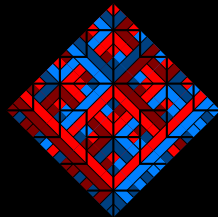


L'indécidable périodicité des automates cellulaires

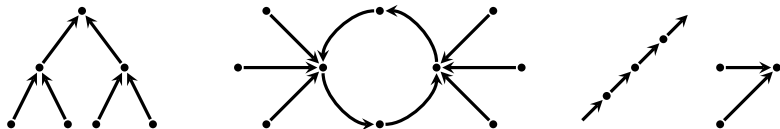
Nicolas Ollinger (LIF, Aix-Marseille Université, CNRS)

Séminaire LIMD — LAMA, 17 janvier 2010



Discrete dynamical systems

Definition A DDS is a pair (X, F) where X is a **topological space** and $F : X \rightarrow X$ 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 **commuting with the shift map** $\sigma : F \circ \sigma = \sigma \circ F$ where $\sigma(x)(z) = x(z + 1)$.

Two dynamical properties

We consider two simple dynamical properties (as opposed to more computational properties like reachability questions).

Definition A DDS (X, F) is **periodic** if for all $x \in X$ there exists $n \in \mathbb{N}$ such that $F^n(x) = x$.

Definition A DDS (X, F) is **nilpotent** if there exists $0 \in X$ such that for all $x \in X$ there exists $n \in \mathbb{N}$ such that $F^n(x) = 0$.

Question With a **proper recursive encoding** of the DDS, can we decide given a DDS if it is periodic? if it is nilpotent?

1. cellular automata

Cellular automata

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 $f : S^{2r+1} \rightarrow S$ is the **local rule** of the cellular automaton.

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



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

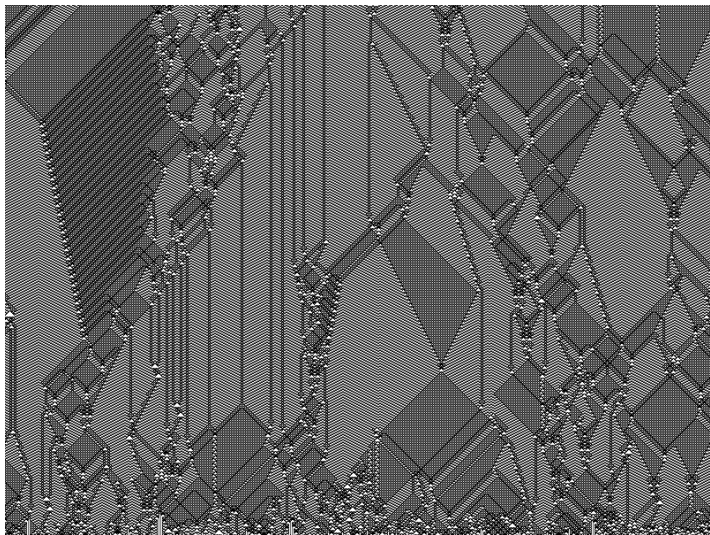
$$\forall c \in S^{\mathbb{Z}}, \forall z \in \mathbb{Z}, \quad F(c)(z) = f(c(z-r), \dots, c(z+r)).$$

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

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

The associated DDS is $(S^{\mathbb{Z}}, F)$.

Space-time diagram



time goes up

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

König's lemma

König's lemma Every infinite tree with finite branching admits an infinite path.

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

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

For all $C \subseteq S^{\mathbb{Z}}$, the **König tree** \mathcal{A}_C is the tree of cylinders intersecting C ordered by inclusion.

The **topping** $\overline{\mathcal{A}_C} \subseteq S^{\mathbb{Z}}$ of a König tree is the set of configurations tagging an infinite path from the root (intersection of the cylinders on the path).

Definition The **König topology** over $S^{\mathbb{Z}}$ is the topology whose closed sets are the toppings of König trees.

Curtis-Hedlund-Lyndon's theorem

König and Cantor topologies coincide: their open sets are unions of cylinders. Compactness arguments have combinatorial counterparts.

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.

Nilpotency

A CA is nilpotent iff there exists a **uniform bound** $n \in \mathbb{Z}^+$ such that F^n is a constant map.

Hint Take the bound of a **universal configuration** containing all words on S .

The Nilpotency Problem (NP)
given a CA decide if it is nilpotent.

