The Periodicity Problem of Cellular Automata

Julien Cassaigne (I2M, CNRS, Marseille)
Anahí Gajardo (CMM, Univ. de Chile, Concepción, Chile)
Jarkko Kari (Univ. of Turku, Turku, Finland)
Nicolas Ollinger (LIFO, Univ. Orléans, France)
Rodrigo Torres (CMM, Univ. de Chile, Concepción, Chile)

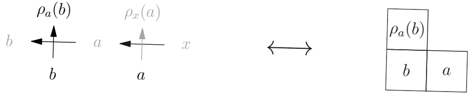
MealyM, Paris — July 12th, 2017

LIFO

From Mealy to OCA

Reset Mealy automata

OCA



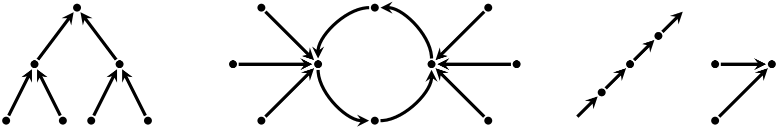
Proposition

The group generated by the reset automaton (X, X, ρ, Id) is finite if and only if the OCA $(X, f : (b, a) \mapsto \rho_a(b))$ is periodic.

- In the general case of CA, the periodicity problem is undecidable (Kari-Ollinger[2008]), the proof uses an encoding of some class of Turing machines and cannot be adapted to OCA.
- In our case, Boyle and Maass [2000] have given a decidable characterization of a subclass of the periodic ones (when every two-letter word is a wall). But the general question is still open.

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.



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

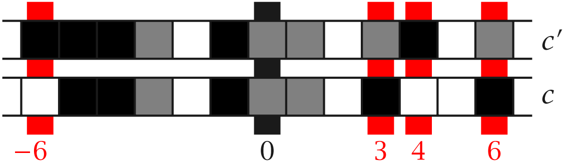
In this talk, $X = S^{\mathbb{Z}}$ is endowed with the **Cantor topology** (product of the discrete topology on S), and F is a continuous map **invariant by translation**.

Cantor topology

Definition The **Cantor topology** on $S^{\mathbb{Z}}$ is the product topology over $\mathbb{Z}$ of the discrete topology on S.

Remark The Cantor topology is **metric** and **compact**.

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



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

Definition A **subshift** is a non-empty set both topologically closed and closed by translation.

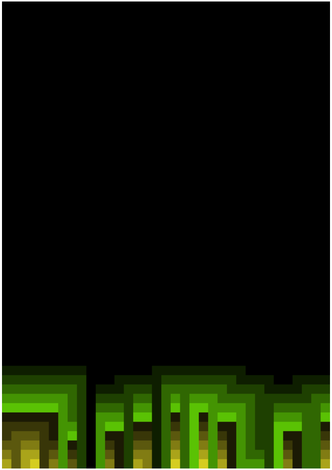
The nilpotency problem (Nil)

Definition A DDS is **nilpotent** if
$\exists z \in X, \forall x \in X, \exists n \in \mathbb{N}, F^n(x) = z.$

Given a recursive encoding of the DDS, can we **decide** nilpotency?

A DDS is **uniformly nilpotent** if
$\exists z \in X, \exists n \in \mathbb{N}, \forall x \in X, F^n(x) = z.$

Given a recursive encoding of the DDS, can we **bound recursively** n?



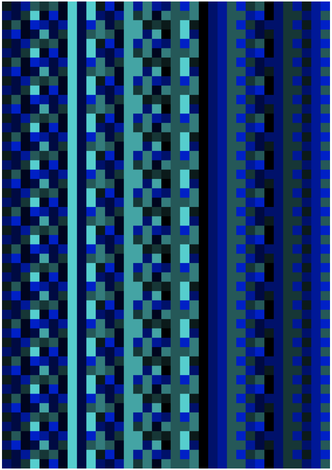
The periodicity problem (Per)

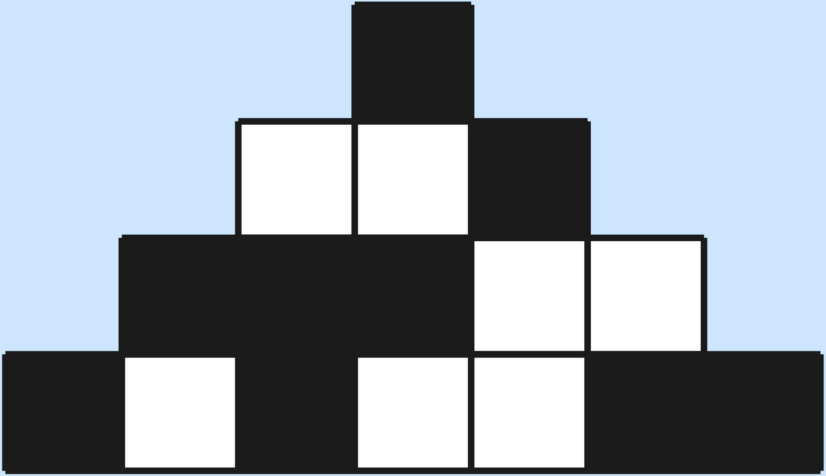
Definition A DDS is **periodic** if
$\forall x \in X, \exists n \in \mathbb{N}, F^n(x) = x$.

Given a recursive encoding of the DDS, can we **decide** periodicity?

A DDS is **uniformly periodic** if
$\exists n \in \mathbb{N}, \forall x \in X, F^n(x) = x$.

Given a recursive encoding of the DDS, can we **bound recursively** n?





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 $t \in \mathbb{Z}^+$,

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

Space-time diagram

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

Turing universality

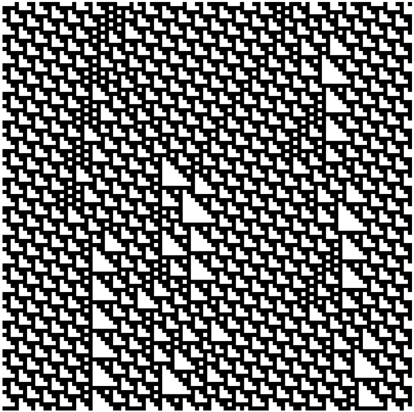
Theorem There exists **Turing-universal** CA.

à la Smith III

. . .BBBBabaabBBBBB. . .
s

$(S \cup \Sigma, 2, f)$

à la Cook (rule 110)



Curtis-Hedlund-Lyndon's theorem

$$[m] = \left\{ c \in S^{\mathbb{Z}} \;\middle|\; \forall p \in \mathbb{Z}, |p| \leqslant r \Rightarrow c(p) = m(p) \right\}$$

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

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

Theorem [Hedlund69] Cellular automata coincide with continuous maps invariant by translation.

Undecidability results

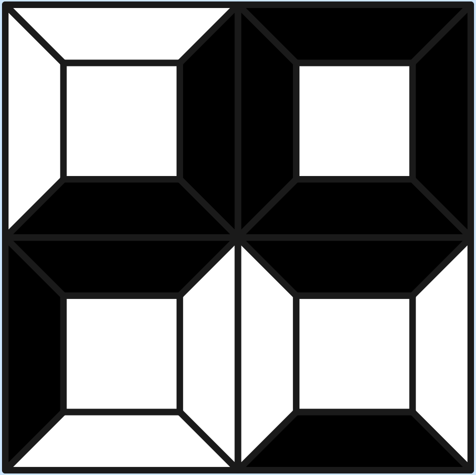
Theorem Both **Nil** and **Per** are **recursively undecidable**.

The proofs inject **computation** into **dynamics**.

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

Remark Due to **universe configurations** both nilpotency and periodicity are uniform.

The bounds grow **faster than any recursive function**: there exists simple nilpotent or periodic CA with huge bounds.

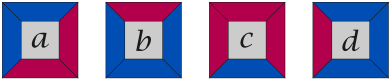


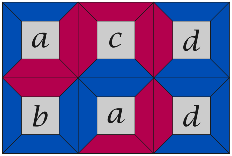
2. Nilpotency and tilings

The Domino Problem (DP)

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

(Wang, 1961)





Undecidability of DP

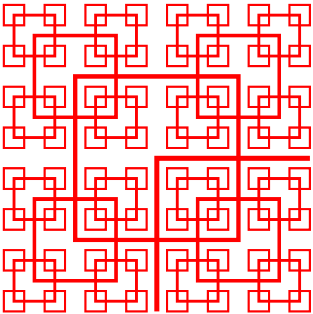
Theorem[Berger64] **DP** is **recursively undecidable**.

Remark To prove it one needs **aperiodic** tile sets.

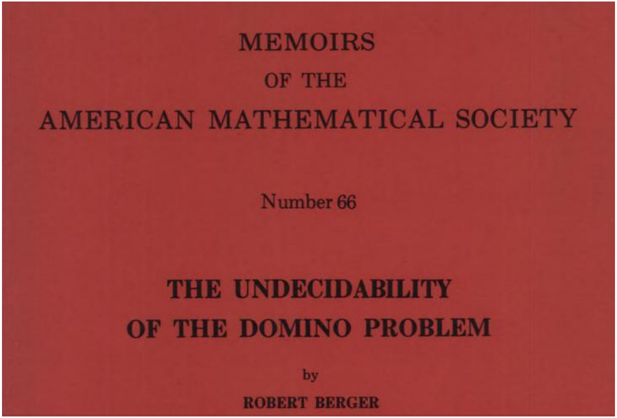
Idea of the proof

Enforce an (aperiodic) **self-similar structure** using local rules.

