Automates cellulaires : du local au global.

Inter-académiques, Lyon

Nathalie AUBRUN

Chargée de recherche CNRS, LIP, ENS de Lyon

Mercredi 11 décembre 2013

ENS DE LYON

Plan de l'exposé

Présentation

Les automates cellulaires sont un modèle de calcul, introduits dans les années 60, qui présentent des intérêts

- en physique, comme modèles discrets de systèmes physiques (lattice gas : propagation/collision de particules)
- en informatique, comme modèle de calcul massivement parallèle
- en mathématiques, comme endomorphismes du décalage plein (dynamique symbolique)

Caractéristiques du modèle

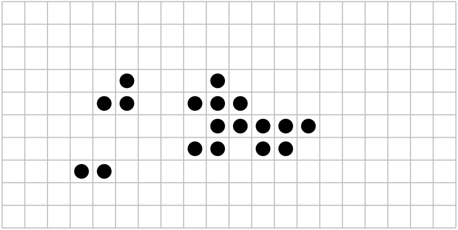
Il possèdent diverses propriétés du monde physique : ce sont des modèles

- massivement parallèles
- homogènes en temps et en espace
- où toutes les interactions sont locales
- qui peuvent respecter le principe de réversibilité et les lois de conservation (choix de la règle locale)

Un exemple célèbre : le jeu de la vie

Automate cellulaire de dimension 2, dû à John Conway (1970).

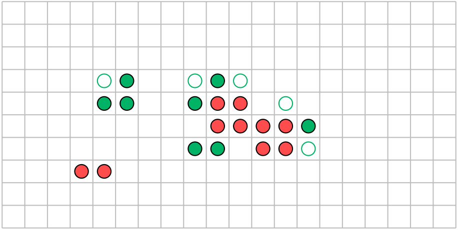
- On se place sur un damier infini dont les cases (cellules) sont colorées en blanc (cellules mortes) ou en noir (cellules vivantes).
- A chaque étape de temps discret, chaque cellule compte le nombre n de cellules vivantes parmi ses huit voisines, qui détermine son nouvel état :
  - une cellule vivante survit ssi $n = 2$ ou 3 ;
  - une cellule nait ssi $n = 3$.
- Toutes les cellules changent d'état en même temps.



Un exemple célèbre : le jeu de la vie

Automate cellulaire de dimension 2, dû à John Conway (1970).

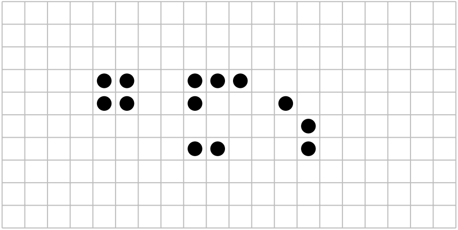
- On se place sur un damier infini dont les cases (cellules) sont colorées en blanc (cellules mortes) ou en noir (cellules vivantes).
- A chaque étape de temps discret, chaque cellule compte le nombre n de cellules vivantes parmi ses huit voisines, qui détermine son nouvel état :
  - une cellule vivante survit ssi $n = 2$ ou 3 ;
  - une cellule nait ssi $n = 3$.
- Toutes les cellules changent d'état en même temps.



Un exemple célèbre : le jeu de la vie

Automate cellulaire de dimension 2, dû à John Conway (1970).

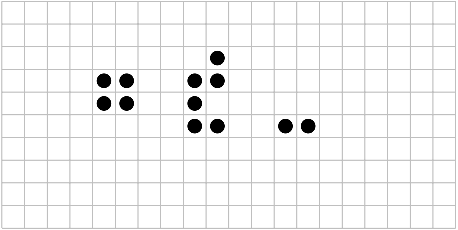
- On se place sur un damier infini dont les cases (cellules) sont colorées en blanc (cellules mortes) ou en noir (cellules vivantes).
- A chaque étape de temps discret, chaque cellule compte le nombre n de cellules vivantes parmi ses huit voisines, qui détermine son nouvel état :
  - une cellule vivante survit ssi $n = 2$ ou 3 ;
  - une cellule nait ssi $n = 3$.
- Toutes les cellules changent d'état en même temps.



Un exemple célèbre : le jeu de la vie

Automate cellulaire de dimension 2, dû à John Conway (1970).