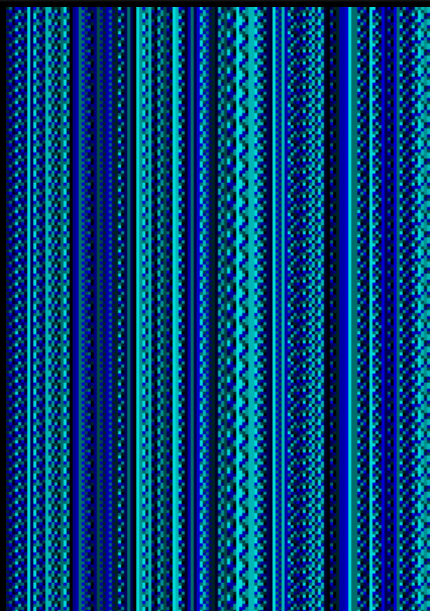


Periodicity

A CA is periodic iff there exists a **uniform period** $n \in \mathbb{Z}^+$ such that F^n is the identity map.

Hint Take the period of a **universal configuration** containing all words on S .

The Periodicity Problem (PP)
given a CA decide if it is periodic.



Undecidability of dynamical properties

Both **NP** and **PP** are **recursively undecidable**.

Undecidability is not necessarily a negative result:
it is a **hint of complexity**.

There exists non trivial nilpotent and periodic CA with a very large bound for quite simple CA (the bound grows faster than any recursive function).

To prove these results we inject computation into dynamics.

A direct reduction of the halting problem of Turing machines does not work.

Back to the nilpotency problem

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

A CA is nilpotent iff its limit set is a **singleton**.

A state $\perp \in S$ is **spreading** if $f(N) = \perp$ when $\perp \in N$.

A CA with a spreading state \perp is not nilpotent iff it admits a biinfinite space-time diagram without \perp .

A tiling problem Find a coloring $\Delta \in (S \setminus \{\perp\})^{\mathbb{Z}^2}$ satisfying the tiling constraints given by f .

Undecidability of the nilpotency problem

A classical undecidability result concerning tilings is the undecidability of the **domino problem (DP)**.

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

Here we need a restriction on the set of tilings.

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

NW-deterministic **DP** reduces to **NP** for spreading CA.

Theorem [Kari 1992] **NP** is undecidable.

Back to the periodicity problem

A periodic CA is **reversible**, which for CA is the same as **bijective** and even **injective**.

One can reduce the **periodicity problem** of **complete reversible Turing machines** to **PP**.

Immortality is the property of having at least one non-halting orbit.

One can reduce the **immortality problem** of reversible Turing machines without periodic orbit to the periodicity problem of complete reversible Turing machines.

Undecidability of the periodicity problem

A classical undecidability result concerning Turing machines is the **immortality problem (IP)**.

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

Here we need a restriction to reversible machines.

Theorem [Kari O 2008] **Reversible IP** is undecidable.

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

Theorem [Kari O 2008] **PP** is undecidable.

Revisiting classical results

For both **NP** and **PP**, we need a **stronger version** of a classical result, essentially a restriction on inputs.

The difficult part of the proofs hides into this task.

The **main difficulty** is to understand the dusty proofs.

Hopefully, we tend to reuse this for other variants.

Now, we will discuss the main ingredients for **PP**.

2. Immortality Problem (MFCS 2008)

The Immortality Problem (IP)

“(T₂) 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 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 .”

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

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.

In this talk

We investigate the **(un)decidability** of **dynamical properties** of three models of **reversible** computation.

We consider the behavior of the models starting from **arbitrary initial configurations**.

Immortality is the property of having at least one non-halting orbit.

Periodicity is the property of always eventually returning back to the starting configuration.

Models of reversible computation

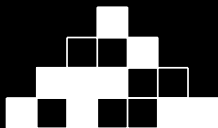
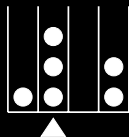
Counter Machines (**CM**)

Turing Machines (**TM**)

Cellular Automata (**CA**)

A machine is **deterministic** if there exists at most one transition from each configuration.

A machine is **reversible** if there exists **another machine** that can inverse **each step** of computation.



The immortality problem (IP)

A configuration on which F is undefined is **halting**.

A configuration is **mortal** if its orbit is eventually halting.

Halting Problem Given $S \in \mathcal{M}$, is $x_0 \in X$ mortal for S ?

S is **mortal** if all its configurations are mortal.

S is **uniformly mortal** if a uniform bound n exists such that F^n is halting for all configuration.