Insert a **Turing machine** computation **everywhere** using the structure.



Remark Plenty of different proofs!


MEMOIRS

OF THE

AMERICAN MATHEMATICAL SOCIETY

Number 66

THE UNDECIDABILITY
OF THE DOMINO PROBLEM

by

ROBERT BERGER

“(. . .) In 1966 R. Berger discovered the first aperiodic tile set. It contains 20,426 Wang tiles, (. . .)
Berger himself managed to reduce the number of tiles to 104 and he described these in his thesis, though they were omitted from the published version (Berger [1966]). (. . .)”

[GrSh, p.584]

THE UNDECIDABILITY OF THE DOMINO PROBLEM

A thesis presented

by

Robert Berger

to

The Division of Engineering and Applied Physics

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in the subject of

Applied Mathematics

Harvard University

Cambridge, Massachusetts

July 1964

APPENDIX II

A SIMPLER SOLVABLE DOMINO SET WITH NO TORUS

The skeleton set, K, analyzed in PART 3, is a solvable domino set with no torus. Since it is designed to serve also as a base set for modeling of Turing machines, it is not surprising that simpler solvable, torus-less domino sets exist. One such set, call it Q, is specified by Tables 9-12. The first three tables show the base, skeleton, and parity prototypes of Q. Although these tables show symbols in the center of domino edges, the base, skeleton, and parity channels should be thought of as distinct. Table 12 serves the same function for Q as did Table 4 for K, namely that of specifying which products of prototypes are permitted. However, since Q is a fairly small set, it is not too cumbersome to enumerate only those dominoes which are actually used in solutions of Q, 104 in all. (No concerted attempt has been made to find the smallest solvable torus-less domino set.)

Figure 24 shows, separately, skeleton signals and parity signals in the same portion of a solution of Q. If Figure 24 is rotated one-eighth turn clockwise, its skeleton signals bear a strong resemblance to the CD-signals of K.

A person who understands the skeleton set should have no trouble convincing himself of the likelihood that all solutions of Q look line extensions of Figure 24. The following hints will help.

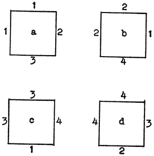
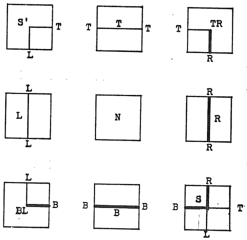


Table 9

Base Prototypes of Q

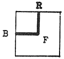


Note:

Use of same line weight for a horizontal and a (different) vertical signal introduces no ambiguity.

Table 10

Skeleton Prototypes of Q



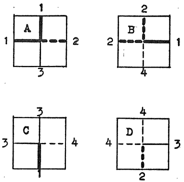
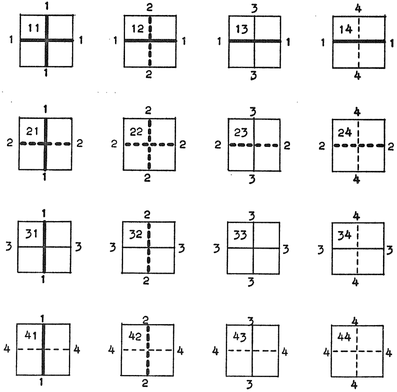


Table 11

Parity Prototypes of Q



Base	Skeleton	A	B	C	D	11	12	13	14	21	22	23	24	31	32	33	34	41	42	43	44
a	S' S T	x																			
	L TR																				
	BL R B																				
	F N		x	x	x																
b	S' S T																				
	L TR							x		x	x	x	x			x				x	
	BL R B					x	x	x	x					x	x	x	x				
	F N					x	x	x	x					x	x	x	x	x	x		x
c	S' S T									x	x		x								
	L TR																				
	BL R B					x	x	x		x	x	x		x	x	x		x	x	x	
	F N					x	x		x	x	x			x	x		x	x	x		x
d	S' S T	x								x	x	x									
	L TR							x		x	x	x	x			x				x	
	BL R B					x x	x x	x x	x	x	x	x		x x	x x	x x	x	x	x	x	
	F N		x	x	x	x	x	x	x	x	x	■	■	x	x	x	x	x	x	■	x

Table 12 Prototype Products in Q

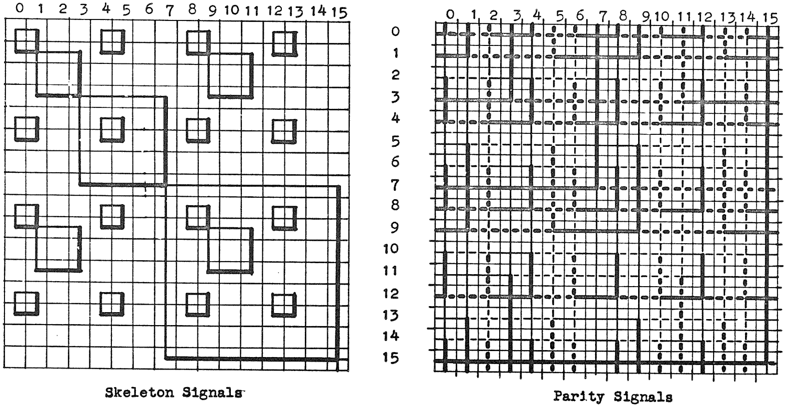
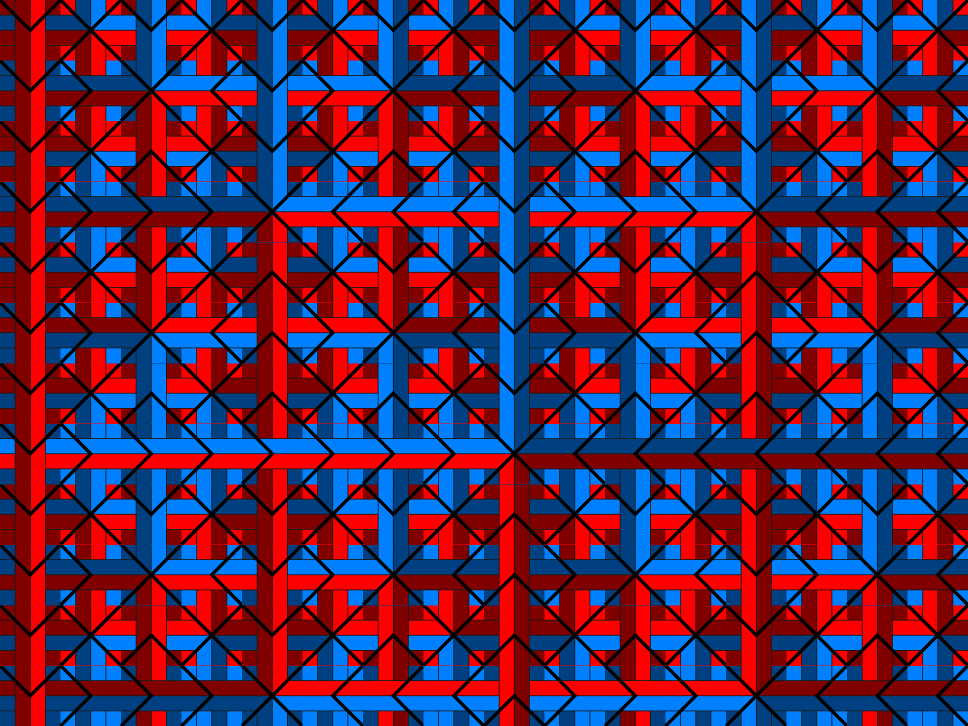


Figure 24 Part of the Solution of Q



Nilpotency and limit set

Definition The **limit set** of a CA F is the non-empty subshift

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

Remark Λ_F is the set of configurations appearing in **biinfinite space-time diagrams** $\Delta \in S^{\mathbb{Z}\times\mathbb{Z}}$ such that $\forall t \in \mathbb{Z}, \Delta(t+1) = F(\Delta(t))$.

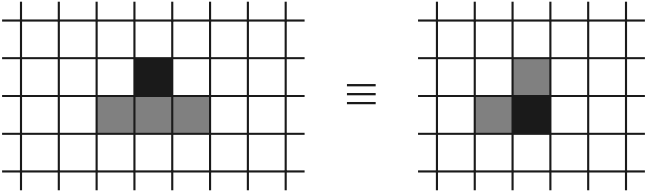
Lemma A CA is nilpotent iff its limit set is a **singleton**.

Reduction

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.



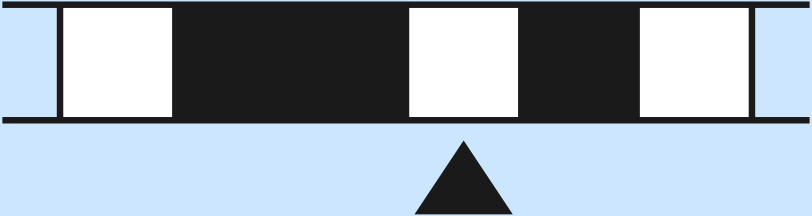
Theorem[Kari92] NW-DP $\leqslant_m$ Nil

Revisiting DP

Theorem[Kari92] **NW-DP** is **recursively undecidable**.

