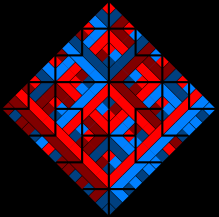
Periodicity and Immortality in Reversible Computing

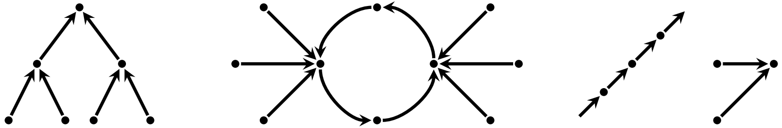
Nicolas Ollinger (LIF, Aix-Marseille Université, CNRS)

Séminaire SSS — **LORIA**, 30 mars 2009



Discrete dynamical systems

Definition A **DDS** is a pair (X, F) where X is a **topological space** and $F : X \to X$ 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** σ: $F \circ \sigma = \sigma \circ F$ 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 $f : S^{2r+1} \to S$ 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** $F : S^{\mathbb{Z}} \to S^{\mathbb{Z}}$ applies f uniformly and locally:

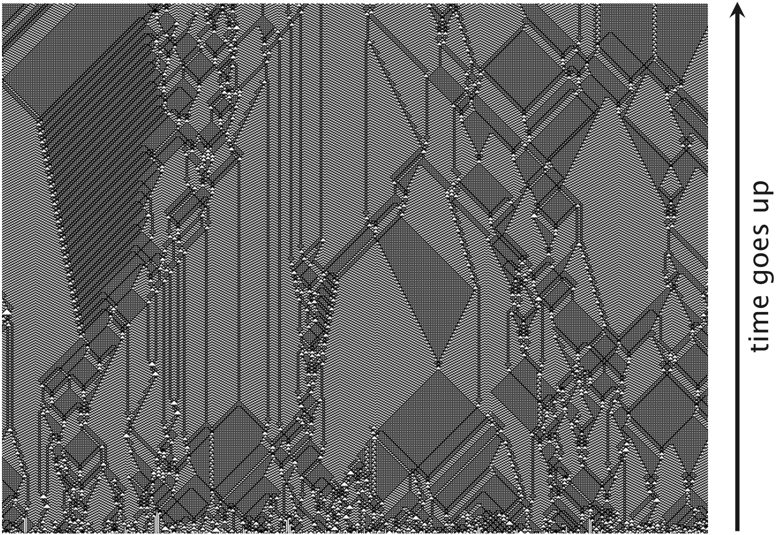
$$\forall c \in S^{\mathbb{Z}}, \forall z \in \mathbb{Z}, \qquad F(c)(z) = f(c(z-r), \ldots, c(z+r)).$$

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

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

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

Space-time diagram



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

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

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

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 close sets are the toppings of König trees.

Curtis-Hedlund-Lyndon's theorem

König and Cantor topologies coincide: their open sets are unions of cylinders. Compacity 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.

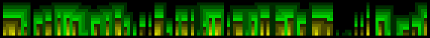
Cellular automata have a dual nature : topological maps with finite automata description.

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 Probem (NP) given a CA decide if it is nilpotent.

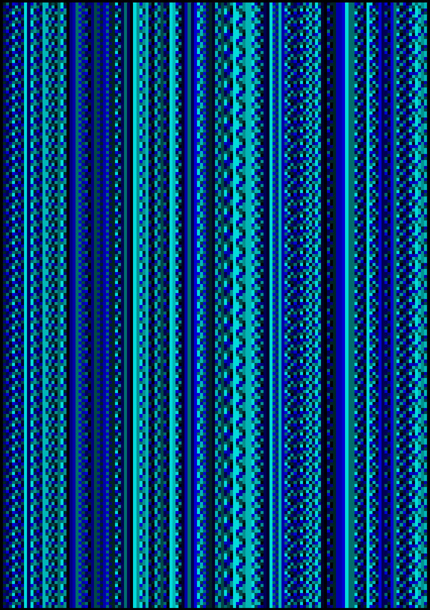


Periodicity

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 Probem (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 trival 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)

"(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 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.