Immortality Problem Given $S \in \mathcal{M}$, is S immortal?

When X is compact and the set of halting configurations is open, uniform mortality is the same as mortality.

The periodicity problem (PP)

S is **complete** if F is total.

A configuration x is **n -periodic** if $F^n(x) = x$.

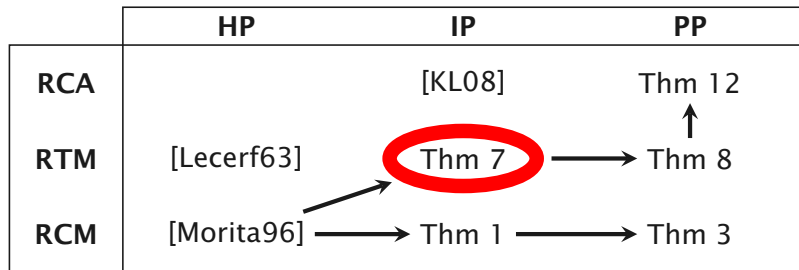
S is **periodic** if all its configurations are periodic.

S is **uniformly periodic** if a uniform bound n exists such that F^n is the identity map.

Periodicity Problem Given $S \in \mathcal{M}$, is S periodic?

When X is compact and the set of n -periodic configurations is open, uniform periodicity is the same as periodicity.

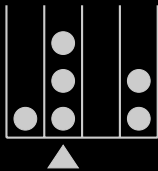
Results



→ denotes many-one reductions.

J. Kari and NO. *Periodicity and Immortality in Reversible Computing*.
Proceedings of **MFCS 2008**, LNCS 5162, pp. 419–430, 2008.

Longer version under revision at **JCSS**.



Reversible Counter Machines

A **k -CM** is a triple (S, k, T) where S is a finite set of states and $T \subseteq S \times \mathbb{Z}_k \times (\{Z, P\} \cup \{-, 0, +\}) \times S$ is a set of instructions.

$(s, u, i, t) \in T$: “in state s with counter i with value u ,
enter state t .”

$(s, \phi, i, t) \in T$: “in state s ,
apply ϕ to counter i and enter state t .”

DDS $(S \times \mathbb{N}^k, G)$ where $G(c)$ is the unique c' such that $c \vdash c'$.



[Minsky67] Every recursive function is computed by a 2-DCM and thus **HP** is undecidable for 2-DCM.



[Minsky67] Every recursive function is computed by a 2-DCM and thus **HP** is undecidable for 2-DCM.

[Hooper66] **IP** is undecidable for 2-DCM.

Idea for new proof Enforce infinite orbits to go through unbounded initial segments of an orbit from x_0 to reduce **HP**. ◇



[Minsky67] Every recursive function is computed by a 2-DCM and thus **HP** is undecidable for 2-DCM.

[Hooper66] **IP** is undecidable for 2-DCM.

Idea for new proof Enforce infinite orbits to go through unbounded initial segments of an orbit from x_0 to reduce **HP**. ◇

[Morita96] Every k -DCM is **simulated** by a 2-RCM.

Idea Encode a stack with two counters to keep an history of simulated instructions. ◇



[Minsky67] Every recursive function is computed by a 2-DCM and thus **HP** is undecidable for 2-DCM.

[Hooper66] **IP** is undecidable for 2-DCM.

Idea for new proof Enforce infinite orbits to go through unbounded initial segments of an orbit from x_0 to reduce **HP**. ◇

[Morita96] Every k -DCM is **simulated** by a 2-RCM.

Idea Encode a stack with two counters to keep an history of simulated instructions. ◇

Theorem 1 **IP** is undecidable for 2-RCM.

Idea Morita's simulation preserves immortality. ◇



Theorem 3 **PP** is undecidable for 2-RCM.

Idea Reduce **IP** to **PP**:

- (1) **IP** is still undecidable for 2-RCM with **mortal reverse**
(add a constantly incremented counter to the k -DCM)



Theorem 3 **PP** is undecidable for 2-RCM.

Idea Reduce **IP** to **PP**:

- (1) **IP** is still undecidable for 2-RCM with **mortal reverse**
(add a constantly incremented counter to the k -DCM)
- (2) Let $\mathcal{M} = (S, 2, T)$ be a 2-RCM with mortal reverse.
 \mathcal{M} admits no periodic orbit.
Let \mathcal{M}' be the 2-RCM with set of states $S \times \{+, -\}$
simulating \mathcal{M} on $+$ and \mathcal{M}^{-1} on $-$ and inverting polarity
on halting states.