Remark Reprove of undecidability of **DP** with the additionnal determinism constraint!

Corollary **Nil** is **recursively undecidable**.



3. Periodicity and mortality

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)*

Definition A **TM** is a triple (S, Σ, T) with S the set of states, Σ the alphabet and T a set of instructions of two kinds:

(s, δ, t) : *"in state s move in direction 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 pair (s, c) where s is the state and the head points at position 0 of the tape c.

For **deterministic** TM, the **global map** $G : S \times \Sigma^{\mathbb{Z}} \to S \times \Sigma^{\mathbb{Z}}$ which applies instructions is a partial continuous map.

Undecidability of IP

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

Theorem[Hooper66] **IP** is **recursively undecidable**.

Remark To prove it one needs **aperiodic** TM.

Idea of the proof

Simulate 2-counters machines *à la* Minsky $(s, \underline{@}1^m x 2^n y)$

Replace **unbounded searches** by **recursive calls** to initial segments of the simulation.

Periodicity and reversibility

Definition A CA F is **reversible** if there exists a CA G such that $G = F^{-1}$.

Theorem A CA is **reversible** iff it is **bijective**.

Remark **Periodicity** implies **reversibility**.

Definition A TM (S, Σ, T) is **reversible** if (S, Σ, T^{-1}) is deterministic, where

$$\begin{aligned} (s, \delta, t)^{-1} &= (t, \delta, s) \\ (s, a, t, b)^{-1} &= (t, b, s, a) \end{aligned}$$

Reduction

Theorem[KO2008] R-IP $\leqslant_m$ TM-Per $\leqslant_m$ Per

Idea for TM-Per $\leqslant_m$ Per

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 $-$.
In case of local inconsistency, invert polarity.
The RCA is periodic iff $\mathcal{M}$ is periodic.

Revisiting IP

Theorem[KO2008] **R-IP** is **recursively undecidable**.

Remark Reprove of undecidability of **IP** with the additionnal reversibility constraint!

Corollary **TM-Per** and **Per** are **recursively undecidable**.

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)* $\Diamond$

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 $(s, \underline{@}1^m x 2^n y)$

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{@}}$111111111111x2222y *search x →*

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:

$@\underset{\overline{s}'_1}{\underline{1}}111111111111x2222y$ *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:

$\texttt{@1}\underset{\bar{s}'_2}{\underline{\texttt{1}}}\texttt{11111111111x2222y}$ *bounded search 2*

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:

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

Immortality: simulating RCM

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

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

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:

$@\underset{s_0}{\underline{@}}_s\mathbf{xy}1111111111x2222y$ *recursive call*

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:

$@@_s\mathbf{11111x22222}\underset{s_c}{\mathbf{y}}\mathtt{x2222y}$ *ultimately in case of collision...*

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:

$@111\underset{\overline{s}'_1}{\underline{1}}11111111x2222y$ *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 $(s, \underline{@}1^m x2^n y)$

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:

$@11111\underset{\overline{s}'_3}{\underline{1}}1111111x2222y$ *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:

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

Immortality: simulating RCM

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

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

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

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*

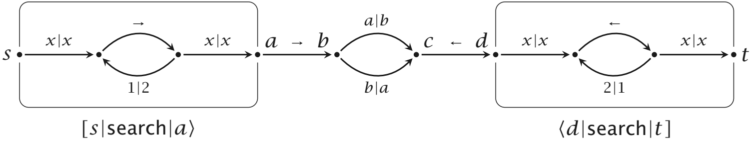
The RTM is immortal iff the 2-RCM is mortal on $(s_0, (0, 0))$.

Programming tips and tricks (1/2)

We designed a TM programming language called Gnirut:

`http://github.com/nopid/gnirut`

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

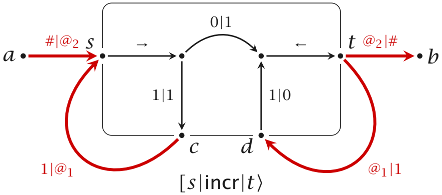



```
def [s|search|t⟩ :
   s. x ⊢ x, u
   u.  →, r
   r. 1 ⊢ 2, u | x ⊢ x, t

[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|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
```

Immortality: skeleton

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

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 ...

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

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 $s.\,\underline{@_\alpha}1^m\mathsf{x} \vdash \underline{@_\alpha}1^m\mathsf{x}, t$ 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

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

Program it!

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

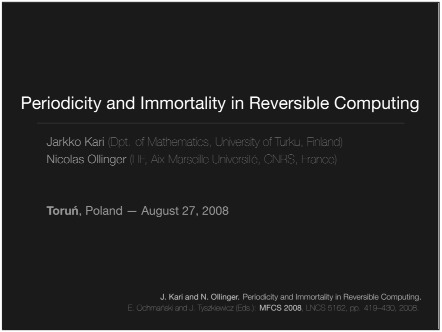
def $[s|\text{RCM}_2|co_1, co_2\rangle$:
$[s|\text{init}_2|s_0\rangle$
$[s_0|\text{test1}|s_{1z}, n\rangle$
$[s_1|\text{inc1}_2|s_2, co_1\rangle$
$[s_2|\text{inc2}_2|s_3, co_2\rangle$
$\left[s_3\,\middle|\,\text{test1}\,\middle|\,n', s_{1p}\right\rangle$
$\left\langle s_{1z}, s_{1p}\,\middle|\,\text{test1}\,\middle|\,s_1\right]$

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

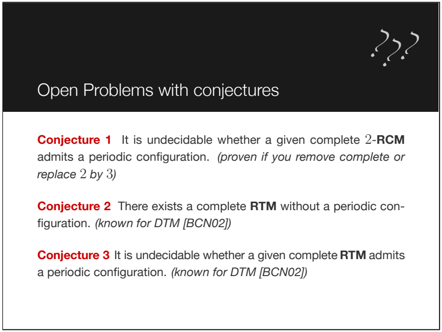
fun $[s|\text{check}_2|t\rangle$:
$[s|\text{RCM}_2|co_1, co_2, \ldots\rangle$
$\langle co_1, co_2, \ldots|\text{RCM}_2|t]$

Going further

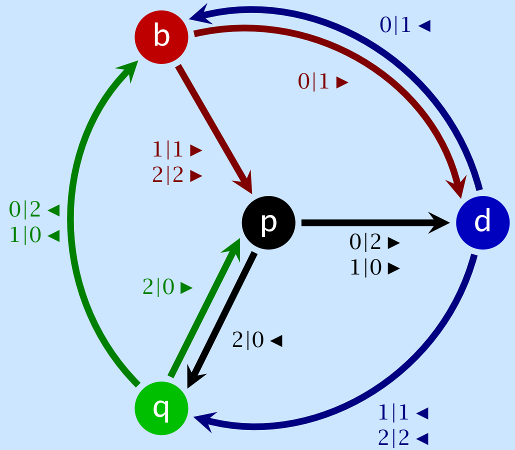
What is the equivalent of an aperiodic tileset for RTM?



...



Theorem To find if a given **complete reversible Turing machine** admits a **periodic orbit** is Σ_1-complete.



4. Dynamics of Turing machines

Turing machines (quintuples)

We go back to more classical TM.

Definition A **Turing machine** is a triple (Q, Σ, δ) where Q is the finite set of states, Σ is the finite set of tape symbols and $\delta : Q \times \Sigma \to Q \times \Sigma \times \{\blacktriangleleft, \blacktriangleright\}$ is the transition function.

Transition $\delta(s, a) = (t, b, d)$ means:
"in state s, when reading the symbol a on the tape,
replace it by b move the head in direction d and enter state t."

Remark We do not care about blank symbol or initial and final states, we see Turing machines as dynamical systems.

Reversible Turing machines

Intuitively, a TM is **reversible** if there exists another TM to compute backwards: "$T_2 = T_1^{-1}$". **Forget technical details**...

Definition A TM is **reversible** if δ can be decomposed as:

$$\begin{aligned} \delta(s,a) &= (t,b,\rho(t)) \qquad \text{where } (t,b) = \sigma(s,a) \\ \rho &: Q \to \{\blacktriangleleft, \blacktriangleright\} \\ \sigma &\in \mathfrak{S}_{Q\times\Sigma} \end{aligned}$$

Remark $\delta^{-1}(t,b) = (s,a,\blacklozenge(\rho(s)))$

Moving head vs moving tape dynamics

TMH

$$X_h \subset (Q \cup \Sigma)^{\mathbb{Z}}$$
$$T_h : X_h \to X_h$$

TMT

$$X_t = {}^{\omega}\Sigma \times Q \times \Sigma^{\omega}$$
$$T_t : X_t \to X_t$$

TMH	TMT
···000000**b**000000000···	···0000000**b**00000000···
···0000001**d**00000000···	···0000001**d**00000000···
···000000**b**110000000···	···0000000**b**11000000···
···0000001**p**10000000···	···0000001**p**10000000···
···00000010**d**0000000···	···0000010**d**00000000···
···0000001**b**01000000···	···0000001**b**01000000···
···00000011**d**1000000···	···0000011**d**10000000···
···0000001**q**11000000···	···0000001**q**11000000···
···000000**b**101000000···	···0000000**b**10100000···
···0000001**p**01000000···	···0000001**p**01000000···
⋮	⋮

Trace-shift dynamics

ST

$$S_T \subseteq (Q \times \Sigma)^\omega$$

$$\sigma : S_T \to S_T$$

TMT

$$X_t = {}^\omega\Sigma \times Q \times \Sigma^\omega$$

$$T_t : X_t \to X_t$$

The column shift of TMT

0 0 1 1 0 0 1 1 1 0 ...
b d b p d b d q b p

```
···0000000b00000000···
···0000001d00000000···
···0000000b11000000···
···0000001p10000000···
···0000010d00000000···
···0000001b01000000···
···0000011d10000000···
···0000001q11000000···
···0000000b10100000···
···0000001p01000000···
⋮
```

Searching for a reduction

We want to prove the following:

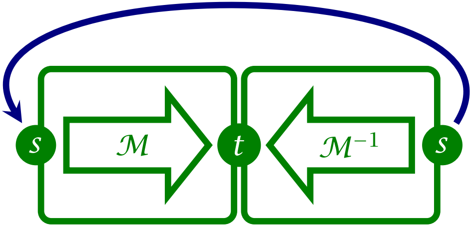
Theorem To find if a given **complete reversible Turing machine** admits a **periodic orbit** is Σ_1-complete.

