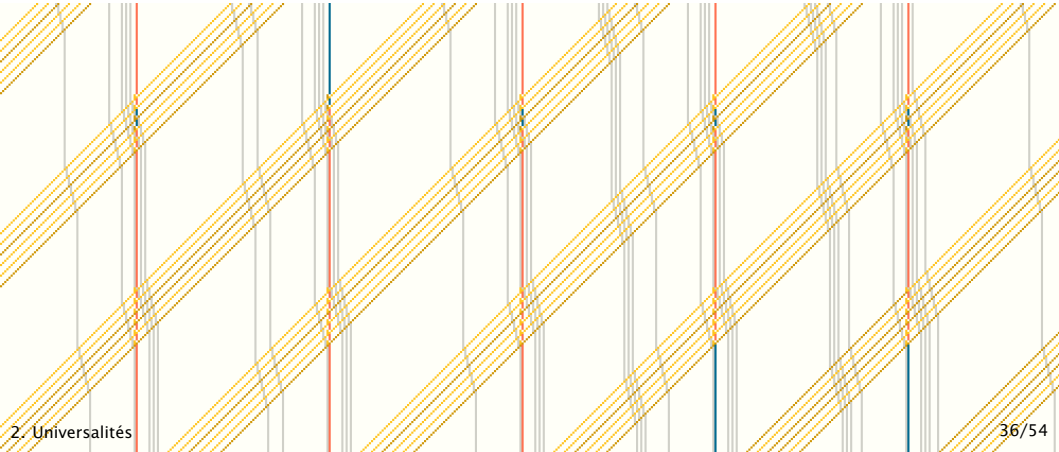


Utiliser une **tête Turing** par macro-cellule, définir une **suite de déplacements** qui soit **indépendante** du calcul.

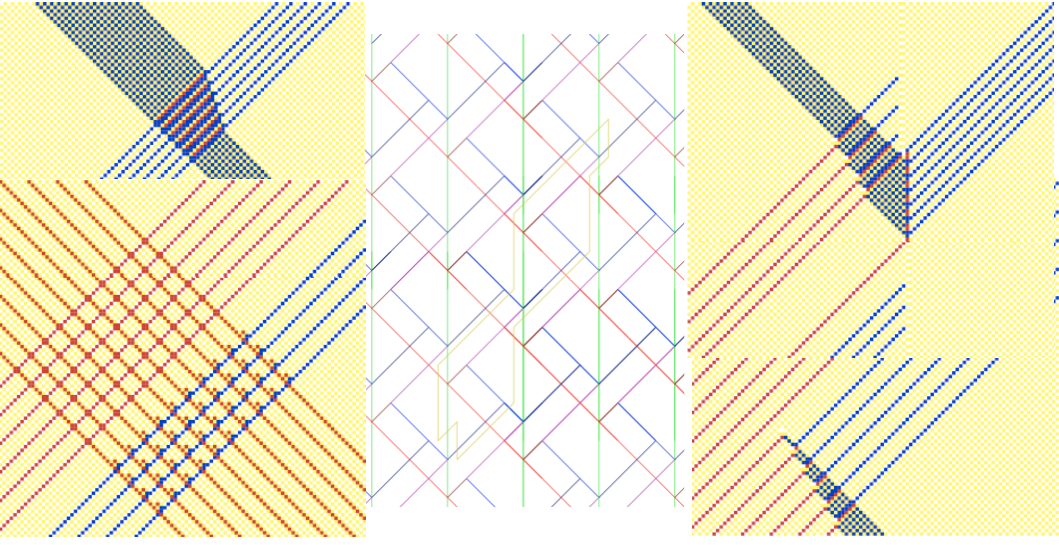
Plus malin : 6 états

Un AC intrinsèquement universel à **6 états** et rayon 1 qui plonge des **circuit booléens** dans la ligne de cellules.