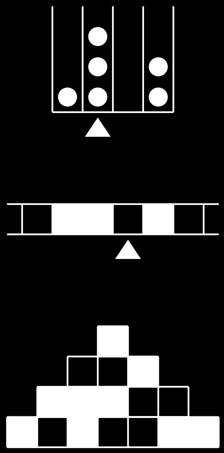
Periodicity is the property of always eventually returning back to the starting configuration.

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

$\mathcal{S}$ is **mortal** if all its configurations are mortal.

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

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

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

The periodicity problem (PP)

$\mathcal{S}$ is **complete** if F is total.

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

$\mathcal{S}$ is **periodic** if all its configurations are periodic.

$\mathcal{S}$ is **uniformly periodic** if a uniform bound n exists such that F^n is the identity map.

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

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

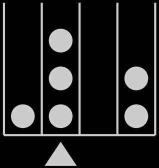
Results

	HP	IP	PP
RCA		[KL08]	Thm 12
RTM	[Lecerf63]	Thm 7	Thm 8
RCM	[Morita96]	Thm 1	Thm 3

→ 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 $G(\mathfrak{c})$ is the unique $\mathfrak{c}'$ such that $\mathfrak{c} \vdash \mathfrak{c}'$.

Immortality

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

Immortality

[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**.* $\Diamond$

Immortality

[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****.* $\Diamond$

[Morita96] Every k-DCM is **simulated** by a 2-RCM.
Idea *Encode a stack with two counters to keep an history of simulated instructions.* $\Diamond$

Immortality

[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**.* $\Diamond$

[Morita96] Every k-DCM is **simulated** by a 2-RCM.
Idea *Encode a stack with two counters to keep an history of simulated instructions.* $\Diamond$

Theorem 1 **IP** is undecidable for 2-RCM.
Idea *Morita's simulation preserves immortality.* $\Diamond$

Periodicity

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)

Periodicity

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 inversing polarity on halting states.

Periodicity

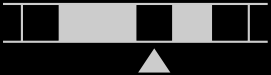
Theorem 3 **PP** is undecidable for 2-RCM.

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 inversing polarity on halting states.

(3) $\mathcal{M}'$ is periodic iff $\mathcal{M}$ is mortal. $\Diamond$



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 $G(\mathfrak{c})$ is the unique $\mathfrak{c}'$ such that $\mathfrak{c} \vdash \mathfrak{c}'$.

Immortality: a first attempt

"(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)*

Immortality: a first attempt

"(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)*

[Hooper66] IP is undecidable for DTM.
Idea *TM with recursive calls! (we will discuss this)* ◊

Immortality: a first attempt

"(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)*

[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.* ◊

Immortality: a first attempt

“(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)*

[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.

Immortality: simulating RCM

Theorem 7 **IP** is undecidable for RTM.

Reduction reduce **HP** for 2-RCM $(s, \underline{@}1^m x2^n y)$

Immortality: simulating RCM

Theorem 7 **IP** is undecidable for RTM.

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

Problem unbounded searches produce immortality.
Idea *by compacity, extract infinite failure sequence*

Immortality: simulating RCM

Theorem 7 **IP** is undecidable for RTM.

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

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{s}{\underline{@}}$1111111111111x2222y *search x →*

Immortality: simulating RCM

Theorem 7 **IP** is undecidable for RTM.

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

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:

$@\underset{s_1'}{\underline{1}}1111111111x2222y$ *bounded search 1*

Immortality: simulating RCM

Theorem 7 **IP** is undecidable for RTM.

Reduction reduce **HP** for 2-RCM $(s, \underline{@}1^m x2^n y)$

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:

$@1\underset{s_2'}{\underline{1}}11111111111x2222y$ *bounded search 2*

Immortality: simulating RCM

Theorem 7 **IP** is undecidable for RTM.

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

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:

$@11\underset{\overline{s}_3'}{\underline{1}}1111111111x2222y$ *bounded search 3*

Immortality: simulating RCM

Theorem 7 **IP** is undecidable for RTM.

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

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:

$@\underline{\mathbf{@}}_{\mathbf{s}}\mathbf{xy}1111111111x2222y$ *recursive call*
$\mathbf{s_0}$

Immortality: simulating RCM

Theorem 7 **IP** is undecidable for RTM.