In the partial case we use the following tool:

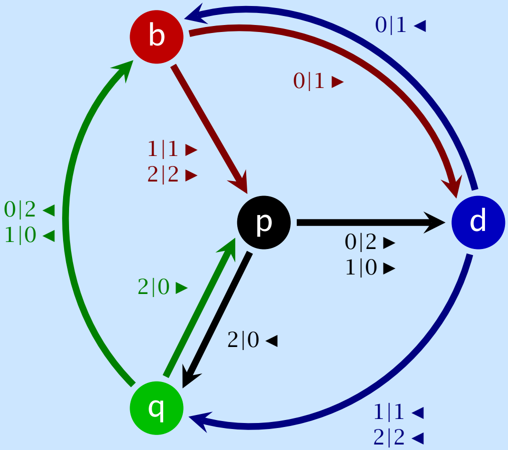
Prop[KO08] To find if a given **(aperiodic) RTM** can reach a given state t from a given state s is Σ_1-complete.

The partial case

Principle of the reduction Associate to an (aperiodic) RTM $\mathcal{M}$ with given s and t a new machine with a periodic orbit if and only if t is reachable from s.



We need to find a way to **complete** the constructed machine.
We will **embed** it into a **complete aperiodic** RTM.



5. a SMART machine

The SMART machine $\mathfrak{C}$

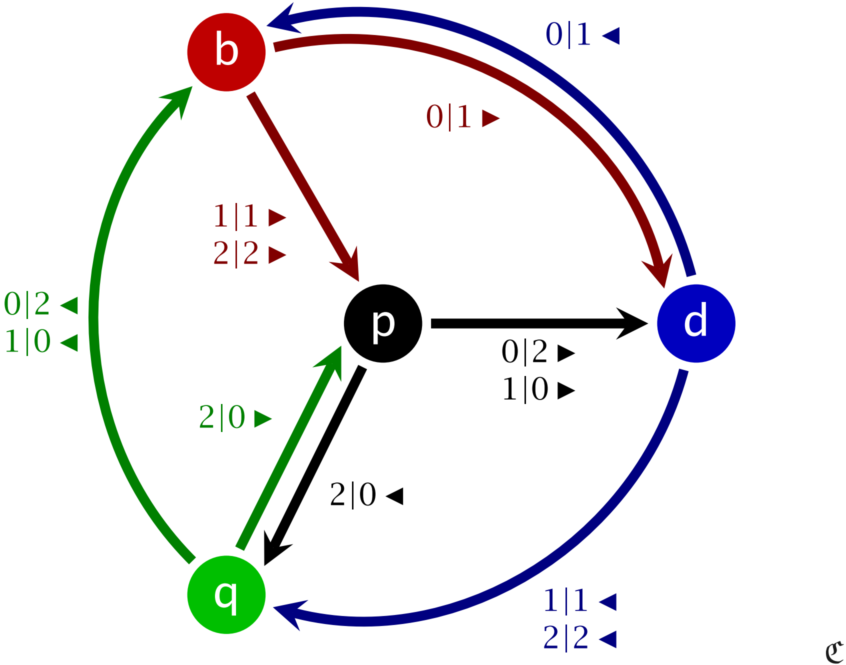
Conj[Kůrka97] Every **complete** TM has a **periodic** point.

Thm[BCN02] No, here is an **aperiodic** complete TM.

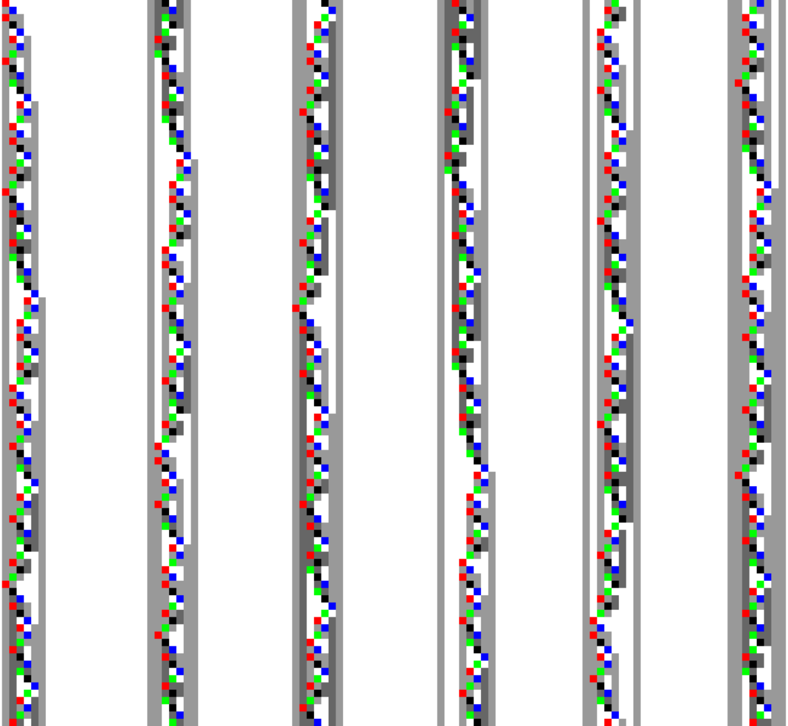
Rk It relies on the **bounded search** technique [Hooper66].

In 2008, I asked **J. Cassaigne** if he had a reversible version of the BCN construction...

...he answered with a small machine $\mathfrak{C}$ which is a reversible and (drastic) simplification of the BCN machine.



b
0|1 ◄
0|1 ►
1|1 ►
2|2 ►
0|2 ◄
1|0 ◄
p
d
0|2 ►
1|0 ►
2|0 ►
2|0 ◄
q
1|1 ◄
2|2 ◄



The SMART machine $\mathfrak{C}$

A 4-state 3-symbols TM with nice properties:

complete no halting configuration

reversible reversed by a TM...

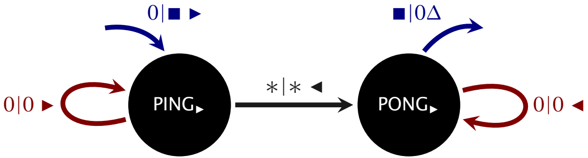
time-symetric ...essentially itself (up to details)

aperiodic no time periodic orbit

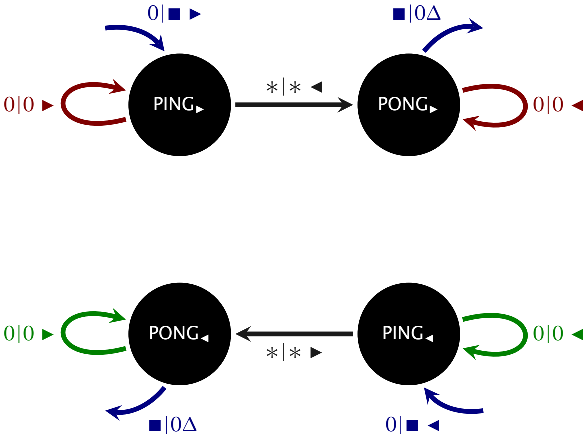
substitutive substitution-generated trace-shift language

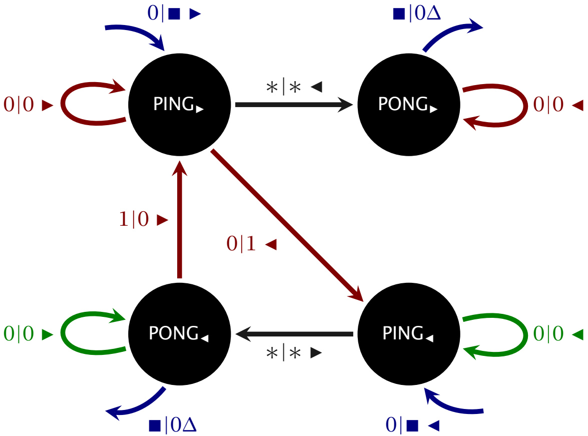
TMT-minimal every orbit is dense with moving tape

How does it work?

