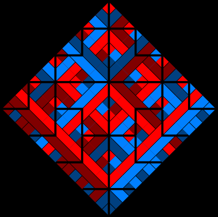
Programmation et indécidabilités dans les systèmes complexes

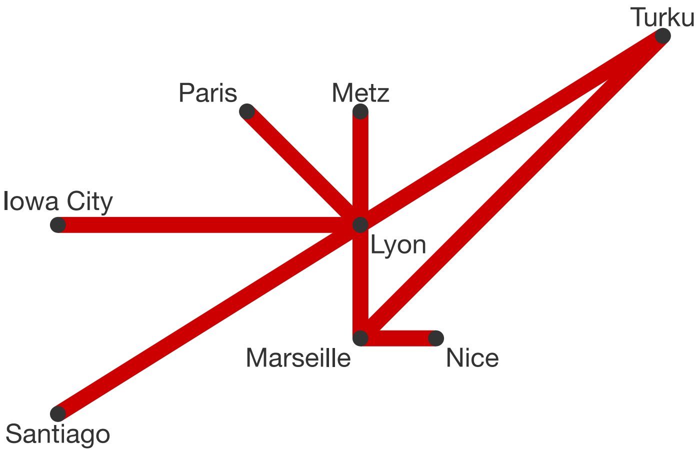
Nicolas Ollinger (Aix-Marseille Université & CNRS)

Habilitation à diriger des recherches
Université de Nice-Sophia Antipolis
3 décembre 2008



Il était une fois...

1998 MIM1 training period: **"Universality of 1D cellular automata"**



a 10 years trip from Metz to Nice (10h by train)

Complex systems

From well understood **local** entities...

Complex systems

...to complex **global** emerging behaviors

Buzzwords

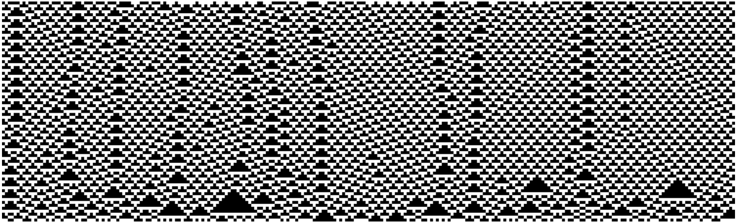
From cellular automata to complex systems
A homogenous collection of well understood entities with local interactions from which global complex behaviors emerge.

From universality to different forms of computation
Computing is all about moving and combining quanta of information.

Our researches focus on complexity and emergence in complex systems driven by computational processes. Deterministic computations can lead to unpredictable behaviors.

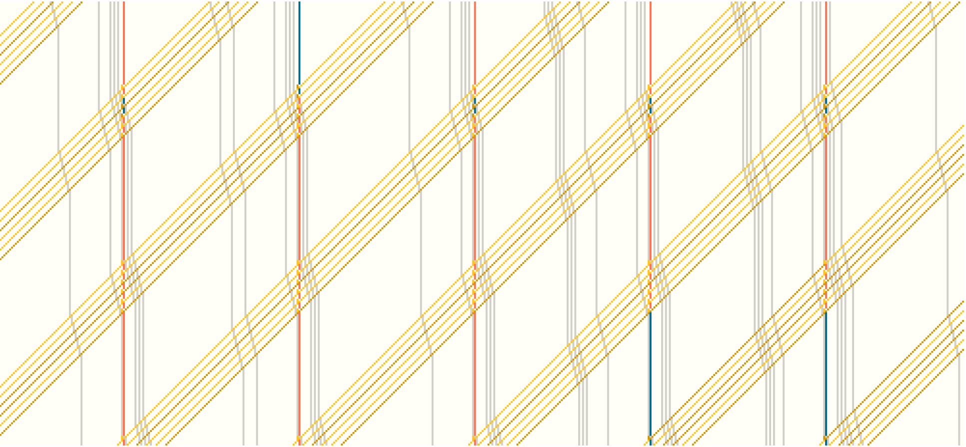
External programming

```
open Graphics

let w = 250 and h = 75

let c = Array.create w 0

let init () =
    let v = ref 1 in
    for i = 0 to w − 1 do
        c.(i) ← !v mod 2;
        v := (75 × !v) mod 65537
    done

let draw y =
    for x = 0 to w − 1 do
        if c.(x) ≡ 1 then plot x y
    done

let f x y z = (54 lsr (4 × x + 2 × y + z)) land 1

let next () =
    let l = ref c.(w − 1)
    and r = c.(0) in
    for i = 0 to w − 2 do
        let v = c.(i) in
        c.(i) ← f !l c.(i) c.(i + 1);
        l := v
    done;
    c.(w − 1) ← f !l c.(w − 1) r

let _ =
    open_graph (Printf.sprintf " %ix%i" w h);
    init ();
    for y = 0 to h − 1 do
        draw y;
        next ()
    done;
    read_key ()
```



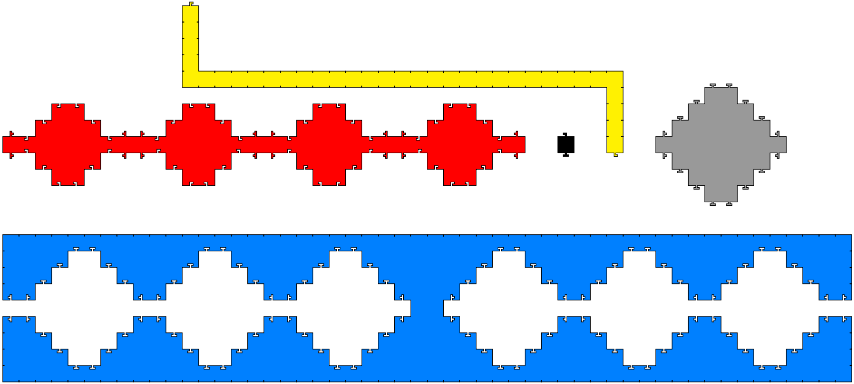
Algorithmic and programming for **simulation**, **visualization**, detection of special behaviors of complex systems