En tirant la langue : 4 états



Intrinsèque universalité

année	auteur	d	voisins	états
1966	von Neumann	2	5	29
1968	Codd	2	5	8
1970	Conway	2	8	2
1970	Banks	2	5	2
		1	3	18
1987	Albert & Culik II	1	3	14
2002	O	1	3	6
2008	O & Richard	1	3	4

Oups, on a perdu Smith III, Lindgren-Nordhal et Cook. . .

Neary & Woods 2006 La **règle 110** a une prédiction **P-complète**.

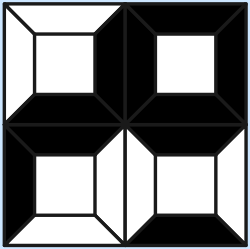
Problème ouvert Est-elle **intrinsèquement universelle** ?

1. Automates cellulaires

2. Universalités

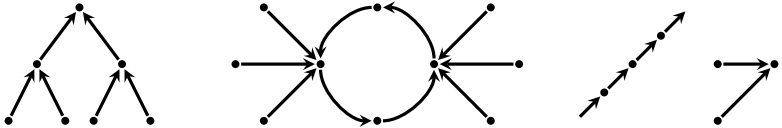
3. Dynamiques indécidables

4. Conclusion



Systèmes dynamiques discrets

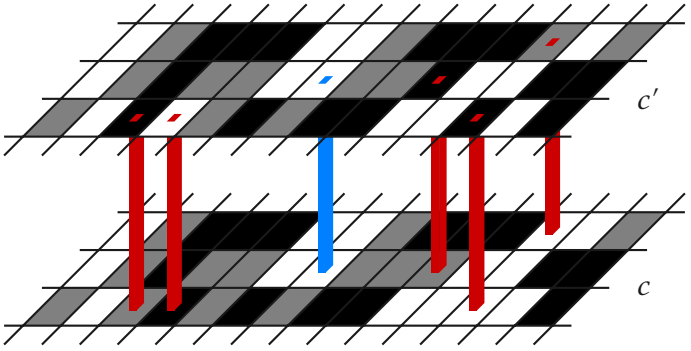
Définition Un **SDD** (X, F) est constitué d'un **espace topologique** muni d'une **fonction continue** $F : X \to X$.



Définition L'**orbite** d'un point $x \in X$ est la suite $(F^n(x))$ obtenue en itérant F.

Topologie de Cantor

$$\forall c, c' \in S^{\mathbb{Z}^2}, \quad d(c,c') = 2^{-\min\left\{\|p\|_\infty \,\middle|\, c_p \neq c'_p\right\}}$$



$d(c,c') = 1/8$

Proposition La topologie de Cantor est **compacte**.

Continuité

Définition Une application $G : S^{\mathbb{Z}^2} \to S^{\mathbb{Z}^2}$ est **locale** en $p \in \mathbb{Z}^2$ s'il existe un rayon r tel que :

$$\forall c, c' \in S^{\mathbb{Z}^2}, \quad \left[c_{|r}\right] = \left[c'_{|r}\right] \Rightarrow G(c)_p = G(c')_p \quad .$$

Proposition Une application $G : S^{\mathbb{Z}^2} \to S^{\mathbb{Z}^2}$ est **continue** si et seulement si elle est **locale en tout point**.

Théorème de Curtis-Hedlund-Lyndon

Définition La **translation** $\sigma_k : S^{\mathbb{Z}^2} \to S^{\mathbb{Z}^2}$ de vecteur $k \in \mathbb{Z}^2$ vérifie :
$$\forall c \in S^{\mathbb{Z}^2}, \forall p \in \mathbb{Z}^2, \quad \sigma_k(c)_p = c_{p-k} \quad .$$

Théorème[Hedlund 1969] Les **fonctions globales** d'AC sont exactement les applications **continues** qui **commutent avec les translations**.

Un **AC** peut être donné par sa **fonction globale**. La **composition** de deux AC, l'**inverse** d'un AC bijectif sont des AC.