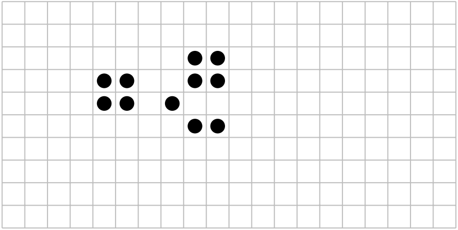
- On se place sur un damier infini dont les cases (cellules) sont colorées en blanc (cellules mortes) ou en noir (cellules vivantes).
- A chaque étape de temps discret, chaque cellule compte le nombre n de cellules vivantes parmi ses huit voisines, qui détermine son nouvel état :
  - une cellule vivante survit ssi $n = 2$ ou 3 ;
  - une cellule nait ssi $n = 3$.
- Toutes les cellules changent d'état en même temps.



Un exemple célèbre : le jeu de la vie

Automate cellulaire de dimension 2, dû à John Conway (1970).

- On se place sur un damier infini dont les cases (cellules) sont colorées en blanc (cellules mortes) ou en noir (cellules vivantes).
- A chaque étape de temps discret, chaque cellule compte le nombre n de cellules vivantes parmi ses huit voisines, qui détermine son nouvel état :
  - une cellule vivante survit ssi $n = 2$ ou 3 ;
  - une cellule nait ssi $n = 3$.
- Toutes les cellules changent d'état en même temps.



Autre exemple classique : l'automate 110

Un automate cellulaire unidimensionnel sur l'alphabet binaire $\{0,1\}$. Les configurations sont donc des suites bi-infinies de 0 et de 1.

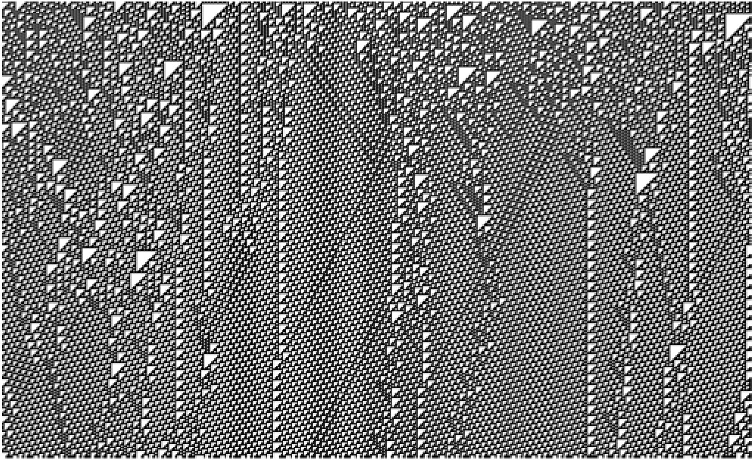
Chaque cellule est mise à jour selon son propre état et les états de ses plus proches voisines selon la règle :

$$\begin{array}{ccc} 111 & \mapsto & 0 \\ 110 & \mapsto & 1 \\ 101 & \mapsto & 1 \\ 100 & \mapsto & 0 \\ 011 & \mapsto & 1 \\ 010 & \mapsto & 1 \\ 001 & \mapsto & 1 \\ 000 & \mapsto & 0 \end{array}$$

Le nom 110 correspond au **nombre de Wolfram** de cet automate.

Diagramme espace-temps

Dessin qui représente l'évolution d'un automate cellulaire unidimensionnel, où l'espace et le temps sont représentés par les directions horizontale et verticale :



Définition des automates cellulaires sur $\mathbb{Z}^d$

- ensemble fini d'**états** S
- les **cellules** sont à coordonnées entières sur $\mathbb{Z}^d$
- une **configuration** : $\mathbb{Z}^d \to S$ attribue un état à chaque cellule
- l'ensemble des configurations est $S^{\mathbb{Z}^d}$
- un **voisinage** est un ensemble fini $V \subset \mathbb{Z}^d$
- les cellules voisines de $x \in \mathbb{Z}^d$ sont les $x + V$.

Règle locale d'un AC

Un automate cellulaire est en général donné par sa **règle locale**, qui est une fonction

$$f : S^n \to S$$

où n est la taille du voisinage.

L'état $f(a_1, a_2, \ldots, a_n)$ est le nouvel état d'une cellule dont les n voisins étaient dans les états $a_1, a_2, \ldots, a_n$ à l'étape de temps précédente.

Une dynamique sur les AC

Dynamique globale des automates cellulaires : une configuration c devient en une étape de temps une configuration e en appliquant la règle locale à chaque cellule de manière synchrone.