Reduction reduce **HP** for 2-RCM $(s, \underline{@}1^m x2^n y)$

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$11111x22222y**x2222y $\quad$ *ultimately in case of collision...*

s_c

Immortality: simulating RCM

Theorem 7 **IP** is undecidable for RTM.

Reduction reduce **HP** for 2-RCM $(s, \underline{@}1^m x2^n y)$

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:

@**$\underline{@}_s$xy**1111111111x2222y *...revert to clean*
s_b

Immortality: simulating RCM

Theorem 7 **IP** is undecidable for RTM.

Reduction reduce **HP** for 2-RCM $(s, \underline{@}1^m x2^n y)$

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:

$@111\underset{\overline{s}'_1}{\underline{1}}111111111x2222y$ *pop and continue bounded search 1*

Immortality: simulating RCM

Theorem 7 **IP** is undecidable for RTM.

Reduction reduce **HP** for 2-RCM $(s, \underline{@}1^m x2^n y)$

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:

$@1111\underset{\overline{s}_2'}{\underline{1}}11111111x2222y$ *bounded search 2*

Immortality: simulating RCM

Theorem 7 **IP** is undecidable for RTM.

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

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:

@11111$\underline{1}$1111111x2222y *bounded search 3*
$\overline{s}_3'$

Immortality: simulating RCM

Theorem 7 **IP** is undecidable for RTM.

Reduction reduce **HP** for 2-RCM $(s, \underline{@}1^m x2^n y)$

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:

$@111\underset{s_0}{\underline{@}}_s\mathbf{xy}1111111x2222y$ *recursive call*

Immortality: simulating RCM

Theorem 7 **IP** is undecidable for RTM.

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

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:

$@111\underset{s_0}{\underline{@}_s}\mathbf{xy}1111111x2222y$ *recursive call*

Immortality: simulating RCM

Theorem 7 **IP** is undecidable for RTM.

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

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$\underline{@}_{s}$**xy**1111111x2222y *recursive call*
s_0

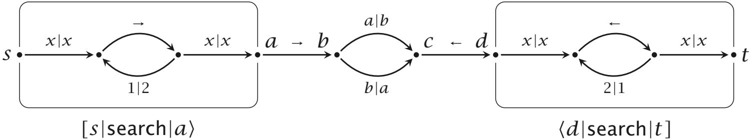
The RTM is immortal iff the 2-RCM is mortal on $(s_0, (0, 0))$.

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:

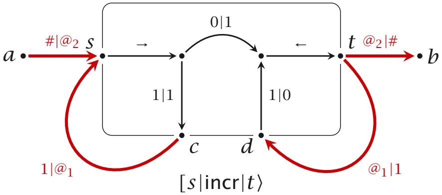



```
def [s|search|t⟩ :
   s. x ⊢ x, u
   u.  →, r
   r. 1 ⊢ 2, u | x ⊢ x, t

[s|search|a⟩
a.  →, b
b. a ⊢ b, c | b ⊢ a, c
c.  ←, d
⟨d|search|t]
```

Programming tips and tricks (2/2)

Second ingredient use **recursive calls**:



fun $[s|\mathrm{incr}|t\rangle$:
  $s.\ \rightarrow, r$
  $r.\ 0 \vdash 1, b \mid 1 \vdash 1, c$
  call $[c|\mathrm{incr}|d\rangle$ from 1 $\Leftarrow$ **call 1**
  $d.\ 1 \vdash 0, b$
  $b.\ \leftarrow, t$

call $[a|\mathrm{incr}|b\rangle$ from # $\Leftarrow$ **call 2**

Immortality: skeleton

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

Immortality: skeleton

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

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

Immortality: skeleton

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

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

RCM ingredients:

testing counters	$[s\,|\,\mathrm{test1}\,|\,z, p\rangle$ and $[s\,|\,\mathrm{test2}\,|\,z, p\rangle$
increment counter	$[s|\mathrm{inc1}|t, co\rangle$ and $[s|\mathrm{inc2}|t, co\rangle$
decrement counter	$[s|\mathrm{dec1}|t, co\rangle$ and $[s|\mathrm{dec2}|t, co\rangle$