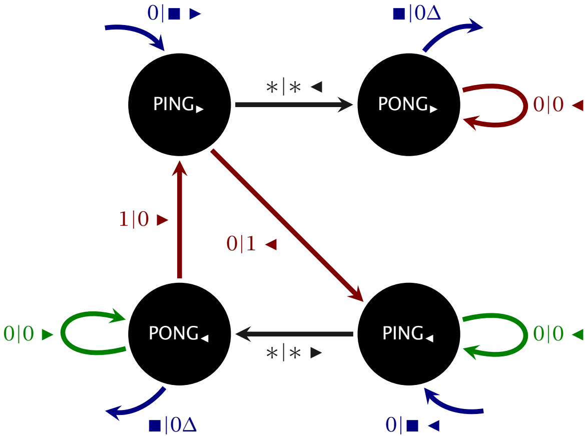


0|■ ▶
■|0Δ
0|0 ▶
PING▸
∗|∗ ◀
PONG▸
0|0 ◀

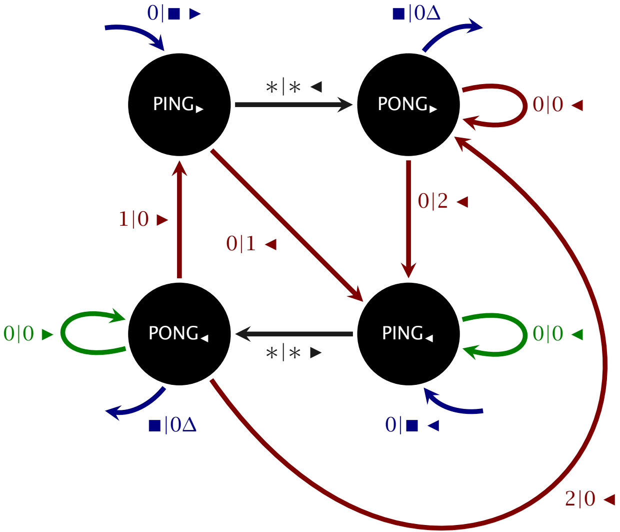

0|■ ▶
■|0Δ
0|0 ▶
PING▶
∗|∗ ◀
PONG▶
0|0 ◀
0|0 ▶
PONG◀
PING◀
0|0 ◀
∗|∗ ▶
■|0Δ
0|■ ◀



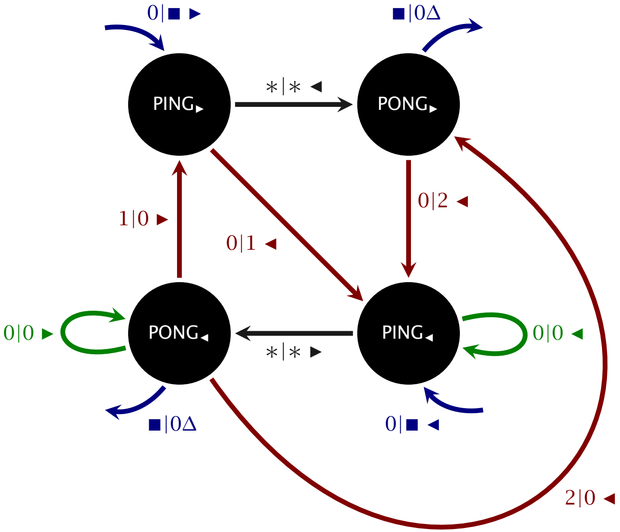
PING▸
PONG▸
PONG◂
PING◂
0|■ ▸
■|0Δ
0|0 ▸
| ◂
0|0 ◂
1|0 ▸
0|1 ◂
0|0 ▸
0|0 ◂
| ▸
■|0Δ
0|■ ◂



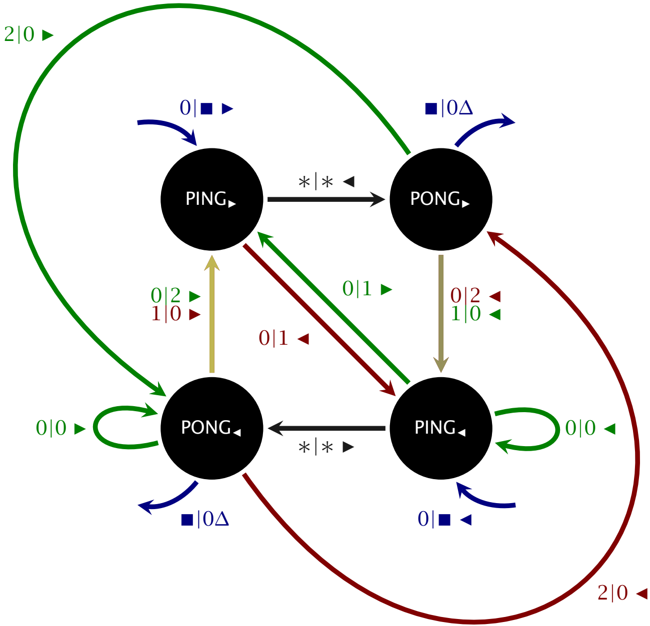
0|■ ▶
■|0Δ
PING▸
∗|∗ ◀
PONG▸
0|0 ◀
1|0 ▶
0|1 ◀
0|0 ▶
PONG◂
PING◂
0|0 ◀
∗|∗ ▶
■|0Δ
0|■ ◀



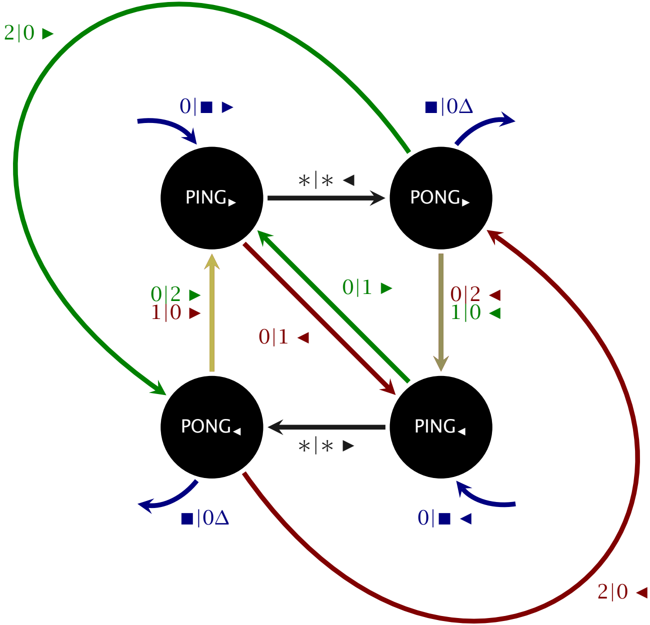
0|■ ▶
■|0Δ
| ◀
PING▶
PONG▶
0|0 ◀
0|2 ◀
1|0 ▶
0|1 ◀
0|0 ▶
PONG◀
PING◀
0|0 ◀
| ▶
■|0Δ
0|■ ◀
2|0 ◀



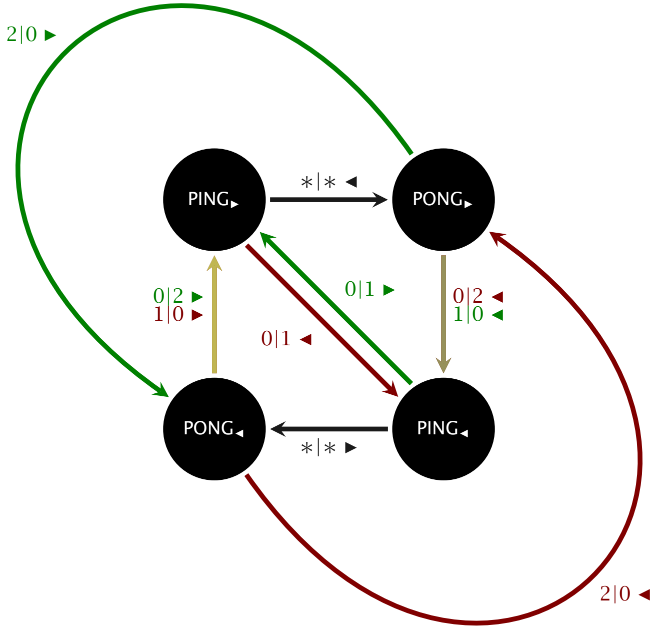
0|■ ▶
■|0Δ
PING▶
∗|∗ ◀
PONG▶
0|2 ◀
1|0 ▶
0|1 ◀
0|0 ▶
PONG◀
PING◀
0|0 ◀
∗|∗ ▶
■|0Δ
0|■ ◀
2|0 ◀