Ensemble limite

Définition L'**ensemble limite** d'un AC $\left(S^{\mathbb{Z}^2}, F\right)$ est l'ensemble des configurations qui peuvent apparaître à tout temps :

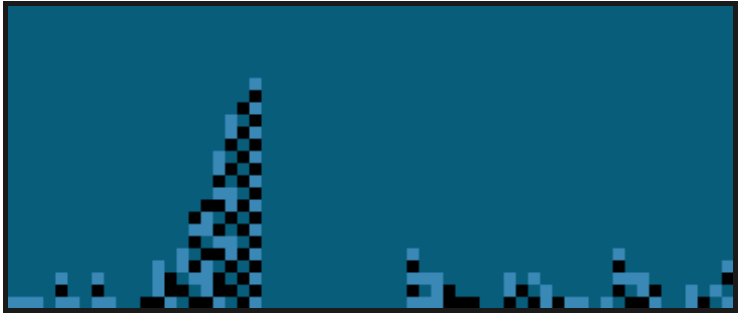
$$\Lambda_F = \bigcap_{n \in \mathbb{N}} F^n \left(S^{\mathbb{Z}^2}\right) \quad .$$

Proposition L'**ensemble limite** est un **compact non vide invariant par translation**.

$F^n \left(S^{\mathbb{Z}^2}\right)$ est un compact non vide invariant par translation et $F^{n+1} \left(S^{\mathbb{Z}^2}\right) \subseteq F^n \left(S^{\mathbb{Z}^2}\right)$.

Convergence vers un même point fixe

Définition Un AC est **nilpotent** si toute configuration atteint en un temps fini la configuration q-monochromatique où q est un état quiescent.



Proposition Un AC est **nilpotent** si et seulement si son ensemble limite est un **singleton**.

Propriétés

Proposition[CPY89] L'**ensemble limite** d'un AC est ou bien un **singleton** ou bien **infini**.

Proposition Un AC nilpotent converge **uniformément** vers son point fixe : $F^t\left(S^{\mathbb{Z}^2}\right) = \{\overline{q}\}$.

Considérer une **configuration univers** qui contient tous les motifs finis possibles.

Propriétés

Proposition[CPY89] L'**ensemble limite** d'un AC est ou bien un **singleton** ou bien **infini**.

Proposition Un AC nilpotent converge **uniformément** vers son point fixe : $F^t\left(S^{\mathbb{Z}^2}\right) = \{\overline{q}\}$.

Considérer une **configuration univers** qui contient tous les motifs finis possibles.

Exercice Écrire un **programme** qui, étant donné un automate cellulaire, **décide** si toute configuration initiale atteint un même point fixe.

Détecter la nilpotence

Semi-algorithme

n = 0
Répéter :
    Si $F^n\left(S^{\mathbb{Z}^2}\right) = \{\overline{q}\}$ alors répondre OUI
    n = n + 1

Détecter la nilpotence

Semi-algorithme

```
n = 0
Répéter  :
    Si F^n(S^{Z^2}) = {q̄} alors répondre OUI
    n = n + 1
```

Affirmation Il n'existe **aucun algorithme** qui, étant donné un AC (S, N, f), **décide** s'il est nilpotent.

Détecter la nilpotence

Semi-algorithme

```
n = 0
Répéter :
    Si F^n (S^{Z^2}) = {q̄} alors répondre OUI
    n = n + 1
```

Nil $= \{(S, N, f) \mid F \text{ est nilpotent}\}$

Avec un codage raisonnable implicite...

Proposition **Nil** n'est **pas récursif**.

Des problèmes indécidables

Une technique pour dériver de nombreux problèmes indécidables à partir d'un premier consiste à utiliser des **réductions** : des pré-ordres sur les langages qui **préservent la récursivité**.

Définition Soient $A \subseteq \Sigma^*$ et $B \subseteq \Gamma^*$ deux langages. Une **réduction** (many-one) de A à B est une fonction totale calculable $f : \Sigma^* \to \Gamma^*$ telle que

$$u \in A \Leftrightarrow f(u) \in B \qquad \forall u \in \Sigma^* \quad .$$

Le langage A **se réduit** au langage B, noté $A \leqslant_m B$ s'il existe une réduction de A à B.