Internal programming



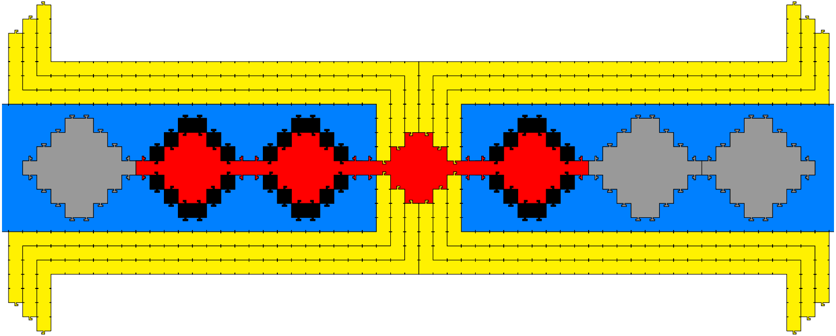
Algorithmic and programming **proper to the model**, to program desired behaviors

Programming by reduction



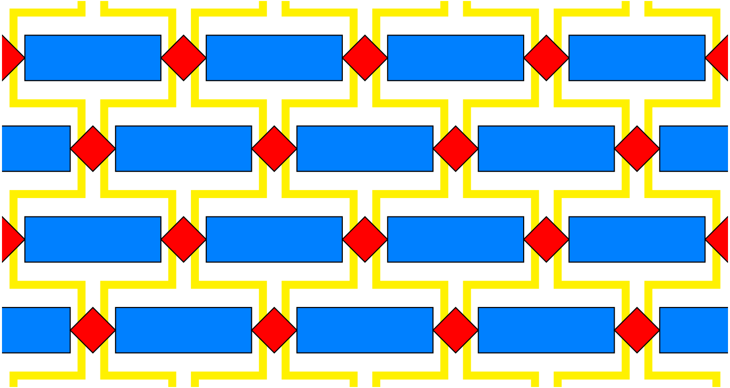
Recursive encoding of any object of a first family as an object of a second family **preserving given properties** to transfer some complexity result (ex. **undecidability**)

Programming by reduction



Recursive encoding of any object of a first family as an object of a second family **preserving given properties** to transfer some complexity result (ex. **undecidability**)

Programming by reduction



Recursive encoding of any object of a first family as an object of a second family **preserving given properties** to transfer some complexity result (ex. **undecidability**)

1. cellular automata, geometry and computation

Cellular automata

Simple discrete continuous uniform complex systems

Definition A **CA** is a triple (S, r, f) where S is a **finite set of states**, $r \in \mathbb{N}$ is the **radius** and $f : S^{2r+1} \longrightarrow S$ is the **local rule**.

A **configuration** $c \in S^{\mathbb{Z}}$ is a coloring of $\mathbb{Z}$ by S.

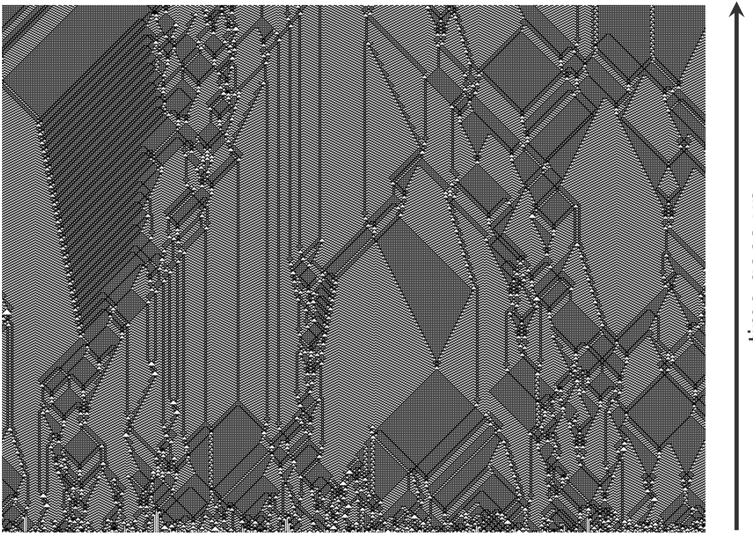
The **global map** $F : S^{\mathbb{Z}} \longrightarrow S^{\mathbb{Z}}$ applies f uniformly and locally:

$$\forall c \in S^{\mathbb{Z}}, \forall z \in \mathbb{Z}, \qquad F(c)(z) = f(c(z-r), \ldots, c(z+r)).$$

A **space-time diagram** $\Delta \in S^{\mathbb{N}\times\mathbb{Z}}$ satisfies, for all $t \in \mathbb{Z}^+$,

$$\Delta(t+1) = F(\Delta(t)).$$

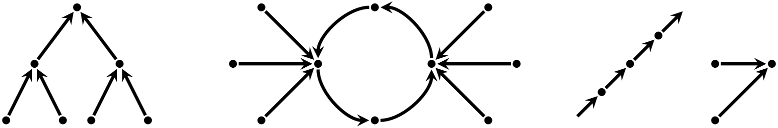
Space-time diagram



$S = \{0, 1, 2\}, r = 1, f(x, y, z) = \lfloor 6430564760289/3^{9x+3y+z} \rfloor \pmod 3$

Discrete dynamical systems

Definition A **DDS** is a pair (X, F) where X is a **topological space** and $F : X \longrightarrow X$ is a **continuous** map.



The **orbit** of $x \in X$ is the sequence $(F^n(x))$ obtained by iterating F.

In this talk, $X = S^{\mathbb{Z}}$ where S is a finite alphabet and X is endowed with the **Cantor topology** (product of the discrete topology on S), and F is a continuous map that **commutes with the shift map** σ: $F \circ \sigma = \sigma \circ F$ where $\sigma(x)(z) = x(z+1)$.

Hedlund-Richardson's theorem

For all $n \in \mathbb{N}$ and $u \in S^{2n+1}$, the **cylinder** $[u] \subseteq S^{\mathbb{Z}}$ is

$$[u] = \left\{ c \in S^{\mathbb{Z}} \middle| \forall i \in [-n, n]\ c(i) = u_{i+n} \right\} \quad .$$

The **clopen sets** are finite unions of cylinders.

Therefore in this topology **continuity** means **locality**.

Theorem [Hedlund 1969] The continuous maps commuting with the shift coincide with the global maps of cellular automata.