Theorem 3 PP is undecidable for 2-RCM.

Idea Reduce IP to PP:

- (1) IP is still undecidable for 2-RCM with **mortal reverse**
(add a constantly incremented counter to the k -DCM)
- (2) Let $\mathcal{M} = (S, 2, T)$ be a 2-RCM with mortal reverse.
 \mathcal{M} admits no periodic orbit.
Let \mathcal{M}' be the 2-RCM with set of states $S \times \{+, -\}$
simulating \mathcal{M} on $+$ and \mathcal{M}^{-1} on $-$ and inverting polarity
on halting states.
- (3) \mathcal{M}' is periodic iff \mathcal{M} is mortal. ◇



Reversible 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 .”

DDS $(S \times \Sigma^{\mathbb{Z}}, G)$ where $G(c)$ is the unique c' such that $c \vdash c'$.



“(T₂) 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)



“(T₂) 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)

[Hooper66] IP is undecidable for DTM.

Idea TM with recursive calls! (we will discuss this)





“(T₂) 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)

[Hooper66] IP is undecidable for DTM.

Idea TM with recursive calls! (we will discuss this) ◇

[Lecerf63] Every DTM is **simulated** by a RTM.

Idea Keep history on a stack encoded on the tape. ◇



“(T₂) 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)

[Hooper66] IP is undecidable for DTM.

Idea TM with recursive calls! (we will discuss this)



[Lecerf63] Every DTM is **simulated** by a RTM.

Idea Keep history on a stack encoded on the tape.



Problem The simulation **does not** preserve immortality due to **unbounded searches**. We need to rewrite Hooper's proof for reversible machines.



Theorem 7 IP is undecidable for RTM.

Reduction reduce HP for 2-RCM $(s, @1^m \times 2^n y)$



Theorem 7 IP is undecidable for RTM.

Reduction reduce HP for 2-RCM $(s, @1^m \times 2^n y)$

Problem unbounded searches produce immortality.

Idea by compacity, extract infinite failure sequence



Theorem 7 IP is undecidable for RTM.

Reduction reduce HP for 2-RCM $(s, @1^m x 2^n y)$

Problem unbounded searches produce immortality.

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:

$\frac{@1111111111111111x2222y}{S}$ search $x \rightarrow$



Theorem 7 IP is undecidable for RTM.

Reduction reduce HP for 2-RCM ($s, @1^m x 2^n y$)

Problem unbounded searches produce immortality.

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:

$@\underset{\bar{s}_1}{1}1111111111111111x2222y$ *bounded search 1*



Theorem 7 IP is undecidable for RTM.

Reduction reduce HP for 2-RCM ($s, @1^m x 2^n y$)

Problem unbounded searches produce immortality.

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:

@11111111111111111x2222y *bounded search 2*
 s_2



Theorem 7 IP is undecidable for RTM.

Reduction reduce HP for 2-RCM ($s, @1^m x 2^n y$)

Problem unbounded searches produce immortality.

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:

@ $\underbrace{1111111111111111}_{S_3} x 2222y$ *bounded search 3*



Theorem 7 IP is undecidable for RTM.

Reduction reduce HP for 2-RCM $(s, @1^m x 2^n y)$

Problem unbounded searches produce immortality.

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:

@ s_0 **s_0** xy1111111111x2222y recursive call



Theorem 7 IP is undecidable for RTM.

Reduction reduce HP for 2-RCM $(s, @1^m x 2^n y)$

Problem unbounded searches produce immortality.

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:

@@_s1111x2222_yx2222y ultimately in case of collision...

_{s_c}



Theorem 7 IP is undecidable for RTM.

Reduction reduce HP for 2-RCM ($s, @1^m x 2^n y$)

Problem unbounded searches produce immortality.

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:

@ s **xy** 1111111111x2222y ...revert to clean
 S_b



Theorem 7 IP is undecidable for RTM.

Reduction reduce HP for 2-RCM ($s, @1^m x 2^n y$)

Problem unbounded searches produce immortality.

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:

@1111111111111111x2222y *pop and continue bounded search 1*
 \bar{s}_1



Theorem 7 IP is undecidable for RTM.

Reduction reduce HP for 2-RCM ($s, @1^m x 2^n y$)

Problem unbounded searches produce immortality.