Immortality: skeleton

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

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

RCM ingredients:

testing counters	$[s\,|\,\mathrm{test1}\,|\,z,p\rangle$ and $[s\,|\,\mathrm{test2}\,|\,z,p\rangle$
increment counter	$[s|\mathrm{inc1}|t, co\rangle$ and $[s|\mathrm{inc2}|t, co\rangle$
decrement counter	$[s|\mathrm{dec1}|t, co\rangle$ and $[s|\mathrm{dec2}|t, co\rangle$

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

Immortality: skeleton

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

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

RCM ingredients:

testing counters	$[s|\mathrm{test1}|z, p\rangle$ and $[s|\mathrm{test2}|z, p\rangle$
increment counter	$[s|\mathrm{inc1}|t, co\rangle$ and $[s|\mathrm{inc2}|t, co\rangle$
decrement counter	$[s|\mathrm{dec1}|t, co\rangle$ and $[s|\mathrm{dec2}|t, co\rangle$

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

$$[s|\mathrm{check}_\alpha|t\rangle = [s|\mathrm{RCM}_\alpha|co_1, co_2, \ldots\rangle + \langle co_1, co_2, \ldots|\mathrm{RCM}_\alpha|s]$$

```
def [s|search_1|t_0, t_1, t_2⟩ :
  s. @_α ⊢ @_α, l
  l. →, u
  u. x ⊢ x, t_0
   | 1x ⊢ 1x, t_1
   | 11x ⊢ 11x, t_2
   | 111 ⊢ 111, c
  call [c|check_1|p⟩ from 1
  p. 111 ⊢ 111, l

def [s|search_2|t_0, t_1, t_2⟩ :
  s. x ⊢ x, l
  l. →, u
  u. y ⊢ y, t_0
   | 2y ⊢ 2y, t_1
   | 22y ⊢ 22y, t_2
   | 222 ⊢ 222, c
  call [c|check_2|p⟩ from 2
  p. 222 ⊢ 222, l

def [s|test1|z, p⟩ :
  s. @_α x ⊢ @_α x, z
   | @_α 1 ⊢ @_α 1, p

def [s|endtest2|z, p⟩ :
  s. xy ⊢ xy, z
   | x2 ⊢ x2, p

def [s|test2|z, p⟩ :
  [s|search_1|t_0, t_1, t_2⟩
  [t_0|endtest2|z_0, p_0⟩
  [t_1|endtest2|z_1, p_1⟩
  [t_2|endtest2|z_2, p_2⟩
  ⟨z_0, z_1, z_2|search_1|z]
  ⟨p_0, p_1, p_2|search_1|p]

def [s|mark_1|t, co⟩ :
  s. y1 ⊢ 2y, t
   | yx ⊢ yx, co

def [s|endinc_1|t, co⟩ :
  [s|search_2|r_0, r_1, r_2⟩
  [r_0|mark_1|t_0, co_0⟩
  [r_1|mark_1|t_1, co_1⟩
  [r_2|mark_1|t_2, co_2⟩
  ⟨t_2, t_0, t_1|search_2|t]
  ⟨co_0, co_1, co_2|search_2|co]

def [s|inc2_1|t, co⟩ :
  [s|search_1|r_0, r_1, r_2⟩
  [r_0|endinc_1|t_0, co_0⟩
  [r_1|endinc_1|t_1, co_1⟩
  [r_2|endinc_1|t_2, co_2⟩
  ⟨t_0, t_1, t_2|search_1|t]
  ⟨co_0, co_1, co_2|search_1|co]

def [s|dec2_1|t⟩ :
  ⟨s, co|inc2_1|t]

def [s|mark_2|t, co⟩ :
  s. y2 ⊢ 2y, t
   | yx ⊢ yx, co

def [s|endinc_2|t, co⟩ :
  [s|search_2|r_0, r_1, r_2⟩
  [r_0|mark_2|t_0, co_0⟩
  [r_1|mark_2|t_1, co_1⟩
  [r_2|mark_2|t_2, co_2⟩
  ⟨t_2, t_0, t_1|search_2|t]
  ⟨co_0, co_1, co_2|search_2|co]

def [s|inc2_2|t, co⟩ :
  [s|search_1|r_0, r_1, r_2⟩
  [r_0|endinc_2|t_0, co_0⟩
  [r_1|endinc_2|t_1, co_1⟩
  [r_2|endinc_2|t_2, co_2⟩
  ⟨t_0, t_1, t_2|search_1|t]