Cellular automata have a **dual nature**: topological maps with finite automata description.

Cellular Automata 101

Introduced by von Neumann at the end of the 40s, **cellular automata** have been extensively studied from different points of view.

As a rudimentary model for experimentation (self-reproduction, physical phenomena, biology, etc)

As a model of massive parallelism where specific programming techniques and algorithms where developped (FSSP, signals, etc)

As a discrete dynamical system to study deterministic chaos, sensitivity to initial conditions and other dynamical properties

As a simple kind of complex system cellular automata are considered for themselves as a playground to understand the **emergence of complexity**.

Wolfram's experimental classification

Wolfram (1984) First unformal classification

"[…] In **class 1**, the behavior is very simple, and almost all initial conditions lead to exactly the same uniform final state.

In **class 2**, there are many different possible final states, but all of them consist just of a certain set of simple structures that either remain the same forever or repeat every few steps.

In **class 3**, the behavior is more complicated, and seems in many respects random, although triangles and other small-scale structures are essentially always at some level seen.

And finally […] **class 4** involves a mixture of order and randomness: localized structures are produced which on their own are fairly simple, but these structures move around and interact with each other in very complicated ways. […] "

S. Wolfram [ANKOS, chapter 6, pp. 231–235]

Wolfram's experimental classification

Wolfram (1984) First unformal classification

Class 1. Nilpotency

In **class 2**, there are many different possible final states, but all of them consist just of a certain set of simple structures that either remain the same forever or repeat every few steps.

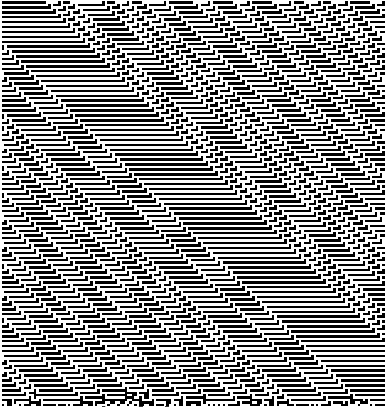
S. Wolfram [ANKOS, chapter 6, pp. 231–235]

Wolfram's experimental classification

Wolfram (1984) First unformal classification

Class 1. Nilpotency

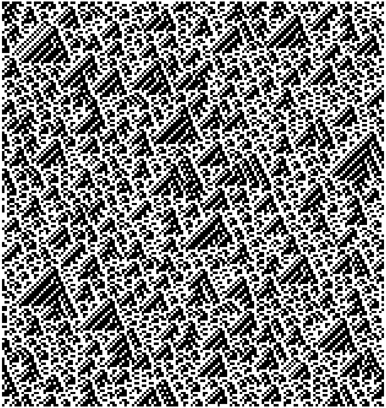
Class 2. Ult. Periodicity (up to shift)



S. Wolfram [ANKOS, chapter 6, pp. 231–235]

Wolfram's experimental classification

Wolfram (1984) First unformal classification



Class 3. Chaoticity

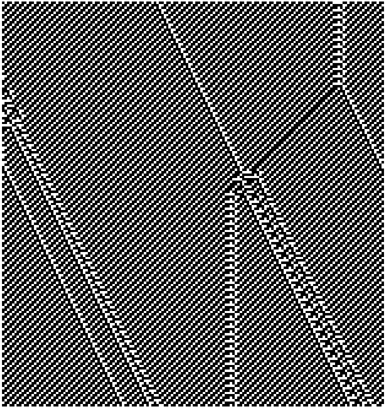
Numerous classifications and tools to understand deterministic chaos, cf **[Formenti 1998]**

And finally […] **class 4** involves a mixture of order and randomness: localized structures are produced which on their own are fairly simple, but these structures move around and interact with each other in very complicated ways. […] "

S. Wolfram [ANKOS, chapter 6, pp. 231–235]

Wolfram’s experimental classification

Wolfram (1984) First unformal classification



Class 3. Chaoticity

Numerous classifications and tools to understand deterministic chaos, cf **[Formenti 1998]**

Class 4. Complexity

Particles, collisions...

Quanta of information

Computation ?

S. Wolfram [ANKOS, chapter 6, pp. 231–235]

Towards a refined and formal classification

Our **first field of contribution** to the study of CA concerns formal **algebraic classifications** capturing algorithmic complexity.

Starting from the **grouping algebraic classification** of **[Mazoyer and Rapaport 1999]**, we extended it to capture universality and studied its structural properties in **[NO PhD 2002]**. The study was further developed in **[Theyssier PhD 2005]**.

A survey in two papers is in preparation **[U1,U2]**.

The sub-automaton relation

A CA $\mathcal{A}$ is **algorithmically simpler** than a CA $\mathcal{B}$ if all the space-time diagrams of $\mathcal{A}$ are space-time diagrams of $\mathcal{B}$.

Formally, $\mathcal{A} \subseteq \mathcal{B}$ if there exists $\varphi : S_{\mathcal{A}} \to S_{\mathcal{B}}$ injective such that

$$\overline{\varphi} \circ G_{\mathcal{A}} = G_{\mathcal{B}} \circ \overline{\varphi}$$

That is, the following diagram commutes:

$$\begin{tikzcd} C \arrow[r, "\varphi"] \arrow[d, "G_{\mathcal{A}}"'] & \overline{\varphi}(C) \arrow[d, "G_{\mathcal{B}}"] \\ G_{\mathcal{A}}(C) \arrow[r, "\varphi"'] & \overline{\varphi}(G_{\mathcal{A}}(C)) \end{tikzcd}$$

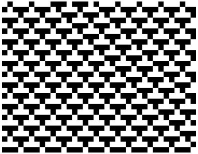
Remark Different elementary relations can be considered.

Bulking

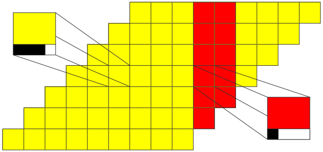
We quotient the set of CA by **discrete affine transformations**, the only geometrical transformations preserving CA.