Proposition La relation $\leqslant_m$ est une relation de pré-ordre.

Un problème indécidable

Théorème Le **problème de l'arrêt** est **indécidable**.

Le lang

Par l'a
constr
1. ex
2. si
3. si

Par co

Coro

Problème de décision

Définition Un **probl**
ensemble récu

Le **langage**
instances po

Problème de l'a

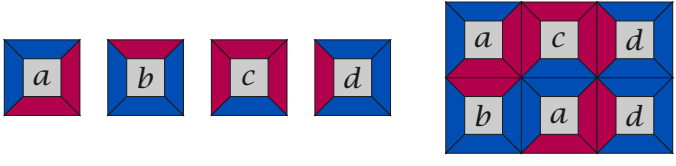
entrée : *une ma*
question : *est-ce*

Définition Un problè
associé est récursif, **in**

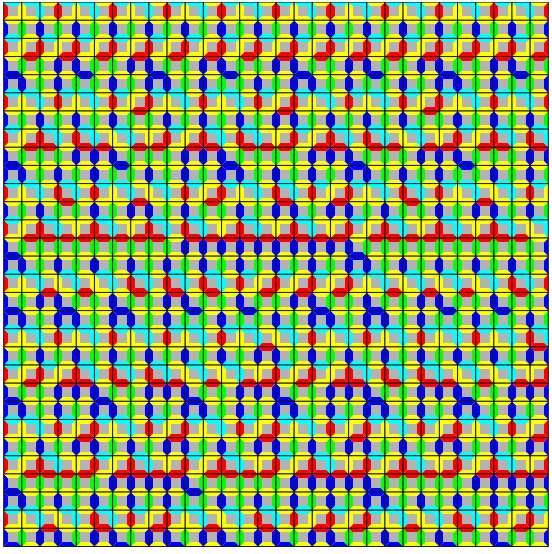
The Domino Problem (DP)

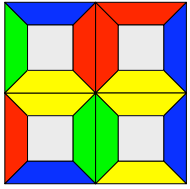
*"Assume we are **given a finite set of square plates** of the same size with **edges colored**, each in a different manner. Suppose further there are infinitely many copies of each plate (plate type). We are **not permitted to rotate or reflect a plate**. The question is to find an effective procedure by which we can **decide**, for each given finite set of plates, **whether we can cover up the whole plane** (or, equivalently, an infinite quadrant thereof) **with copies of the plates subject to the restriction that adjoining edges must have the same color**."*

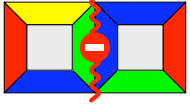
(Wang, 1961)



Tuiles de Wang







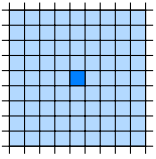
co-Pavabilité

Proposition Il existe un semi-algorithme pour décider si un jeu de tuiles ne pave pas le plan.

Considérer des pavages de carrés de plus en plus grands, en extraire un pavage du plan tout entier.

Apériodicité

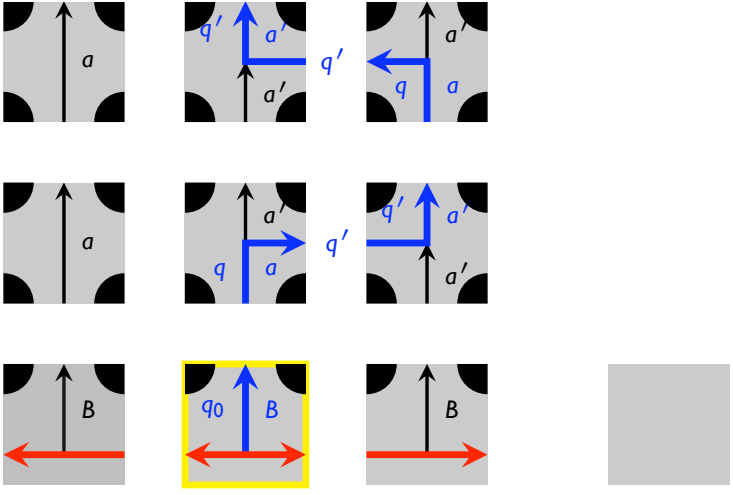
Définition Un pavage est **apériodique** s'il n'admet aucune période non triviale.