  ⟨co_0, co_1, co_2|search_1|co]

def [s|dec2_2|t⟩ :
  ⟨s, co|inc2_2|t]

def [s|pushinc_1|t, co⟩ :
  s. x2 ⊢ 1x, c
   | xy1 ⊢ 1xy, pt
   | xyx ⊢ 1yx, pco
  [c|endinc_1|pt0, pco0⟩
  pt0. →, t0
  t0. 2 ⊢ 2, pt
  pt. ←, t
  pco0. x ⊢ 2, pco
  pco. ←, zco
  zco. 1 ⊢ x, co

def [s|inc1_1|t, co⟩ :
  [s|search_1|r_0, r_1, r_2⟩
  [r_0|pushinc_1|t_0, co_0⟩
  [r_1|pushinc_1|t_1, co_1⟩
  [r_2|pushinc_1|t_2, co_2⟩
  ⟨t_2, t_0, t_1|search_1|t]
  ⟨co_0, co_1, co_2|search_1|co]

def [s|dec1_1|t⟩ :
  ⟨s, co|inc1_1|t]

def [s|pushinc_2|t, co⟩ :
  s. x2 ⊢ 1x, c
   | xy2 ⊢ 1xy, pt
   | xyy ⊢ 1yy, pco
  [c|endinc_2|pt0, pco0⟩
  pt0. →, t0
  t0. 2 ⊢ 2, pt
  pt. ←, t
  pco0. x ⊢ 2, pco
  pco. ←, zco
  zco. 1 ⊢ x, co

def [s|inc1_2|t, co⟩ :
  [s|search_1|r_0, r_1, r_2⟩
  [r_0|pushinc_2|t_0, co_0⟩
  [r_1|pushinc_2|t_1, co_1⟩
  [r_2|pushinc_2|t_2, co_2⟩
  ⟨t_2, t_0, t_1|search_1|t]
  ⟨co_0, co_1, co_2|search_1|co]

def [s|dec1_2|t⟩ :
  ⟨s, co|inc1_2|t]

def [s|init_1|r⟩ :
  s. →, u
  u. 11 ⊢ xy, e
  e. ←, r

def [s|RCM_1|co_1, co_2⟩ :
  [s|init_1|s_0⟩
  [s_0|test1|s_1z, n⟩
  [s_1|inc1_1|s_2, co_1⟩
  [s_2|inc2_1|s_3, co_2⟩
  [s_3|test1|n', s_1p⟩
  ⟨s_1z, s_1p|test1|s_1]

def [s|init_2|r⟩ :
  s. →, u
  u. 22 ⊢ xy, e
  e. ←, r

def [s|RCM_2|co_1, co_2⟩ :
  [s|init_2|s_0⟩
  [s_0|test1|s_1z, n⟩
  [s_1|inc1_2|s_2, co_1⟩
  [s_2|inc2_2|s_3, co_2⟩
  [s_3|test1|n', s_1p⟩
  ⟨s_1z, s_1p|test1|s_1]

fun [s|check_1|t⟩ :
  [s|RCM_1|co_1, co_2, ...⟩
  ⟨co_1, co_2, ...|RCM_1|t]

fun [s|check_2|t⟩ :
  [s|RCM_2|co_1, co_2, ...⟩
  ⟨co_1, co_2, ...|RCM_2|t]
```

Periodicity

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

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

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

Periodicity

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 inversing polarity on halting states.

Periodicity

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

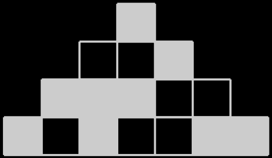
Idea Reduce **IP** to **PP**:

(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 inversing polarity on halting states.

(3) $\mathcal{M}'$ is periodic iff $\mathcal{M}$ is mortal. $\Diamond$

Reversible Cellular Automata



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

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

Periodicity

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

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

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

Periodicity

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.

Periodicity

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.

Periodicity

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])*