The $\langle m, n, k\rangle$ transformation of $\mathcal{A}$ satisfies:

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



$\mathcal{A}$

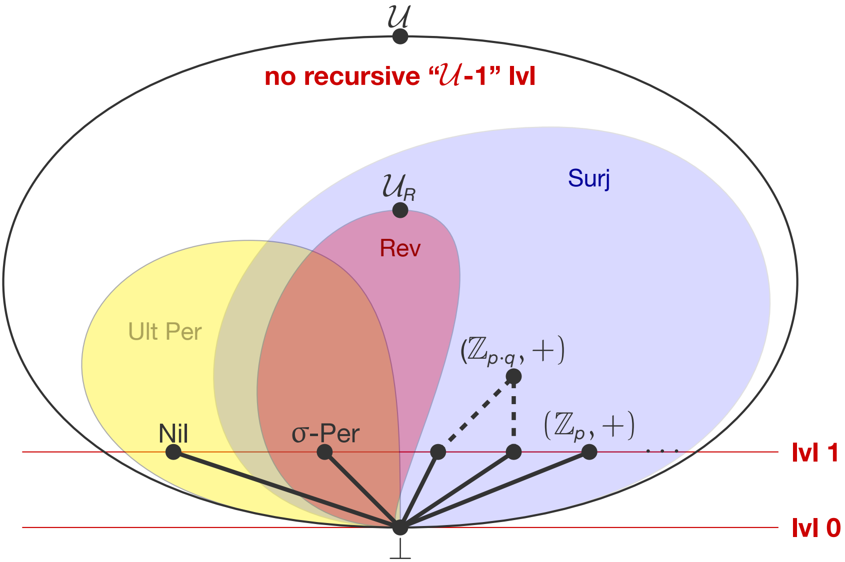


$\mathcal{A}^{\langle 4,4,1\rangle}$

The **bulking quasi-order** is defined by $\mathcal{A} \leqslant \mathcal{B}$ if there exists $\langle m, n, k\rangle$ and $\langle m', n', k'\rangle$ such that

$$\mathcal{A}^{\langle m,n,k\rangle} \subseteq \mathcal{B}^{\langle m',n',k'\rangle} \quad .$$

The big picture



Intrinsic universality

Our **second field of contribution** to the study of CA concerns **universalities**, more precisely **intrinsic universality**.

For decision problem, **creative sets** play the role of universal objects. A good definition of **universality** for Turing machines remains to be found.

Bulking provides a natural notion of **intrinsic universality**.

A CA $\mathcal{U}$ is **intrinsically universal** if it is maximal for $\leqslant$,
i.e. **[NO PhD 2002]** for all CA $\mathcal{A}$, there exists α such that $\mathcal{A} \subseteq \mathcal{U}^{\alpha}$.

Universalities

Theorem [U2] There exists **Turing universal** CA that are not intrinsically universal.

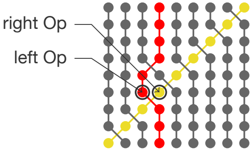
Theorem [U1] There exists no **real-time** intrinsically universal CA.

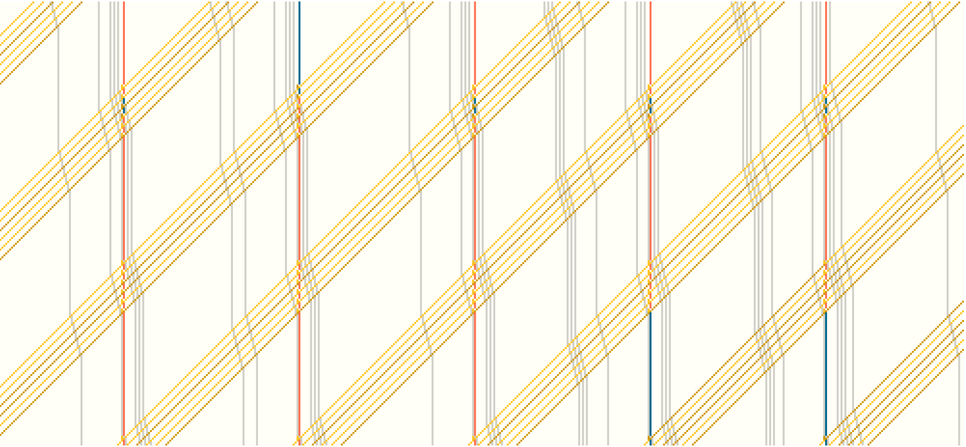
Theorem [NO STACS 2003] It is **undecidable**, given a CA to determine if it is intrinsically universal.

The proof proceeds by reduction of the nilpotency problem on spatially periodic configurations.

6 states

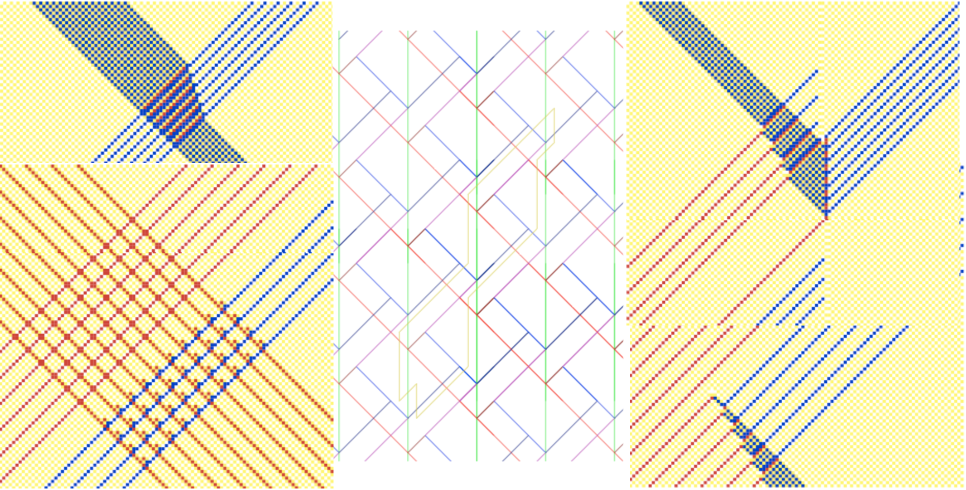
We constructed a **6 states** intrinsically universal CA of radius 1 embedding **boolean circuits** into the line **[NO ICALP 2002]**.