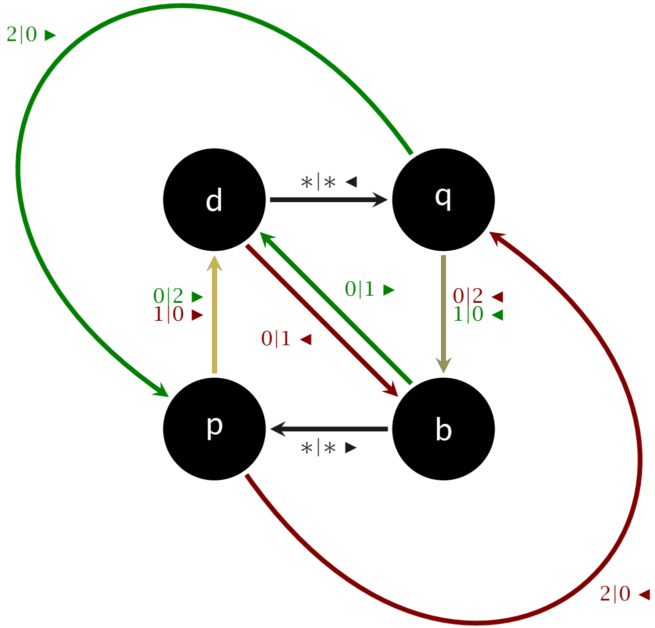


2|0 ▶
0|■ ▶
■|0Δ
| ◀
PING▸
PONG▸
0|1 ▶
0|2 ▶
1|0 ▶
0|2 ◀
1|0 ◀
0|1 ◀
0|0 ▶
PONG◂
PING◂
0|0 ◀
| ▶
■|0Δ
0|■ ◀
2|0 ◀



2|0 ▶
0|■ ▶
■|0Δ
PING▸
| ◀
PONG▸
0|2 ▶
1|0 ▶
0|1 ▶
0|1 ◀
0|2 ◀
1|0 ◀
PONG◂
PING◂
| ▶
■|0Δ
0|■ ◀
2|0 ◀


2|0 ▶
PING▶
∗|∗ ◀
PONG▶
0|2 ▶
1|0 ▶
0|1 ▶
0|2 ◀
1|0 ◀
0|1 ◀
PONG◀
PING◀
∗|∗ ▶
2|0 ◀



2|0 ▶
d
∗|∗ ◀
q
0|2 ▶
1|0 ▶
0|1 ▶
0|2 ◀
1|0 ◀
0|1 ◀
p
b
∗|∗ ▶
2|0 ◀

Recursive behavior

PING$_{\blacktriangleright}(n)$:
    for i=1 to n:
        d. 0|1, b ◀
        PING$_{\blacktriangleleft}(i-1)$
    d. $x|x$, q ◀
    for i=n downto 1:
        q. 0|2, b ◀
        PING$_{\blacktriangleleft}(i-1)$
    q. y|0, $\alpha(y)$ $\tau(y)$

PING$_{\blacktriangleleft}(n)$:
    for i=1 to n:
        b. 0|1, d ▶
        PING$_{\blacktriangleright}(i-1)$
    b. $x|x$, p ▶
    for i=n downto 1:
        p. 0|2, d ▶
        PING$_{\blacktriangleright}(i-1)$
    p. y|0, $\alpha'(y)$ $\tau'(y)$

$$\begin{cases} f(0) & = 2 \\ f(n+1) & = 3f(n) \end{cases}$$

Substitutive trace subshift

$$\varphi\begin{pmatrix}0\\ \mathbf{b}\end{pmatrix} = \begin{matrix}0&0&1&1\\ \mathbf{b}&\mathbf{d}&\mathbf{b}&\mathbf{p}\end{matrix} \qquad \varphi\begin{pmatrix}0\\ \mathbf{d}\end{pmatrix} = \begin{matrix}0&0&1&1\\ \mathbf{d}&\mathbf{b}&\mathbf{d}&\mathbf{q}\end{matrix}$$

$$\varphi\begin{pmatrix}x\\ \mathbf{b}\end{pmatrix} = \begin{matrix}x\\ \mathbf{b}\end{matrix} \qquad \varphi\begin{pmatrix}x\\ \mathbf{d}\end{pmatrix} = \begin{matrix}x\\ \mathbf{d}\end{matrix}$$

$$\varphi\begin{pmatrix}0\\ \mathbf{p}\end{pmatrix} = \begin{matrix}0&0&2&1\\ \mathbf{p}&\mathbf{d}&\mathbf{b}&\mathbf{p}\end{matrix} \qquad \varphi\begin{pmatrix}0\\ \mathbf{q}\end{pmatrix} = \begin{matrix}0&0&2&1\\ \mathbf{q}&\mathbf{b}&\mathbf{d}&\mathbf{q}\end{matrix}$$

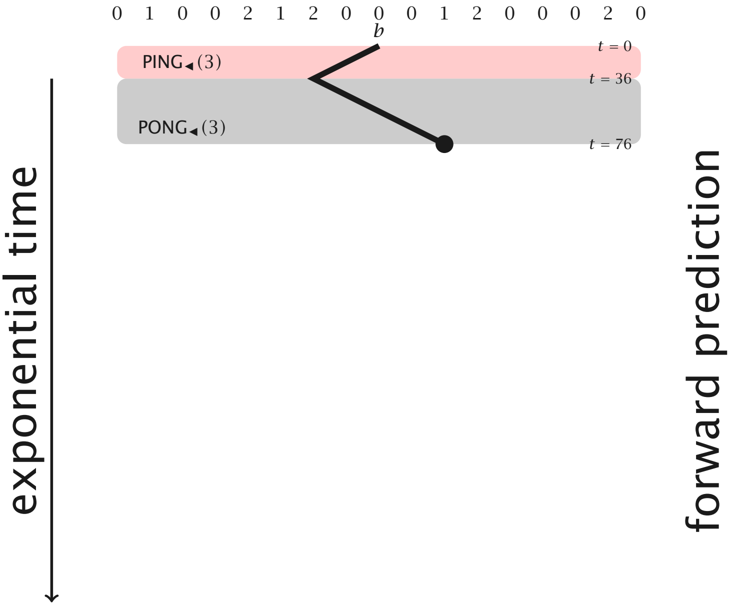
$$\varphi\begin{pmatrix}x\\ \mathbf{p}\end{pmatrix} = \begin{matrix}0&x&2&x\\ \mathbf{p}&\mathbf{d}&\mathbf{q}&\mathbf{p}\end{matrix} \qquad \varphi\begin{pmatrix}x\\ \mathbf{q}\end{pmatrix} = \begin{matrix}0&x&2&x\\ \mathbf{q}&\mathbf{b}&\mathbf{p}&\mathbf{q}\end{matrix}$$

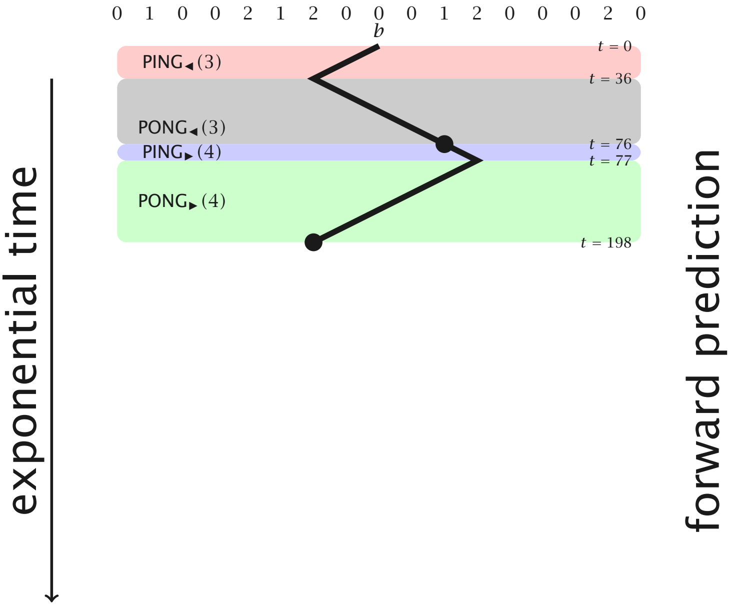
0 1 0 0 2 1 2 0 0 0 1 2 0 0 0 2 0

b

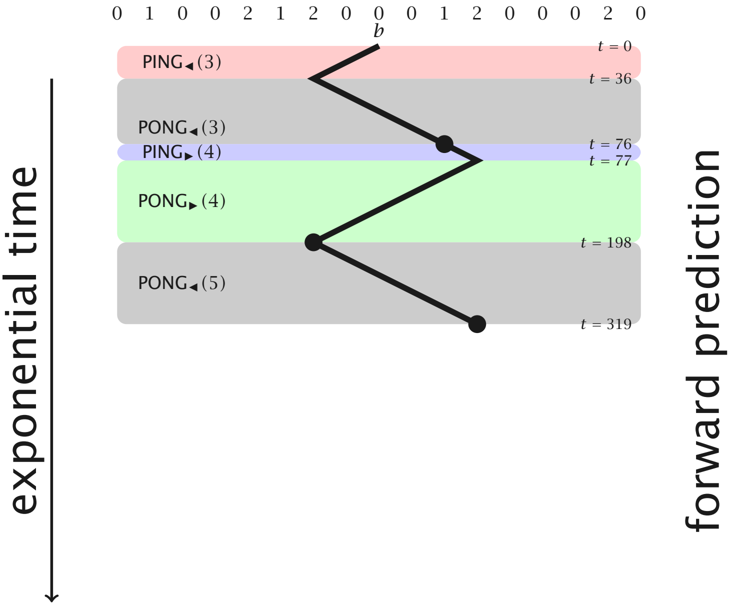
exponential time

forward prediction

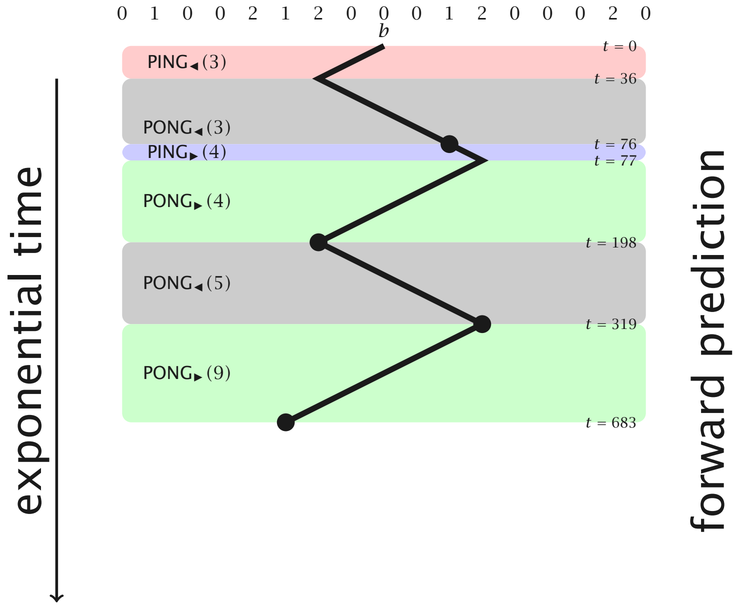

0 1 0 0 2 1 2 0 0 0 1 2 0 0 0 2 0
b
t = 0
PING◂(3)
t = 36
PONG◂(3)
t = 76
exponential time
forward prediction