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:

@11111111111111x2222y *bounded search 2*
 \bar{s}_2



Theorem 7 IP is undecidable for RTM.

Reduction reduce HP for 2-RCM ($s, @1^m x 2^n y$)

Problem unbounded searches produce immortality.

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:

@1111111111111x2222y *bounded search 3*
 \bar{S}_3



Theorem 7 IP is undecidable for RTM.

Reduction reduce HP for 2-RCM ($s, @1^m x 2^n y$)

Problem unbounded searches produce immortality.

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:

@111@ s_0 xy1111111x2222y recursive call



Theorem 7 IP is undecidable for RTM.

Reduction reduce HP for 2-RCM ($s, @1^m x 2^n y$)

Problem unbounded searches produce immortality.

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:

@111@ s_0 xy1111111x2222y recursive call



Theorem 7 IP is undecidable for RTM.

Reduction reduce HP for 2-RCM $(s, @1^m x 2^n y)$

Problem unbounded searches produce immortality.

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:

@111@ s_0 xy1111111x2222y recursive call

The RTM is immortal iff the 2-RCM is mortal on $(s_0, (0, 0))$.

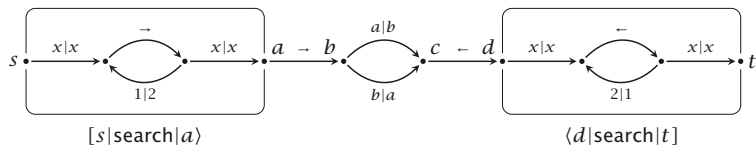
Programming tips and tricks (1/2)



We designed a TM programming language called Gnirut:

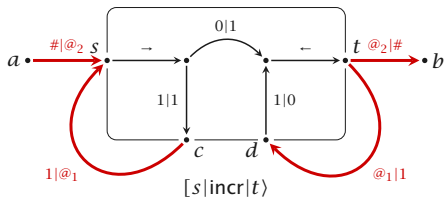
<http://www.lif.univ-mrs.fr/~nollinge/rec/gnirut/>

First ingredient use **macros** to avoid repetitions:



```
1 def [s|search|t] :      6 [s|search|a]
2   s. x ⊢ x, u          7 a. →, b
3   u. →, r              8 b. a ⊢ b, c | b ⊢ a, c
4   r. 1 ⊢ 2, u | x ⊢ x, t 9 c. ←, d
5                          10 <d|search|t]
```

Second ingredient use **recursive calls**:



```
1 fun [s|incr|t) :           8 call [a|incr|b) from # ← call 2
2   s. →, r
3   r. 0 ⊢ 1, b | 1 ⊢ 1, c
4   call [c|incr|d) from 1 ← call 1
5   d. 1 ⊢ 0, b
6   b. ←, t
7
```

Immortality: skeleton



$[s|\text{check}_1|t\rangle$ satisfies $s. \underline{a}_\alpha 1^m x \vdash \underline{a}_\alpha 1^m x, t$ or $s. \underline{a}_\alpha 1^\omega \uparrow$ or halt.

Immortality: skeleton



$[s|\text{check}_1|t\rangle$ satisfies $s. @_{\alpha}1^m \mathbf{x} \vdash @_{\alpha}1^m \mathbf{x}, t$ or $s. @_{\alpha}1^{\omega} \uparrow$ or halt.

$[s|\text{search}_1|t_0, t_1, t_2\rangle$ satisfies $s. @_{\alpha}1^m \mathbf{x} \vdash @_{\alpha}1^m \mathbf{x}, t_{m[3]}$ or ...



$[s|\text{check}_1|t\rangle$ satisfies $s. @_{\alpha}1^m x \vdash @_{\alpha}1^m x, t$ or $s. @_{\alpha}1^{\omega} \uparrow$ or halt.

$[s|\text{search}_1|t_0, t_1, t_2\rangle$ satisfies $s. @_{\alpha}1^m x \vdash @_{\alpha}1^m x, t_{m[3]}$ or ...

RCM ingredients:

testing counters

increment counter

decrement counter

$[s|\text{test1}|z, p\rangle$ and $[s|\text{test2}|z, p\rangle$

$[s|\text{inc1}|t, co\rangle$ and $[s|\text{inc2}|t, co\rangle$

$[s|\text{dec1}|t, co\rangle$ and $[s|\text{dec2}|t, co\rangle$



$[s|\text{check}_1|t\rangle$ satisfies $s. @_{\alpha}1^m \underline{x} \vdash @_{\alpha}1^m \underline{x}, t$ or $s. @_{\alpha}1^{\omega} \uparrow$ or halt.