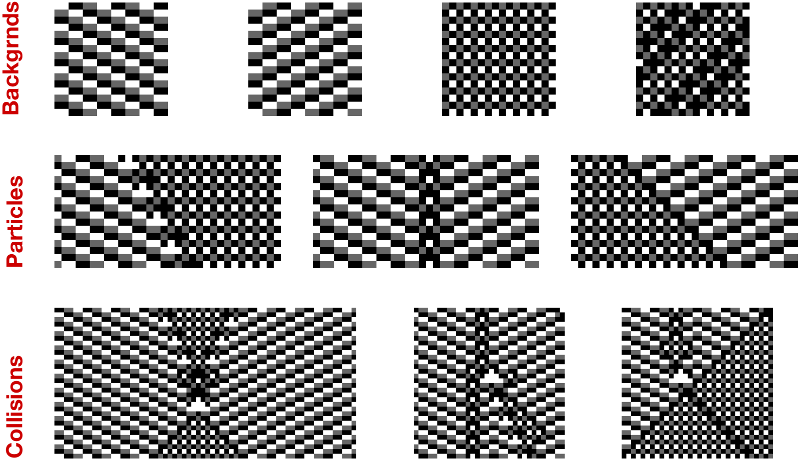
4 states

Using our framework for particles and collisions, this was improved to **4 states** by **arithmetical encoding** **[NO Richard CSP 2008]**.



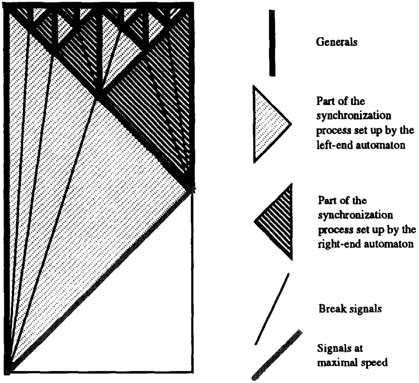
Back to class 4

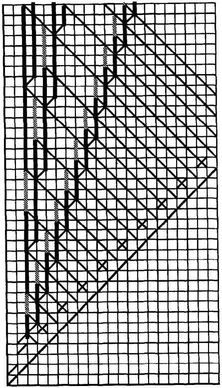
Our **third field of contribution** to the study of CA concerns the study of **backgrounds, particles and collisions**.



Algorithmic of CA

Algorithmic of CA makes heavy usage of **signals** and linear algebra **synchronization constraints** resolution.



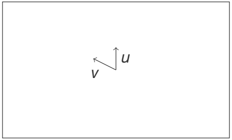


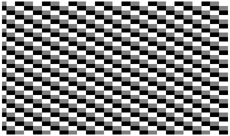
Pictures from Mazoyer 1996

Particles and collisions exhibit the characteristics of signals.

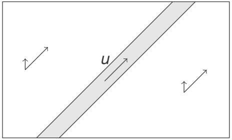
Map Automata

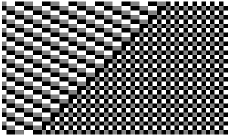
Backgrounds, particles and collisions are characterized as **regular colorings** produced by finite **counter-automata** painting the plane **[NO Richard TCS 2009]**.



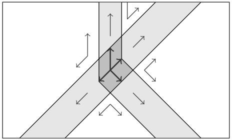


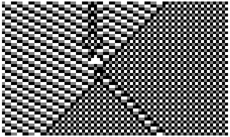
Bakgrounds are captured by **0-counter map automata**





Particles are captured by **1-counter map automata**

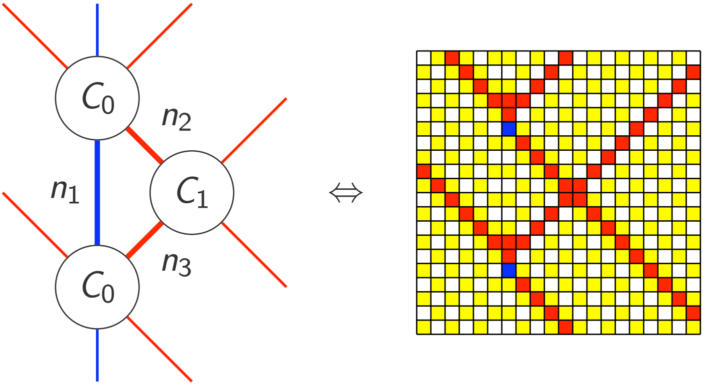




Collisions are captured by **aperiodic 2-counter map automata**

Computing with PaCo systems

Bindings of collisions can be manipulated as **catenation schemes**: planar maps with collisions as vertices and particles as edges. Valid catenation schemes can be recursively captured by **semi-linear sets** **[NO Richard IFIP-TCS 2008]**.



Advertisement

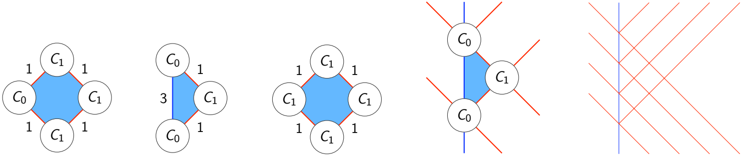
Systèmes de particules et collisions discrètes dans les automates cellulaires

Gaétan Richard

le **4 décembre 2008**, à 14h

au CMI, Château-Gombert, salle 001

Université de Provence (Aix-Marseille I)



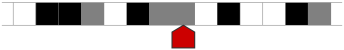
2. undecidability, machines and aperiodicity

Turing machines

A **TM** is a triple (S, Σ, T) where S is a finite set of states, Σ a finite alphabet and $T \subseteq (S \times \{\leftarrow, \rightarrow\} \times S) \cup (S \times \Sigma \times S \times \Sigma)$ is a set of instructions.

(s, δ, t) : *"in state s move according to δ and enter state t."*
(s, a, t, b) : *"in state s, reading letter a, write letter b and enter state t."*

A **configuration** $\mathfrak{c} \in S \times \Sigma^{\mathbb{Z}}$ is a coloring of $\mathbb{Z}$ by Σ plus a state of S, the state of the head looking at cell 0.



A **DTM** is a **TM** where at most one instruction can be applied from any configuration.

Partial DDS $(S \times \Sigma^{\mathbb{Z}}, G)$ where G is a partial continuous map.

Undecidability

By adding an **initial state** and encoding input words on **finite configurations**, classical problems on TM can be considered.