La transformation

$$F : S^{\mathbb{Z}^d} \to S^{\mathbb{Z}^d}$$

qui envoie c sur e est la fonction globale de l'automate cellulaire.

Une dynamique sur les AC

Dynamique globale des automates cellulaires : une configuration c devient en une étape de temps une configuration e en appliquant la règle locale à chaque cellule de manière synchrone.

La transformation

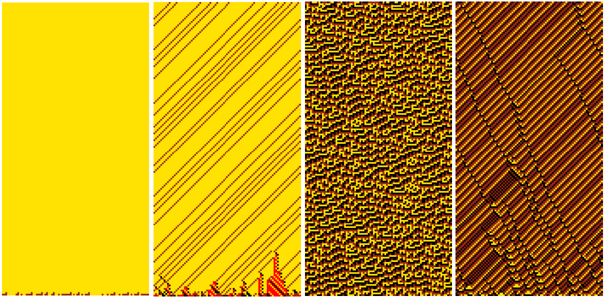
$$F : S^{\mathbb{Z}^d} \to S^{\mathbb{Z}^d}$$

qui envoie c sur e est la fonction globale de l'automate cellulaire.

Une fonction $F : S^{\mathbb{Z}^d} \to S^{\mathbb{Z}^d}$ est une fonction d'AC ssi elle est continue et commute avec le décalage.

Différents aspects informatiques

- observation empirique des diagrammes espace-temps (Wolfram)



Différents aspects informatiques

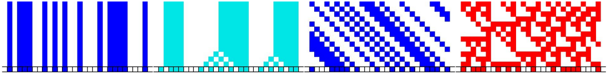
- observation empirique des diagrammes espace-temps (Wolfram)
- problèmes de nature algorithmique

  $\Rightarrow$ *test de primalité d'un entier, problèmes de synchronisation*

Différents aspects informatiques

- observation empirique des diagrammes espace-temps (Wolfram)
- problèmes de nature algorithmique

  $\Rightarrow$ *test de primalité d'un entier, problèmes de synchronisation*
- classification topologique (sensibilité aux conditions initiales, point d'équicontinuité)



Différents aspects informatiques

- observation empirique des diagrammes espace-temps (Wolfram)
- problèmes de nature algorithmique

  $\Rightarrow$ *test de primalité d'un entier, problèmes de synchronisation*

- classification topologique (sensibilité aux conditions initiales, point d'équicontinuité)
- les AC comme modèle de calcul (universalité, simulation)

  $\Rightarrow$ *Universalité Turing, génération universelle de motifs, simulation d'un AC par un autre*

Théorèmes classiques

Si un AC est bijectif, alors son inverse est aussi un AC.

$\Rightarrow$ argument de compacité.

Théorèmes classiques

Si un AC est bijectif, alors son inverse est aussi un AC.

$\Rightarrow$ argument de compacité.

Tout AC pré-injectif est surjectif.

$\Rightarrow$ preuve combinatoire, où l'on exprime la non-surjectivité (propriété globale) en termes de motifs orphelins (propriété locale).

Théorèmes classiques

Si un AC est bijectif, alors son inverse est aussi un AC.

$\Rightarrow$ argument de compacité.

Tout AC pré-injectif est surjectif.

$\Rightarrow$ preuve combinatoire, où l'on exprime la non-surjectivité (propriété globale) en termes de motifs orphelins (propriété locale).

Savoir si un AC est nilpotent, surjectif ($d \geq 2$) ou encore périodique est indécidable.