$[s|\text{search}_1|t_0, t_1, t_2\rangle$ satisfies $s. @_{\alpha}1^m \underline{x} \vdash @_{\alpha}1^m \underline{x}, t_{m[3]}$ or ...

RCM ingredients:

testing counters

$[s|\text{test1}|z, p\rangle$ and $[s|\text{test2}|z, p\rangle$

increment counter

$[s|\text{inc1}|t, co\rangle$ and $[s|\text{inc2}|t, co\rangle$

decrement counter

$[s|\text{dec1}|t, co\rangle$ and $[s|\text{dec2}|t, co\rangle$

Simulator $[s|\text{RCM}_{\alpha}|co_1, co_2, \dots)$ initialize then compute

Immortality: skeleton



$[s|\text{check}_1|t\rangle$ satisfies $s. @_{\alpha}1^m x \vdash @_{\alpha}1^m x, t$ or $s. @_{\alpha}1^{\omega} \uparrow$ or halt.

$[s|\text{search}_1|t_0, t_1, t_2\rangle$ satisfies $s. @_{\alpha}1^m x \vdash @_{\alpha}1^m x, t_{m[3]}$ or ...

RCM ingredients:

testing counters

$[s|\text{test1}|z, p\rangle$ and $[s|\text{test2}|z, p\rangle$

increment counter

$[s|\text{inc1}|t, co\rangle$ and $[s|\text{inc2}|t, co\rangle$

decrement counter

$[s|\text{dec1}|t, co\rangle$ and $[s|\text{dec2}|t, co\rangle$

Simulator $[s|\text{RCM}_{\alpha}|co_1, co_2, \dots\rangle$ initialize then compute