Theorem [Turing 1936] The **halting problem** for TM is undecidable.

Combine this result on **machines** with **many-one reductions** to establish undecidability results.

Many-one reduction $A \leqslant_m B$ if there exists a recursive φ such that for all x, $x \in A \leftrightarrow \varphi(x) \in B$

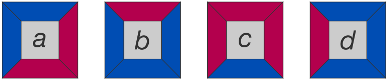
We study decidability of some properties of complex systems and establish $0'_m$-completeness results.

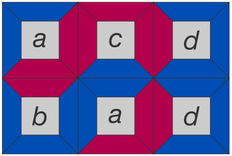
The Domino Problem (**DP**)

Our **first field of contribution** to the study of decidability of properties of complex systems concerns **decidability in tilings**.

"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)





Aperiodicity in **DP**

The set of tilings of a tile set T is a compact subset of $T^{\mathbb{Z}^2}$.

By compacity, if a tile set does not tile the plane, there exists a square of size $n \times n$ that cannot be tiled.

Tile sets without tilings are recursively enumerable.

A set of Wang tiles with a periodic tiling admits a biperiodic tiling.

Tile sets with a biperiodic tiling are recursively enumerable.

Undecidability is to be found in **aperiodic tile sets**, tile sets that only admit aperiodic tilings.

Theorem [Berger 1964] **DP** is undecidable.

Undecidability of **DP**

Composition technique [Robinson 1971, NO 2008] Define an unambiguous substitution, encode it with local constraints to obtain an aperiodic tile set. Modify the tile set to insert everywhere prefixes of unbounded length of TM computation.

Fixpoint technique [Durand, Romashchenko, Shen 2008] Define a tile set with prototiles enforcing tiling constraints using a Turing machine. A fixpoint tile set is aperiodic. Modify the tile set to insert everywhere prefixes of unbounded length of TM computation.

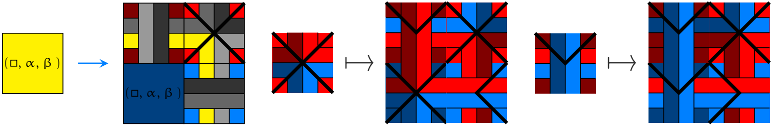
Transducer and sturmian words [Kari 2007] Consider lines of tilings as a transducer coding a relation on biinfinite words. Encode tuples of real numbers in a sturmian way, the transducer enforcing affine relations. Reduce the immortality problem of Turing machines to the immortality problem of affine maps.

104 tiles

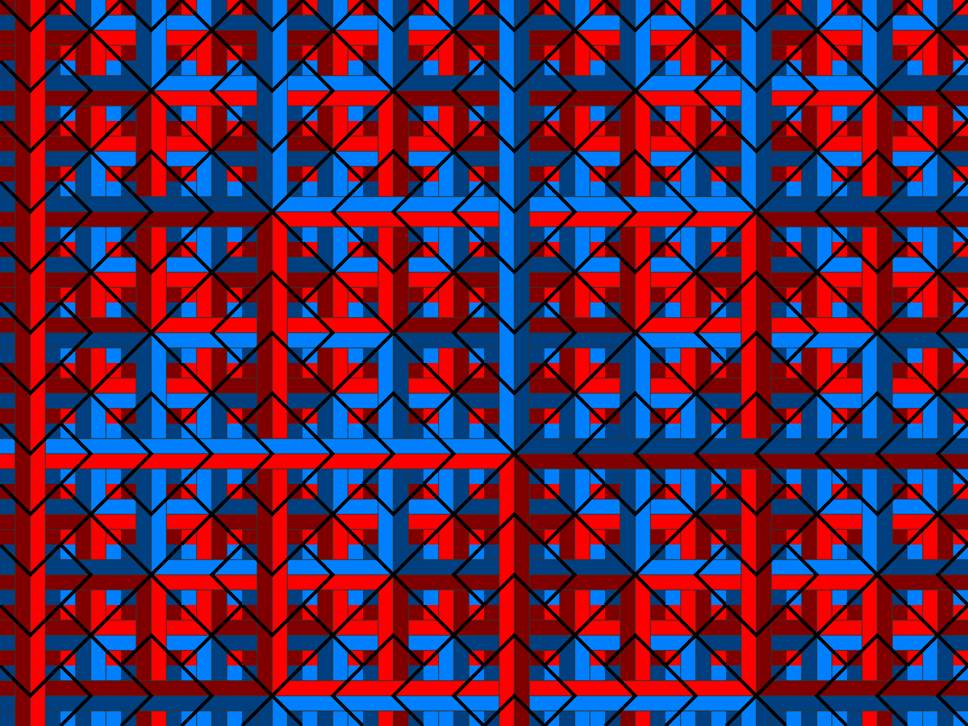
We factorized proof techniques from several authors into a convenient aperiodic set of 104 tiles **[NO CiE 2008]**.

(1) Every **unambiguous substitutions** has an aperiodic subshift.

(2) Enforce an unambiguous 2 × 2 substitution with **Wang tiles**.



(3) **Decorate** this tile set to encode any given 2 × 2 substitution.



Undecidability of the nilpotency problem

A tile set is **NW-deterministic** if, for each pair of colors, there exists at most one tile with these colors on N and W sides.

Theorem [Kari 1992] NW-deterministic **DP** is undecidable.

The **limit set** Λ_F of a CA F is the non-empty subshift $\Lambda_F = \bigcap_{n\in\mathbb{N}} F^n(S^{\mathbb{Z}})$ of configurations appearing in biinfinite space-time diagrams $\Delta \in S^{\mathbb{Z}\times\mathbb{Z}}$ such that $\forall t \in \mathbb{Z}, \Delta(t+1) = F(\Delta(t))$.

NW-deterministic **DP** reduces to **NP**.

Theorem [Kari 1992] NP is undecidable.

Variations permit to obtain numerous undecidability results on CA (Rice theorem on limit sets, **Intrinsic Universality**, etc).

The Immortality Problem (**IP**)

Our **second field of contribution** to the study of decidability of properties of complex systems concerns **mortality properties of various models**.