Plan de l'exposé

Crible d'Ératosthène

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Crible d'Ératosthène

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Crible d'Ératosthène

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

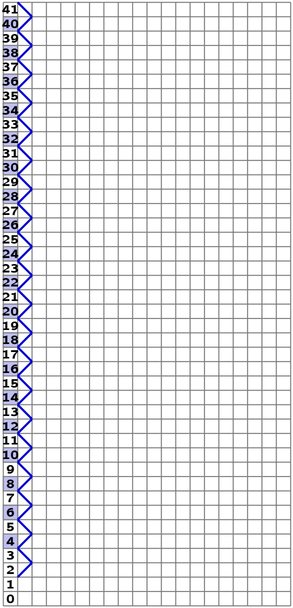
Crible d'Ératosthène

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

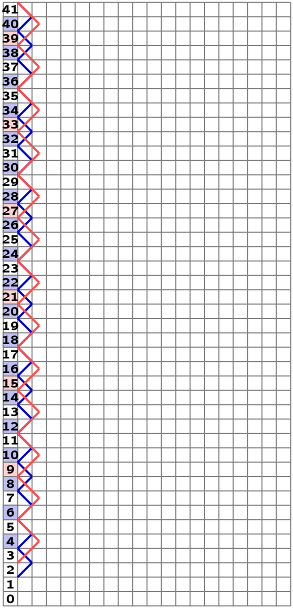
Crible d'Ératosthène

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

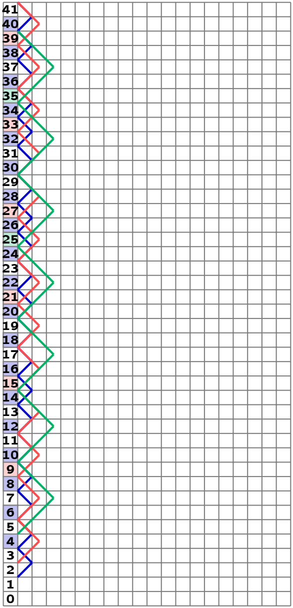
Un AC qui reconnaît les nombres premiers



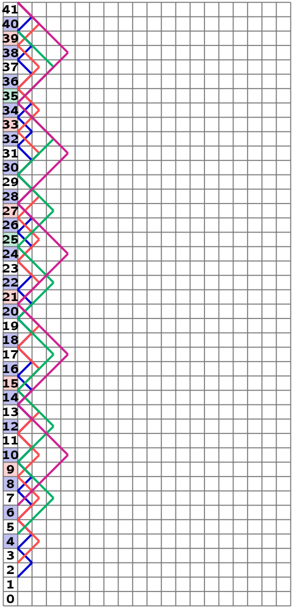
Un AC qui reconnaît les nombres premiers



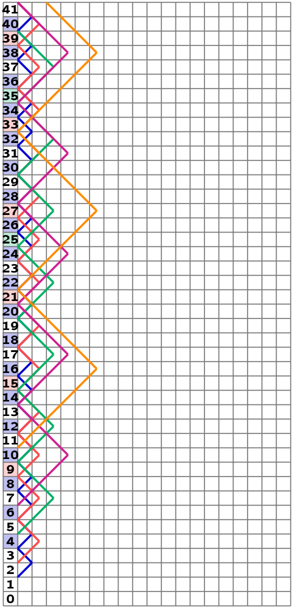
Un AC qui reconnaît les nombres premiers



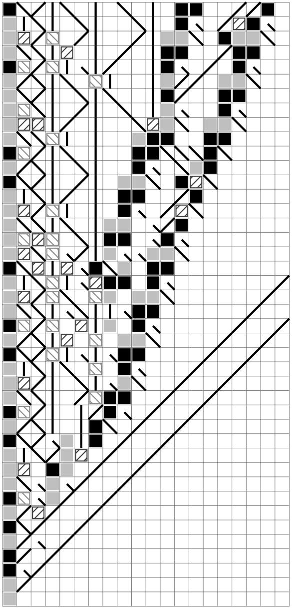
Un AC qui reconnaît les nombres premiers



Un AC qui reconnaît les nombres premiers



Un AC qui reconnaît les nombres premiers



Le problème des fusiliers

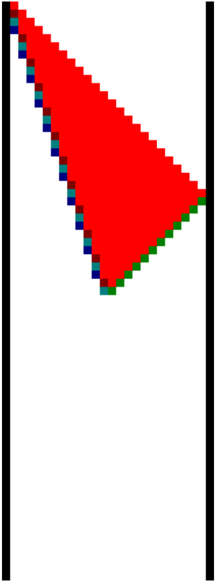
Problème de synchronisation, énoncé par Myhill en 1957.

- Une ligne de n fusiliers, tous identiques.
- Chaque fusilier peut parler uniquement à ses plus proches voisins.
- Les fusiliers doivent se mettre d'accord pour tirer tous **au même moment**.

Illustre le coté modèle de calcul massivement parallèle des AC.

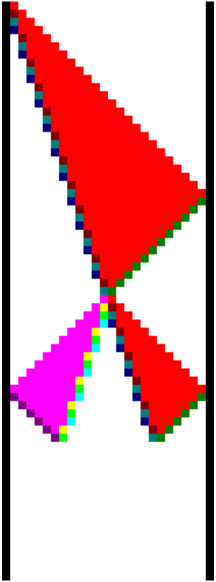
Exemple de résolution

Solution géométrique à base de signaux : méthode diviser pour régner.