$$[s|\text{check}_{\alpha}|t\rangle = [s|\text{RCM}_{\alpha}|co_1, co_2, \dots\rangle + \langle co_1, co_2, \dots|\text{RCM}_{\alpha}|s\rangle$$



```

1 def [s|search1|t0, t1, t2] :
2   s.  $\underline{\alpha}_x \vdash \underline{\alpha}_x, l$ 
3   l.  $\rightarrow, u$ 
4   u.  $\underline{x} \vdash \underline{x}, t_0$ 
5   |  $\underline{1x} \vdash \underline{1x}, t_1$ 
6   |  $\underline{11x} \vdash \underline{11x}, t_2$ 
7   |  $\underline{111} \vdash \underline{111}, c$ 
8   call [c|check1|p] from 1
9   p.  $\underline{111} \vdash \underline{111}, l$ 
10
11 def [s|search2|t0, t1, t2] :
12   s.  $\underline{x} \vdash \underline{x}, l$ 
13   l.  $\rightarrow, u$ 
14   u.  $\underline{y} \vdash \underline{y}, t_0$ 
15   |  $\underline{2y} \vdash \underline{2y}, t_1$ 
16   |  $\underline{22y} \vdash \underline{22y}, t_2$ 
17   |  $\underline{222} \vdash \underline{222}, c$ 
18   call [c|check2|p] from 2
19   p.  $\underline{222} \vdash \underline{222}, l$ 
20
21 def [s|test1|z, p] :
22   s.  $\underline{\alpha}_x \vdash \underline{\alpha}_x, z$ 
23   |  $\underline{\alpha}_x \vdash \underline{\alpha}_x, p$ 
24
25 def [s|endtest2|z, p] :
26   s.  $\underline{xy} \vdash \underline{xy}, z$ 
27   |  $\underline{x2} \vdash \underline{x2}, p$ 
28
29 def [s|test2|z, p] :
30   [s|search1|t0, t1, t2]
31   [t0|endtest2|z0, p0]
32   [t1|endtest2|z1, p1]
33   [t2|endtest2|z2, p2]
34   (z0, z1, z2|search1|z]
35   (p0, p1, p2|search1|p]
36
37 def [s|mark1|t, co] :
38   s.  $\underline{y1} \vdash \underline{2y}, t$ 
39   |  $\underline{yx} \vdash \underline{yx}, co$ 
40
41 def [s|endinc1|t, co] :
42   [s|search2|r0, r1, r2]
43   [r0|mark1|t0, co0]
44   [r1|mark1|t1, co1]
45   [r2|mark1|t2, co2]
46   (t2, t0, t1|search2|t]
47   (co0, co1, co2|search2|co]
48
49 def [s|inc21|t, co] :
50   [s|search1|r0, r1, r2]
51   [r0|endinc1|t0, co0]
52   [r1|endinc1|t1, co1]
53   [r2|endinc1|t2, co2]
54   (t0, t1, t2|search1|t]
55   (co0, co1, co2|search1|co]
56
57 def [s|dec21|t] :
58   (s, co|inc21|t]
59
60 def [s|mark2|t, co] :
61   s.  $\underline{y2} \vdash \underline{2y}, t$ 
62   |  $\underline{yx} \vdash \underline{yx}, co$ 
63
64 def [s|endinc2|t, co] :
65   [s|search2|r0, r1, r2]
66   [r0|mark2|t0, co0]
67   [r1|mark2|t1, co1]
68   [r2|mark2|t2, co2]
69   (t2, t0, t1|search2|t]
70   (co0, co1, co2|search2|co]
71
72 def [s|inc22|t, co] :
73   [s|search1|r0, r1, r2]
74   [r0|endinc2|t0, co0]
75   [r1|endinc2|t1, co1]
76   [r2|endinc2|t2, co2]
77   (t0, t1, t2|search1|t]
78   (co0, co1, co2|search1|co]
79
80 def [s|dec22|t] :
81   (s, co|inc22|t]
82
83 def [s|pushinc1|t, co] :
84   s.  $\underline{x2} \vdash \underline{1x}, c$ 
85   |  $\underline{xy1} \vdash \underline{1xy}, pt$ 
86   |  $\underline{yxy} \vdash \underline{1yx}, pco$ 
87   [c|endinc1|pt0, pco0]
88   pt0.  $\rightarrow, t0$ 
89   t0.  $2 \vdash 2, pt$ 
90   pt.  $\rightarrow, t$ 
91   pco0.  $x \vdash 2, pco$ 
92   pco.  $\rightarrow, zco$ 
93   zco.  $1 \vdash x, co$ 
94
95 def [s|inc11|t, co] :
96   [s|search1|r0, r1, r2]
97   [r0|pushinc1|t0, co0]
98   [r1|pushinc1|t1, co1]
99   [r2|pushinc1|t2, co2]
100  (t2, t0, t1|search1|t]
101  (co0, co1, co2|search1|co]
102
103 def [s|dec11|t] :
104  (s, co|inc11|t]
105
106 def [s|pushinc2|t, co] :
107  s.  $\underline{x2} \vdash \underline{1x}, c$ 
108  |  $\underline{xy2} \vdash \underline{1xy}, pt$ 
109  |  $\underline{xyy} \vdash \underline{1yy}, pco$ 
110  [c|endinc2|pt0, pco0]
111  pt0.  $\rightarrow, t0$ 
112  t0.  $2 \vdash 2, pt$ 
113  pt.  $\rightarrow, t$ 
114  pco0.  $x \vdash 2, pco$ 
115  pco.  $\rightarrow, zco$ 
116  zco.  $1 \vdash x, co$ 
117
118 def [s|inc12|t, co] :
119  [s|search1|r0, r1, r2]
120  [r0|pushinc2|t0, co0]
121  [r1|pushinc2|t1, co1]
122  [r2|pushinc2|t2, co2]
123  (t2, t0, t1|search1|t]
124  (co0, co1, co2|search1|co]
125
126 def [s|dec12|t] :
127  (s, co|inc12|t]
128
129 def [s|init1|r] :
130  s.  $\rightarrow, u$ 
131  u.  $\underline{11} \vdash \underline{xy}, e$ 
132  e.  $\rightarrow, r$ 
133
134 def [s|RCM1|co1, co2] :
135  [s|init1|s0]
136  [s0|test1|s1z, n]
137  [s1|inc11|s2, co1]
138  [s2|inc21|s3, co2]
139  [s3|test1|n', s1p]
140  [s1z, s1p|test1|s1]
141
142 def [s|init2|r] :
143  s.  $\rightarrow, u$ 
144  u.  $\underline{22} \vdash \underline{xy}, e$ 
145  e.  $\rightarrow, r$ 
146
147 def [s|RCM2|co1, co2] :
148  [s|init2|s0]
149  [s0|test1|s1z, n]
150  [s1|inc12|s2, co1]
151  [s2|inc22|s3, co2]
152  [s3|test1|n', s1p]
153  [s1z, s1p|test1|s1]
154
155 fun [s|check1|t] :
156  [s|RCM1|co1, co2, ...]
157  (co1, co2, ...|RCM1|t]
158
159 fun [s|check2|t] :
160  [s|RCM2|co1, co2, ...]
161  (co1, co2, ...|RCM2|t]

```



Theorem 8 **PP** is undecidable for RTM.