Définition Un jeu de tuiles est **apériodique** s'il admet un pavage et si tous ses pavages sont apériodiques.

Remarque S'il n'existait **pas de jeux de tuiles apériodiques**, le Domino Problem serait **décidable**.

Pavabilité à tuile fixée



Indécidabilité de DP

Théorème[Berger64] **DP** est **indécidable**.

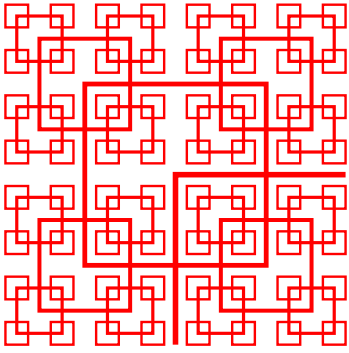
Remarque Nécessite des jeux de tuiles **apériodiques**.

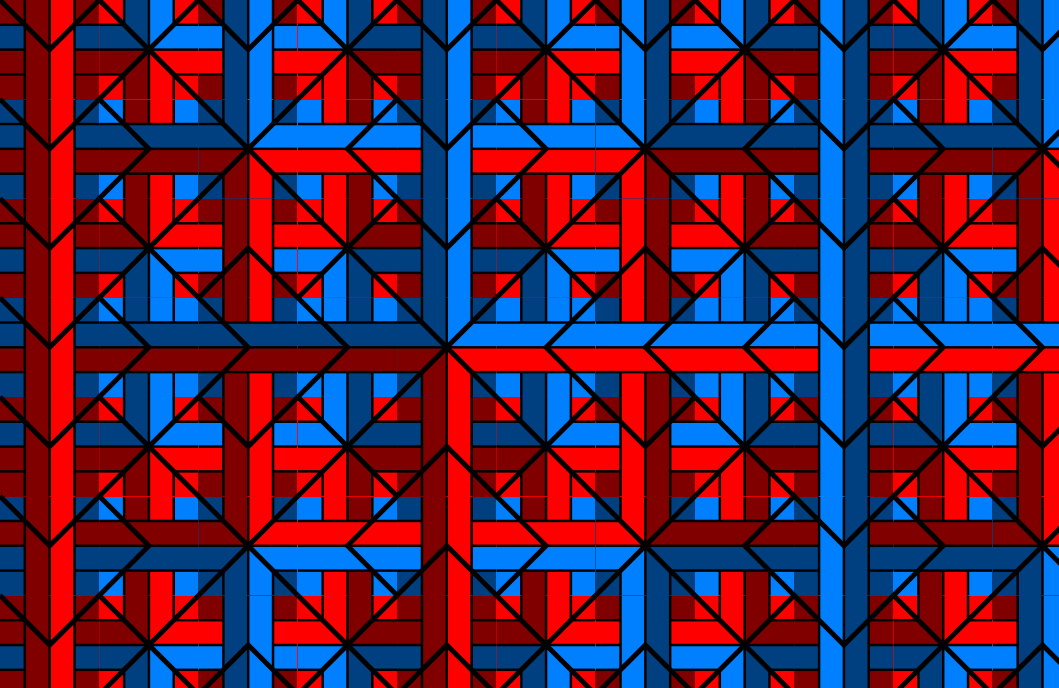
Idée de la preuve

Forcer une **structure auto-simulaire** (apériodique) à l'aide de règles locales.

Insérer **partout** le calcul d'une **machine de Turing**.

Remarque Nombreuses preuves !





Détecter la nilpotence

Théorème[CPY89] **Nil** est **indécidable** en 2D.

Montrons que $K_0 \leqslant_m \overline{\textbf{DP}} \leqslant_m$ **Nil**

Détecter la nilpotence

Théorème[CPY89] **Nil** est **indécidable** en 2D.