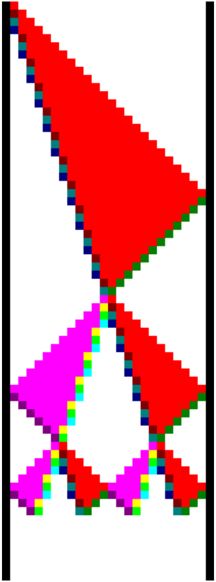
Exemple de résolution

Solution géométrique à base de signaux : méthode diviser pour régner.



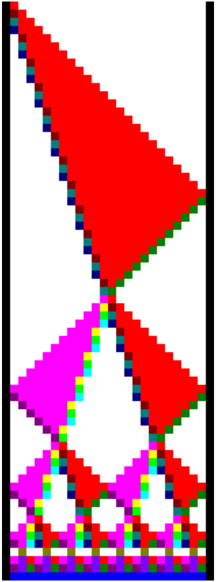
Exemple de résolution

Solution géométrique à base de signaux : méthode diviser pour régner.



Exemple de résolution

Solution géométrique à base de signaux : méthode diviser pour régner.



Exemple de résolution

Solution géométrique à base de signaux : méthode diviser pour régner.

Plan de l'exposé

Fourmi de Langton : description

Une fourmi se déplace sur la grille bi-dimensionnelle selon les règles suivantes :

- une cellule contient un galet (noté $\blacksquare$) ou non (noté $\square$)
- chaque fois qu'elle visite une cellule, la fourmi ramasse ($\blacksquare \to \square$) ou dépose ($\square \to \blacksquare$) un galet
- si la cellule visitée contient **un galet**, la fourmi tourne à **gauche** et avance
- si la cellule visitée est **vide**, la fourmi tourne à **droite** et avance

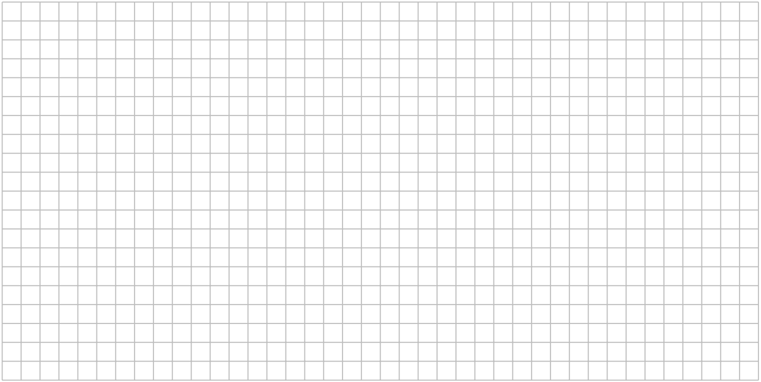
Fourmi de Langton : quelques propriétés

Il n'existe pas de trajectoire bornée.

Fourmi de Langton : quelques propriétés

Il n'existe pas de trajectoire bornée.

Remarque : une cellule donnée est atteinte soit toujours par sa voisine E ou W (cellule de **type H**), soit toujours par sa voisine N ou S (cellule de **type V**).



Fourmi de Langton : quelques propriétés

Il n'existe pas de trajectoire bornée.

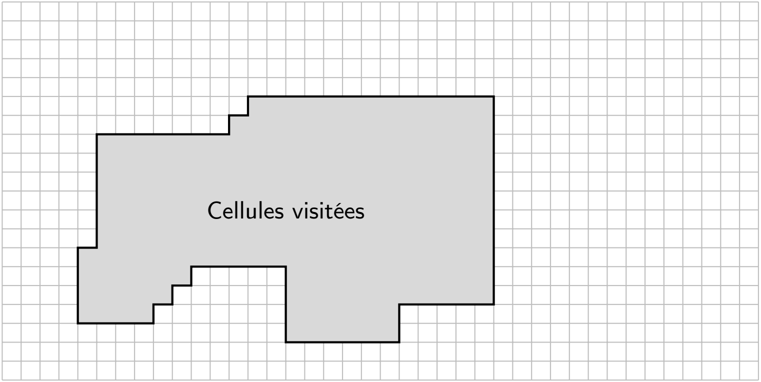
Remarque : une cellule donnée est atteinte soit toujours par sa voisine E ou W (cellule de **type H**), soit toujours par sa voisine N ou S (cellule de **type V**).