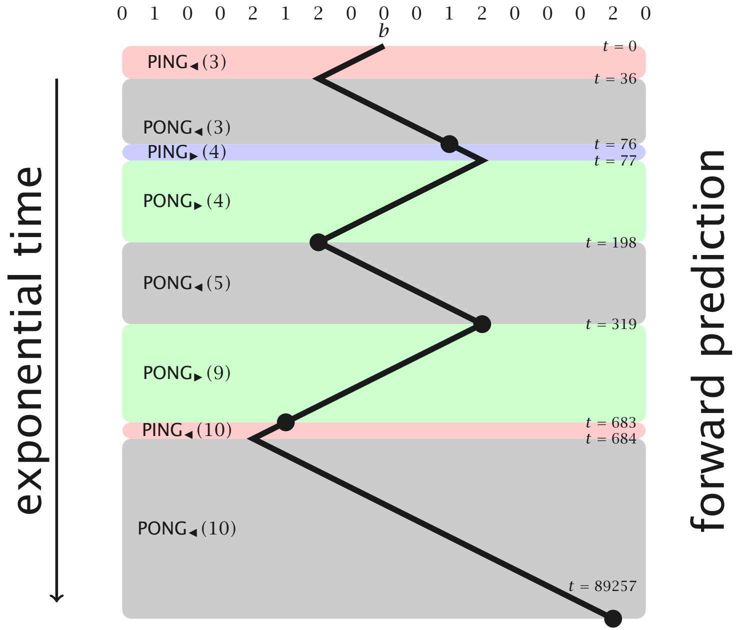
0 1 0 0 2 1 2 0 0 0 1 2 0 0 0 2 0
b
t = 0
PING◀(3)
t = 36
PONG◀(3)
t = 76
PING▶(4)
t = 77
PONG▶(4)
t = 198
exponential time
forward prediction



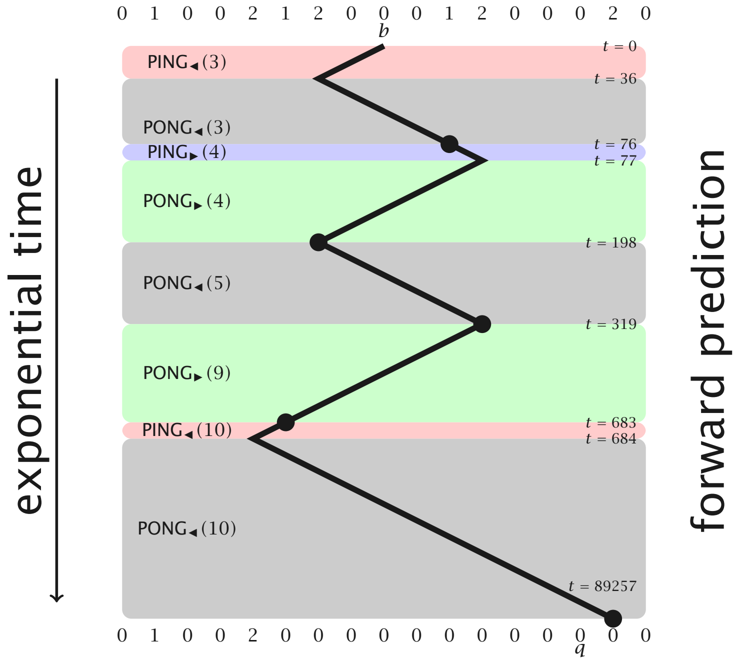
0 1 0 0 2 1 2 0 0 0 1 2 0 0 0 2 0
b
t = 0
PING◀(3)
t = 36
PONG◀(3)
t = 76
PING▶(4)
t = 77
PONG▶(4)
t = 198
PONG◀(5)
t = 319
exponential time
forward prediction



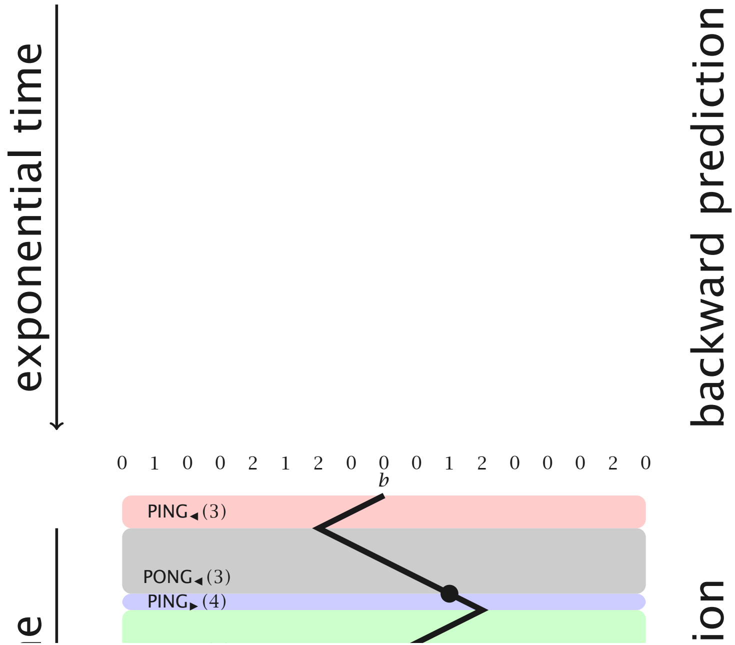
0 1 0 0 2 1 2 0 0 0 1 2 0 0 0 2 0
b
t = 0
PING◀(3)
t = 36
PONG◀(3)
t = 76
PING▶(4)
t = 77
PONG▶(4)
t = 198
PONG◀(5)
t = 319
PONG▶(9)
t = 683
exponential time
forward prediction

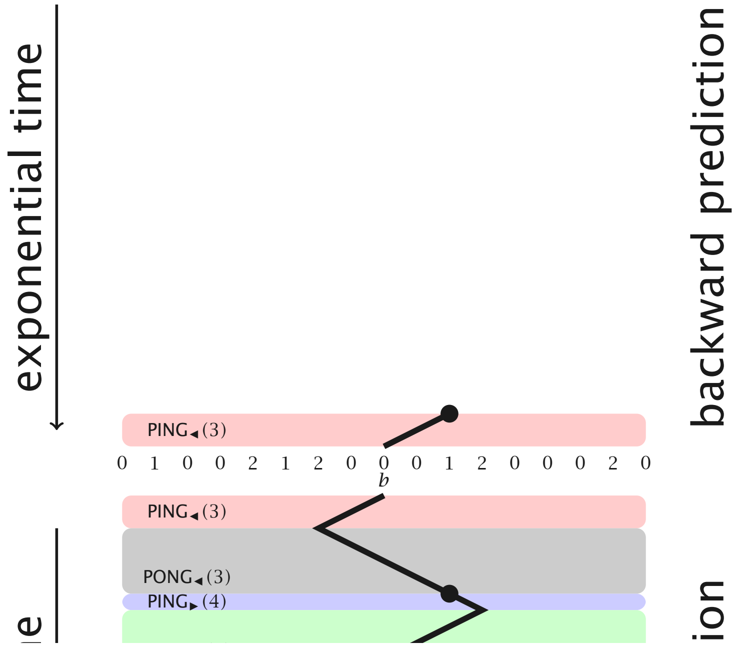


0 1 0 0 2 1 2 0 0 0 1 2 0 0 0 2 0
b
t = 0
PING◀(3)
t = 36
PONG◀(3)
t = 76
PING▶(4)
t = 77
PONG▶(4)
t = 198
PONG◀(5)
t = 319
PONG▶(9)
t = 683
PING◀(10)
t = 684
PONG◀(10)
t = 89257
exponential time
forward prediction