Montrons que $K_0 \leqslant_m \overline{\textbf{DP}} \leqslant_m$ **Nil**

Proposition $\overline{\textbf{DP}} \leqslant_m$ **Nil**

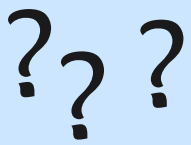
Étant donné un jeu de tuiles de Wang τ, construire un AC d'ensemble d'états $\tau \cup \{\perp\}$ où $\perp$ est un état envahissant d'erreur de pavage.

1. Automates cellulaires

2. Universalités

3. Dynamiques indécidables

4. Conclusion



Le mot de la fin

?? ?

Les **automates cellulaires**, ce n'est pas juste **simuler expérimentalement** ou **construire** des automates qui font pouet.

La dualité **système dynamique** / objet **combinatoire discret** fait émerger des phénomènes **calculatoires** : c'est aussi de l'informatique !

Pour une version longue en anglais avec conseil logiciel et biblio :
`https://www.univ-orleans.fr/lifo/Members/Nicolas.Ollinger/acuc/`

Le mot de la fin

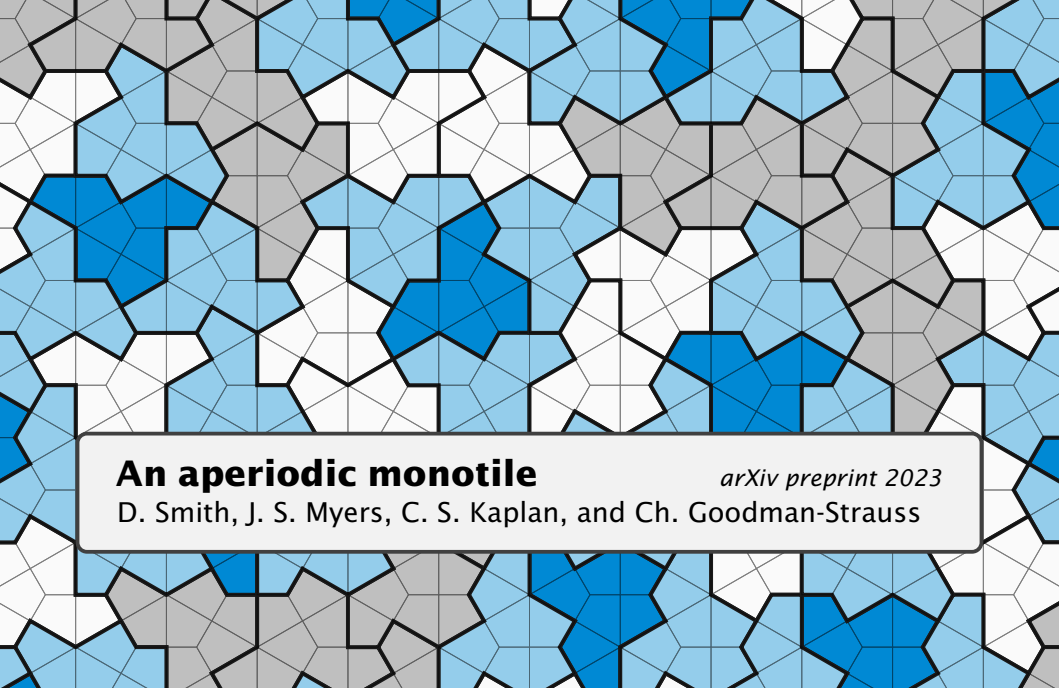
?? ?

Les **automates cellulaires**, ce n'est pas juste **simuler expérimentalement** ou **construire** des automates qui font pouet.

La dualité **système dynamique** / objet **combinatoire discret** fait émerger des phénomènes **calculatoires** : c'est aussi de l'informatique !

Pour une version longue en anglais avec conseil logiciel et biblio :
`https://www.univ-orleans.fr/lifo/Members/Nicolas.Ollinger/acuc/`

One last thing...



An aperiodic monotile

arXiv preprint 2023

D. Smith, J. S. Myers, C. S. Kaplan, and Ch. Goodman-Strauss