Idea Reduce **IP** to **PP**:

- (1) **IP** is still undecidable for RTM **without periodic orbit**.



Theorem 8 **PP** is undecidable for RTM.

Idea Reduce **IP** to **PP**:

- (1) **IP** is still undecidable for RTM **without periodic orbit**.
- (2) Let $\mathcal{M} = (S, \Sigma, T)$ be a RTM without periodic orbit
Let \mathcal{M}' be the complete RTM with set of states $S \times \{+, -\}$
simulating \mathcal{M} on $+$ and \mathcal{M}^{-1} on $-$ and inverting polarity
on halting states.

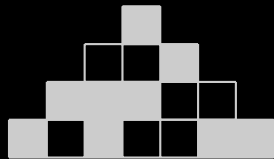


Theorem 8 **PP** is undecidable for RTM.

Idea Reduce **IP** to **PP**:

- (1) **IP** is still undecidable for RTM **without periodic orbit**.
- (2) Let $\mathcal{M} = (S, \Sigma, T)$ be a RTM without periodic orbit
Let \mathcal{M}' be the complete RTM with set of states $S \times \{+, -\}$
simulating \mathcal{M} on $+$ and \mathcal{M}^{-1} on $-$ and inverting polarity
on halting states.
- (3) \mathcal{M}' is periodic iff \mathcal{M} is mortal. ◇

Reversible Cellular Automata



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

DDS $(S^{\mathbb{Z}}, G)$ where $\forall z \in \mathbb{Z}, G(c)(z) = f(c(z-r), \dots, c(z+r))$



Theorem 12 **PP** is undecidable for RCA.

Idea Reduce **PP** for RTM to **PP** for RCA:

(1) **PP** is still undecidable for **complete** RTM.



Theorem 12 **PP** is undecidable for RCA.

Idea Reduce **PP** for RTM to **PP** for RCA:

(1) **PP** is still undecidable for **complete** RTM.

(2) Let $\mathcal{M} = (S, \Sigma, T)$ be a complete RTM

Let $(S', 2, f)$ be the RCA with set of states

$\Sigma \times (S \times \{+, -\} \cup \{\leftarrow, \rightarrow\})$ simulating \mathcal{M} on $+$ and \mathcal{M}^{-1} on $-$ on two levels.



Theorem 12 **PP** is undecidable for RCA.

Idea Reduce **PP** for RTM to **PP** for RCA:

- (1) **PP** is still undecidable for **complete** RTM.
- (2) Let $\mathcal{M} = (S, \Sigma, T)$ be a complete RTM
Let $(S', 2, f)$ be the RCA with set of states
 $\Sigma \times (S \times \{+, -\} \cup \{\leftarrow, \rightarrow\})$ simulating \mathcal{M} on $+$ and \mathcal{M}^{-1}
on $-$ on two levels.
- (3) In case of local inconsistency, invert polarity.



Theorem 12 **PP** is undecidable for RCA.

Idea Reduce **PP** for RTM to **PP** for RCA:

- (1) **PP** is still undecidable for **complete** RTM.
- (2) Let $\mathcal{M} = (S, \Sigma, T)$ be a complete RTM
Let $(S', 2, f)$ be the RCA with set of states
 $\Sigma \times (S \times \{+, -\} \cup \{\leftarrow, \rightarrow\})$ simulating \mathcal{M} on $+$ and \mathcal{M}^{-1}
on $-$ on two levels.
- (3) In case of local inconsistency, invert polarity.
- (4) The RCA is periodic iff \mathcal{M} is periodic. ◇



Open Problems with conjectures

Conjecture 1 It is undecidable whether a given complete 2-RCM admits a periodic configuration. (*proven if you remove complete or replace 2 by 3*)

Conjecture 2 There exists a complete RTM without a periodic configuration. (*known for DTM [BCN02]*)

→ **solved by J. Cassaigne**

Conjecture 3 It is undecidable whether a given complete RTM admits a periodic configuration. (*known for DTM [BCN02]*)