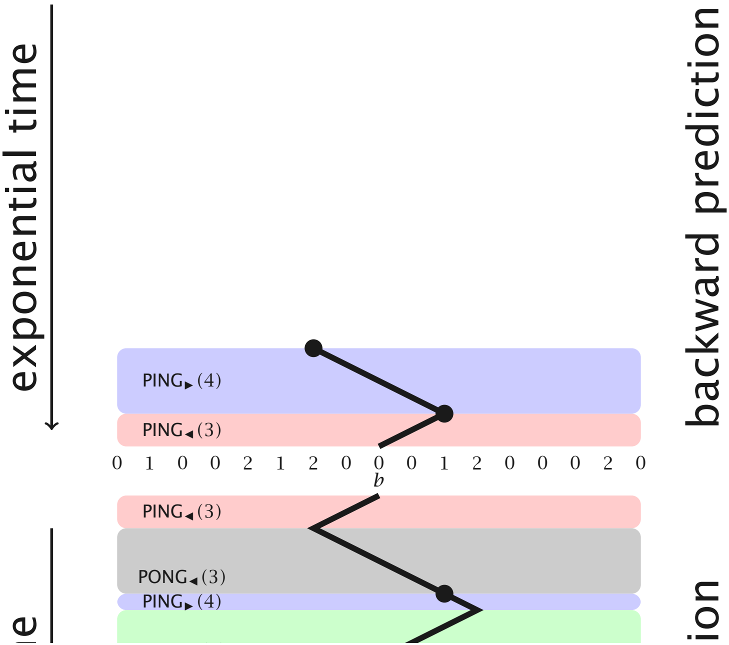


0 1 0 0 2 1 2 0 0 0 1 2 0 0 0 2 0
b
t = 0
PING◂(3)
t = 36
PONG◂(3)
t = 76
PING▸(4)
t = 77
PONG▸(4)
t = 198
PONG◂(5)
t = 319
PONG▸(9)
t = 683
PING◂(10)
t = 684
PONG◂(10)
t = 89257
0 1 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0
q
exponential time
forward prediction

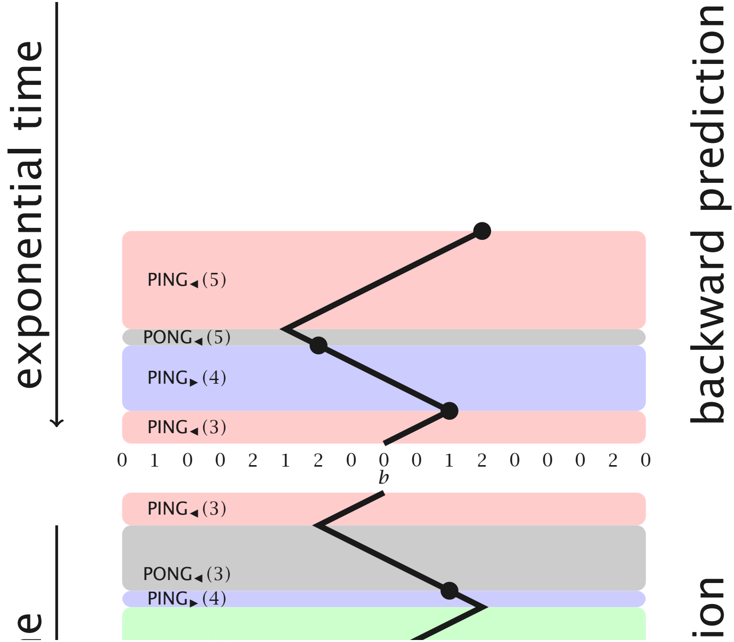

exponential time
backward prediction
0 1 0 0 2 1 2 0 0 0 1 2 0 0 0 2 0
b
PING◂ (3)
PONG◂ (3)
PING▸ (4)

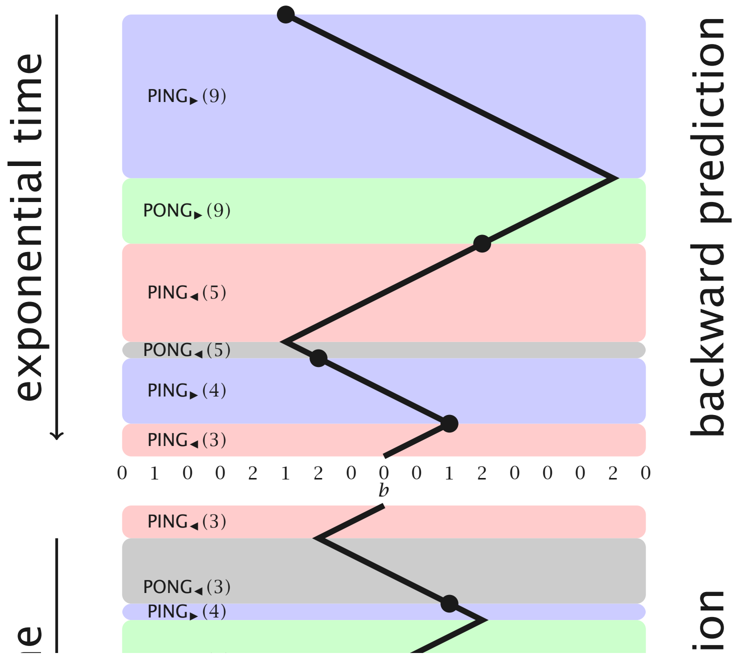


exponential time
backward prediction
PING◂(3)
0 1 0 0 2 1 2 0 0 0 1 2 0 0 0 2 0
b
PING◂(3)
PONG◂(3)
PING▸(4)

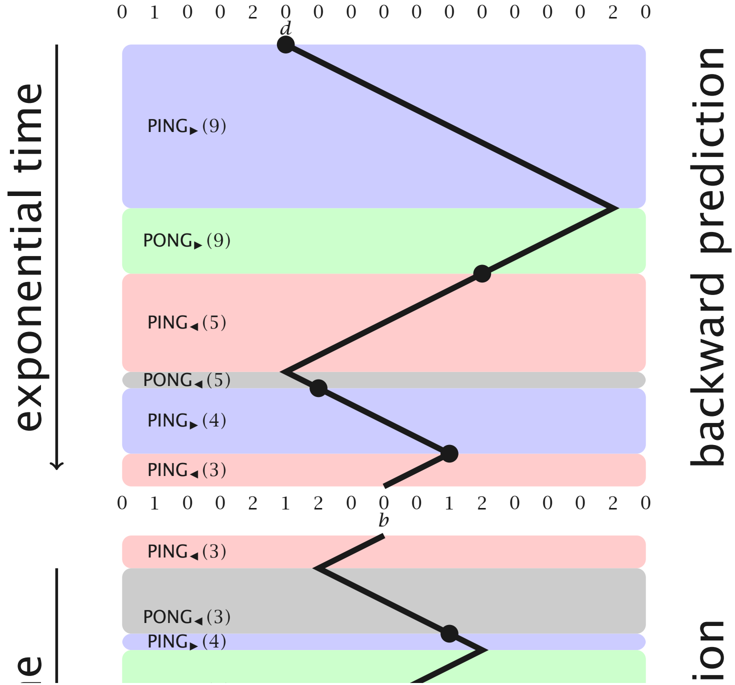


exponential time
backward prediction
PING▸(4)
PING◂(3)
0 1 0 0 2 1 2 0 0 0 1 2 0 0 0 2 0
b
PING◂(3)
PONG◂(3)
PING▸(4)


exponential time
backward prediction
PING◂(5)
PONG◂(5)
PING▸(4)
PING◂(3)
0 1 0 0 2 1 2 0 0 0 1 2 0 0 0 2 0
b
PING◂(3)
PONG◂(3)
PING▸(4)
ion



exponential time
backward prediction
PING▸(9)
PONG▸(9)
PING◂(5)
PONG◂(5)
PING▸(4)
PING◂(3)
0 1 0 0 2 1 2 0 0 0 1 2 0 0 0 2 0
b
PING◂(3)
PONG◂(3)
PING▸(4)



0 1 0 0 2 0 0 0 0 0 0 0 0 0 0 2 0
d
exponential time
backward prediction
PING▸ (9)
PONG▸ (9)
PING◂ (5)
PONG◂ (5)
PING▸ (4)
PING◂ (3)
0 1 0 0 2 1 2 0 0 0 1 2 0 0 0 2 0
b
PING◂ (3)
PONG◂ (3)
PING▸ (4)

SMART is transitive in TMH, TMT and ST

Proposition $\left({}^{\omega}2 \cdot \underset{p}{2}\ 2^{\omega} \right)$ is a **transitive point**.

Proof

(Forward) For all $k \geqslant 0$:

$$\left({}^{\omega}2 \cdot \underset{p}{2}\ 2^{\omega} \right) \vdash^{*} \left({}^{\omega}2\ \underset{q}{2}\ 0^{k} . 0\ 0^{k}\ 2^{\omega} \right) \quad .$$

(Backward) For every partial configuration $\left(\underset{\leftarrow}{u}\ \underset{\alpha}{\cdot}\ \underset{\rightarrow}{v} \right)$, there exist $w, w' \in \{0,1,2\}^{*}$ and $k > 0$ big enough such that

$$\left({}^{\omega}2\ \underset{q}{2}\ 0^{k} . 0\ 0^{k}\ 2^{\omega} \right) \vdash^{*} \left({}^{\omega}2\ w\ \underset{\leftarrow}{u}\ \underset{\alpha}{\cdot}\ \underset{\rightarrow}{v}\ w'\ 2^{\omega} \right) \quad .$$

■

Typical use

Combining the SMART machine with a generic **Embedding technique** provides new undecidability results.

Theorem **Transitivity** is Π^0_1-hard in TMH, TMT and ST.

Theorem **Minimality** is Σ^0_1-hard in TMT and ST.

Table of contents