"(T_2) To find an effective method, which for every Turing-machine M decides whether or not, for all tapes I (finite and infinite) and all states B, M will eventually halt if started in state B on tape I" *(Büchi, 1962)*

A TM is **mortal** if all configurations are ultimately halting.

Aperiodicity in **IP**

As $S \times \Sigma^{\mathbb{Z}}$ is compact, G is continuous and the set of halting configurations is open, **mortality** implies **uniform mortality**.

Mortal TM are recursively enumerable.

TM with a periodic orbit are recursively enumerable.

Undecidability is to be found in **aperiodic TM**, TM whose infinite orbits are all aperiodic.

Undecidability of **IP**

Theorem [Hooper 1966] **IP** is undecidable.

Reduction reduce **HP** for 2-CM $(s, \underline{@}1^m x 2^n y)$

Problem unbounded searches produce immortal configurations.

Idea *by compacity, extract infinite failure sequence*

Hooper's trick use bounded searches with **recursive calls** to initial segments of the simulation of increasing sizes:

$\texttt{@}\underline{\texttt{@}}_{\texttt{s}}\texttt{xy1111111111x2222y}$ *recursive call*

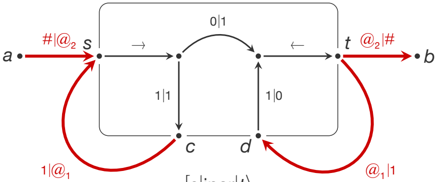
$\texttt{s}_0$

The TM is immortal iff the 2-CM halts from $(s_0, (0, 0))$.

Programming tips and tricks

We designed a TM programming language with **recursive calls**:

`http://www.lif.univ-mrs.fr/~nollinge/rec/gnirut/`



$[s|\mathrm{incr}|t\rangle$

```
fun [s|incr|t⟩ :
  s.  ⟶, r
  r. 0 ⊢ 1, b | 1 ⊢ 1, c
  call [c|incr|d⟩ from 1   ⇐ call 1
  d. 1 ⊢ 0, b
  b.  ⟵, t

call [a|incr|b⟩ from #   ⇐ call 2
```

Program it!

def $[s|\mathrm{search}_1|t_0, t_1, t_2\rangle$:
$s.\ \underline{@_\alpha} \vdash \underline{@_\alpha}, l$
$l.\ \rightarrow, u$
$u.\ \underline{x} \vdash \underline{x}, t_0$
$\mid \underline{1}x \vdash 1\underline{x}, t_1$
$\mid \underline{1}1x \vdash 11\underline{x}, t_2$
$\mid \underline{1}11 \vdash \underline{1}11, c$
call $[c|\mathrm{check}_1|p\rangle$ from 1
$p.\ \underline{1}11 \vdash 11\underline{1}, l$

def $[s|\mathrm{search}_2|t_0, t_1, t_2\rangle$:
$s.\ \underline{x} \vdash \underline{x}, l$
$l.\ \rightarrow, u$
$u.\ \underline{y} \vdash \underline{y}, t_0$
$\mid \underline{2}y \vdash 2\underline{y}, t_1$
$\mid \underline{2}2y \vdash 22\underline{y}, t_2$
$\mid \underline{2}22 \vdash \underline{2}22, c$
call $[c|\mathrm{check}_2|p\rangle$ from 2
$p.\ \underline{2}22 \vdash 22\underline{2}, l$

def $[s|\mathrm{test1}|z, p\rangle$:
$s.\ \underline{@_\alpha}x \vdash \underline{@_\alpha}x, z$
$\mid \underline{@_\alpha}1 \vdash \underline{@_\alpha}1, p$

def $[s|\mathrm{endtest2}|z, p\rangle$:
$s.\ \underline{x}y \vdash \underline{x}y, z$
$\mid \underline{x}2 \vdash \underline{x}2, p$

def $[s|\mathrm{test2}|z, p\rangle$:
$[s|\mathrm{search}_1|t_0, t_1, t_2\rangle$
$[t_0|\mathrm{endtest2}|z_0, p_0\rangle$
$[t_1|\mathrm{endtest2}|z_1, p_1\rangle$
$[t_2|\mathrm{endtest2}|z_2, p_2\rangle$
$\langle z_0, z_1, z_2|\mathrm{search}_1|z]$
$\langle p_0, p_1, p_2|\mathrm{search}_1|p]$

def $[s|\mathrm{mark}_1|t, co\rangle$:
$s.\ \underline{y}1 \vdash 2\underline{y}, t$
$\mid \underline{y}x \vdash \underline{y}x, co$

def $[s|\mathrm{endinc}_1|t, co\rangle$:
$[s|\mathrm{search}_2|r_0, r_1, r_2\rangle$
$[r_0|\mathrm{mark}_1|t_0, co_0\rangle$
$[r_1|\mathrm{mark}_1|t_1, co_1\rangle$
$[r_2|\mathrm{mark}_1|t_2, co_2\rangle$
$\langle t_2, t_0, t_1|\mathrm{search}_2|t]$
$\langle co_0, co_1, co_2|\mathrm{search}_2|co]$

def $[s|\mathrm{inc2}_1|t, co\rangle$:
$[s|\mathrm{search}_1|r_0, r_1, r_2\rangle$
$[r_0|\mathrm{endinc}_1|t_0, co_0\rangle$
$[r_1|\mathrm{endinc}_1|t_1, co_1\rangle$
$[r_2|\mathrm{endinc}_1|t_2, co_2\rangle$
$\langle t_0, t_1, t_2|\mathrm{search}_1|t]$
$\langle co_0, co_1, co_2|\mathrm{search}_1|co]$

def $[s|\mathrm{dec2}_1|t\rangle$:
$\langle s, co|\mathrm{inc2}_1|t]$

def $[s|\mathrm{mark}_2|t, co\rangle$:
$s.\ \underline{y}2 \vdash 2\underline{y}, t$
$\mid \underline{y}x \vdash \underline{y}x, co$

def $[s|\mathrm{endinc}_2|t, co\rangle$:
$[s|\mathrm{search}_2|r_0, r_1, r_2\rangle$
$[r_0|\mathrm{mark}_2|t_0, co_0\rangle$
$[r_1|\mathrm{mark}_2|t_1, co_1\rangle$
$[r_2|\mathrm{mark}_2|t_2, co_2\rangle$
$\langle t_2, t_0, t_1|\mathrm{search}_2|t]$
$\langle co_0, co_1, co_2|\mathrm{search}_2|co]$