V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H
H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V
V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H
H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V
V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H
H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V
V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H
H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V
V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H
H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V
V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H
H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V
V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H
H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V
V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H
H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V
V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H
H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V
V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H
H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V	H	V

Fourmi de Langton : quelques propriétés

Il n'existe pas de trajectoire bornée.

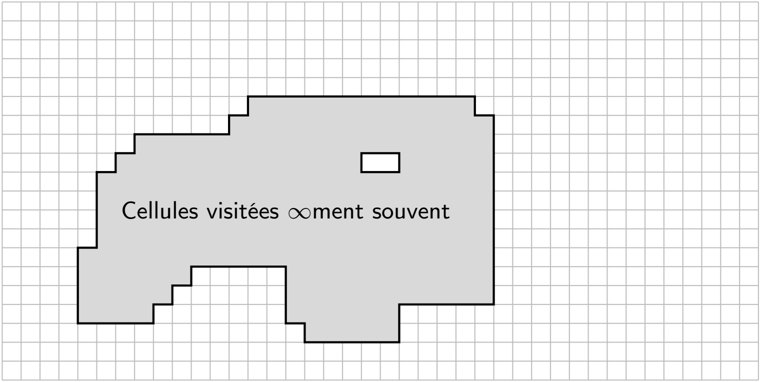
Supposons que la fourmi visite un ensemble borné de cellules.



Fourmi de Langton : quelques propriétés

Il n'existe pas de trajectoire bornée.

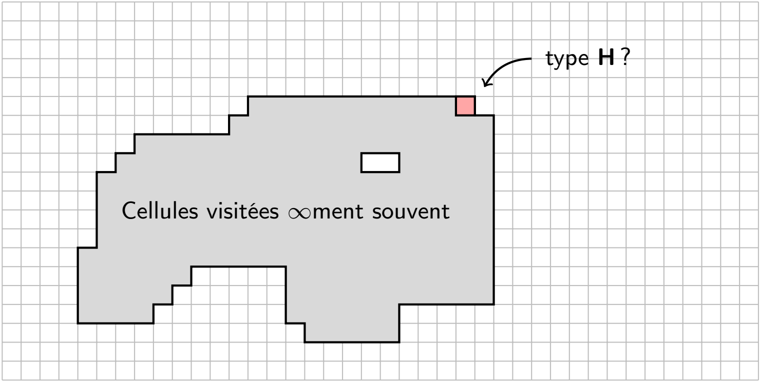
Supposons que la fourmi visite un ensemble borné de cellules.



Fourmi de Langton : quelques propriétés

Il n'existe pas de trajectoire bornée.

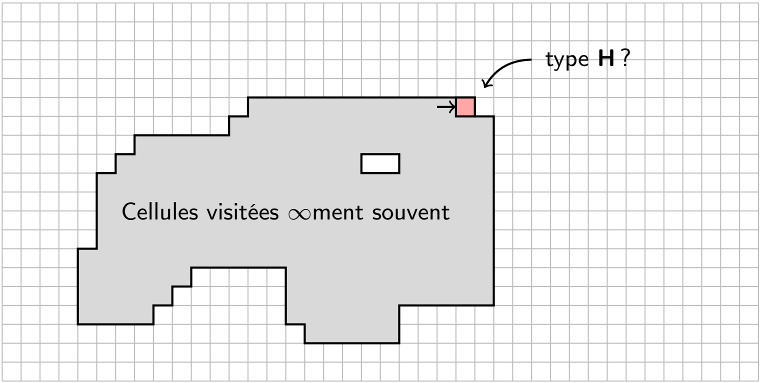
Supposons que la fourmi visite un ensemble borné de cellules.



Fourmi de Langton : quelques propriétés

Il n'existe pas de trajectoire bornée.