def $[s|\mathrm{inc2}_2|t, co\rangle$:
$[s|\mathrm{search}_1|r_0, r_1, r_2\rangle$
$[r_0|\mathrm{endinc}_2|t_0, co_0\rangle$
$[r_1|\mathrm{endinc}_2|t_1, co_1\rangle$
$[r_2|\mathrm{endinc}_2|t_2, co_2\rangle$
$\langle t_0, t_1, t_2|\mathrm{search}_1|t]$
$\langle co_0, co_1, co_2|\mathrm{search}_1|co]$

def $[s|\mathrm{dec2}_2|t\rangle$:
$\langle s, co|\mathrm{inc2}_2|t]$

def $[s|\mathrm{pushinc}_1|t, co\rangle$:
$s.\ \underline{x}2 \vdash 1\underline{x}, c$
$\mid \underline{x}y1 \vdash 1x\underline{y}, pt$
$\mid \underline{x}yx \vdash 1\underline{y}x, pco$
$[c|\mathrm{endinc}_1|pt0, pco0\rangle$
$pt0.\ \rightarrow, t0$
$t0.\ 2 \vdash 2, pt$
$pt.\ \leftarrow, t$
$pco0.\ x \vdash 2, pco$
$pco.\ \leftarrow, zco$
$zco.\ 1 \vdash x, co$

def $[s|\mathrm{inc1}_1|t, co\rangle$:
$[s|\mathrm{search}_1|r_0, r_1, r_2\rangle$
$[r_0|\mathrm{pushinc}_1|t_0, co_0\rangle$
$[r_1|\mathrm{pushinc}_1|t_1, co_1\rangle$
$[r_2|\mathrm{pushinc}_1|t_2, co_2\rangle$
$\langle t_2, t_0, t_1|\mathrm{search}_1|t]$
$\langle co_0, co_1, co_2|\mathrm{search}_1|co]$

def $[s|\mathrm{dec1}_1|t\rangle$:
$\langle s, co|\mathrm{inc1}_1|t]$

def $[s|\mathrm{pushinc}_2|t, co\rangle$:
$s.\ \underline{x}2 \vdash 1\underline{x}, c$
$\mid \underline{x}y2 \vdash 1x\underline{y}, pt$
$\mid \underline{x}yy \vdash 1y\underline{y}, pco$
$[c|\mathrm{endinc}_2|pt0, pco0\rangle$
$pt0.\ \rightarrow, t0$
$t0.\ 2 \vdash 2, pt$
$pt.\ \leftarrow, t$
$pco0.\ x \vdash 2, pco$
$pco.\ \leftarrow, zco$
$zco.\ 1 \vdash x, co$

def $[s|\mathrm{inc1}_2|t, co\rangle$:
$[s|\mathrm{search}_1|r_0, r_1, r_2\rangle$
$[r_0|\mathrm{pushinc}_2|t_0, co_0\rangle$
$[r_1|\mathrm{pushinc}_2|t_1, co_1\rangle$
$[r_2|\mathrm{pushinc}_2|t_2, co_2\rangle$
$\langle t_2, t_0, t_1|\mathrm{search}_1|t]$
$\langle co_0, co_1, co_2|\mathrm{search}_1|co]$

def $[s|\mathrm{dec1}_2|t\rangle$:
$\langle s, co|\mathrm{inc1}_2|t]$

def $[s|\mathrm{init}_1|r\rangle$:
$s.\ \rightarrow, u$
$u.\ \underline{1}1 \vdash \underline{x}y, e$
$e.\ \leftarrow, r$

def $[s|\mathrm{RCM}_1|co_1, co_2\rangle$:
$[s|\mathrm{init}_1|s_0\rangle$
$[s_0|\mathrm{test1}|s_{1z}, n\rangle$
$[s_1|\mathrm{inc1}_1|s_2, co_1\rangle$
$[s_2|\mathrm{inc2}_1|s_3, co_2\rangle$
$[s_3|\mathrm{test1}|n', s_{1p}\rangle$
$\langle s_{1z}, s_{1p}|\mathrm{test1}|s_1]$

def $[s|\mathrm{init}_2|r\rangle$:
$s.\ \rightarrow, u$
$u.\ \underline{2}2 \vdash \underline{x}y, e$
$e.\ \leftarrow, r$

def $[s|\mathrm{RCM}_2|co_1, co_2\rangle$:
$[s|\mathrm{init}_2|s_0\rangle$
$[s_0|\mathrm{test1}|s_{1z}, n\rangle$
$[s_1|\mathrm{inc1}_2|s_2, co_1\rangle$
$[s_2|\mathrm{inc2}_2|s_3, co_2\rangle$
$[s_3|\mathrm{test1}|n', s_{1p}\rangle$
$\langle s_{1z}, s_{1p}|\mathrm{test1}|s_1]$

fun $[s|\mathrm{check}_1|t\rangle$:
$[s|\mathrm{RCM}_1|co_1, co_2, \ldots\rangle$
$\langle co_1, co_2, \ldots|\mathrm{RCM}_1|t]$

fun $[s|\mathrm{check}_2|t\rangle$:
$[s|\mathrm{RCM}_2|co_1, co_2, \ldots\rangle$
$\langle co_1, co_2, \ldots|\mathrm{RCM}_2|t]$

Undecidability of the periodicity problem

A TM is **reversible** if it is deterministic with a deterministic inverse.

Theorem [Kari NO MFCS 2008] reversible **IP** is undecidable.

This implies to prove Hooper's result again with more constraints (no easy reduction to the reversible case preserving mortality).

A CA is **periodic** if one of its iterates is the identity map.

Reversible **IP** reduces to **PP**.

Theorem [Kari NO MFCS 2008] PP is undecidable.

Variations might provide new undecidability results?

3. perspectives

Going further...

One selected technical question by topic:

Bulking Identify precise tools to prove negative simulation results (the general problem is undecidable).

Universality Is rule 110 (or 54) intrinsically universal?

Particules and collisions Characterize CA with emerging particles and collisions.

(Un)decidability Study the decidability of dynamical properties (positive expansivity, *etc*)