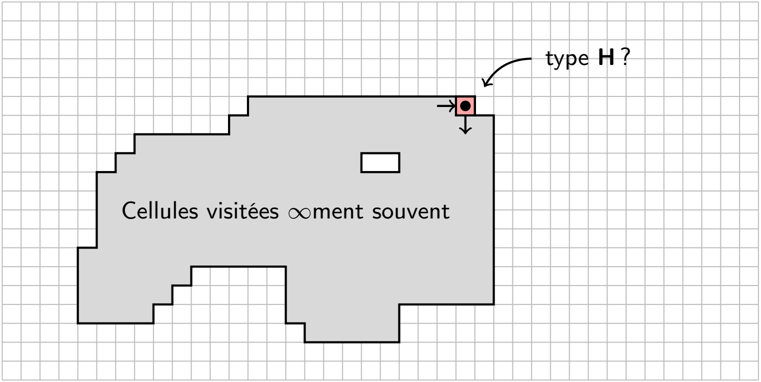
Supposons que la fourmi visite un ensemble borné de cellules.



Fourmi de Langton : quelques propriétés

Il n'existe pas de trajectoire bornée.

Supposons que la fourmi visite un ensemble borné de cellules.



Fourmi de Langton : quelques propriétés

Il n'existe pas de trajectoire bornée.

L'ensemble des cellules visitées infiniment souvent n'a pas de *coin*.

Fourmi de Langton : quelques propriétés

Il n'existe pas de trajectoire bornée.

L'ensemble des cellules visitées infiniment souvent n'a pas de *coin*.

D'après ce qu'on a pu observer sur les simulations...

A partir d'une configuration finie, la fourmi construit toujours une autoroute ?

Fourmi de Langton : quelques propriétés

Il n'existe pas de trajectoire bornée.

L'ensemble des cellules visitées infiniment souvent n'a pas de *coin*.

D'après ce qu'on a pu observer sur les simulations. . .

A partir d'une configuration finie, la fourmi construit toujours une autoroute ?

$\Rightarrow$ Pas de preuve connue à ce jour !

Fourmi de Langton : quelques propriétés

Il n'existe pas de trajectoire bornée.

L'ensemble des cellules visitées infiniment souvent n'a pas de *coin*.

D'après ce qu'on a pu observer sur les simulations. . .

A partir d'une configuration finie, la fourmi construit toujours une autoroute ?

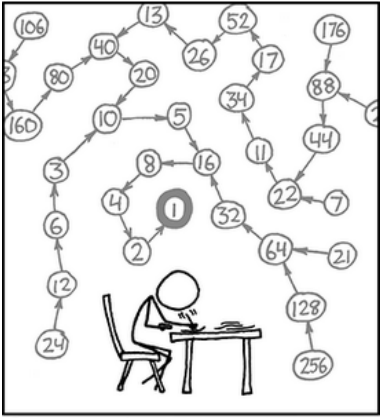
$\Rightarrow$ Pas de preuve connue à ce jour !

La fourmi de Langton est *Turing-universelle*.

Algorithme de Syracuse

Soit $u_0 \geq 1$.

- si u_n est pair, alors $u_{n+1} \leftarrow \frac{u_n}{2}$
- sinon, alors $u_{n+1} \leftarrow 3u_n + 1$



THE COLLATZ CONJECTURE STATES THAT IF YOU PICK A NUMBER, AND IF IT'S EVEN DIVIDE IT BY TWO AND IF IT'S ODD MULTIPLY IT BY THREE AND ADD ONE, AND YOU REPEAT THIS PROCEDURE LONG ENOUGH, EVENTUALLY YOUR FRIENDS WILL STOP CALLING TO SEE IF YOU WANT TO HANG OUT.

La conjecture de Collatz version AC

Comment encoder la multiplication par un entier dans un AC ?

- Problème du choix de la base : si on multiplie par 3 en base 10, les retenues peuvent se propager arbitrairement loin !
- On peut construire un AC qui multiplie par d en base n, avec d **divisant** n.
- En base 6, c'est donc possible de multiplier par 3 avec un AC, mais aussi de diviser par 2.
- $\Rightarrow$ Construction d'un AC simulant l'algorithme de Syracuse.

Quelques remarques finales...

- Modèle de calcul massivement parallèle.
- Règles locales très simples qui peuvent induire des comportements globaux complexes.
- Encore plus qu'en dimension 1, les automates cellulaires 2D restent encore très mal compris.
- Logiciel utilisé : **GOLLY**

  `http://golly.sourceforge.net/`

Quelques remarques finales...

- Modèle de calcul massivement parallèle.
- Règles locales très simples qui peuvent induire des comportements globaux complexes.
- Encore plus qu'en dimension 1, les automates cellulaires 2D restent encore très mal compris.
- Logiciel utilisé : **GOLLY**

  `http://golly.sourceforge.net/`

Merci !