# On the Complexity of Limit Sets of Cellular Automata Associated with Probability Measures

MFCS 2006

L. Boyer[1] V. Poupet[1] G. Theyssier[2]

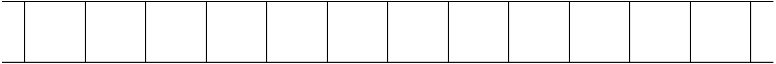
1. LIP (CNRS, ENS Lyon, UCB Lyon, INRIA,), France

2. LAMA, (CNRS, Université de Savoie), France

August 28, 2006

# Cellular Automata

- a lattice of cells (in this talk: $\mathbb{Z}$)

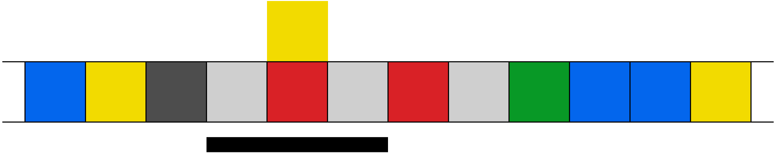


# Cellular Automata

- a lattice of cells (in this talk: $\mathbb{Z}$)
- each cell has a state among a finite set

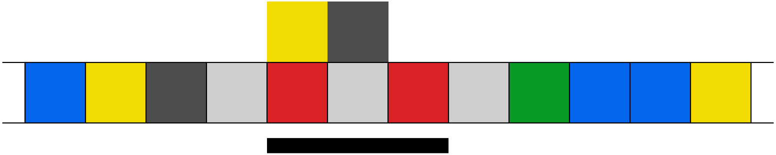
# Cellular Automata

- a lattice of cells (in this talk: $\mathbb{Z}$)
- each cell has a state among a finite set
- cells update their state by looking at a finite neighbourhood



# Cellular Automata

- a lattice of cells (in this talk: $\mathbb{Z}$)
- each cell has a state among a finite set
- cells update their state by looking at a finite neighbourhood
- same neighbourhood and same updating rule for all cells

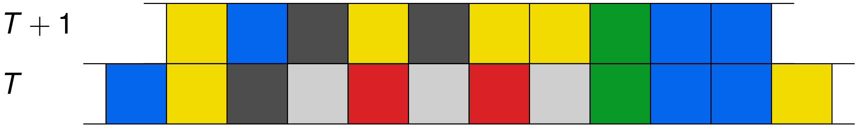


# Cellular Automata

- a lattice of cells (in this talk: $\mathbb{Z}$)
- each cell has a state among a finite set
- cells update their state by looking at a finite neighbourhood
- same neighbourhood and same updating rule for all cells

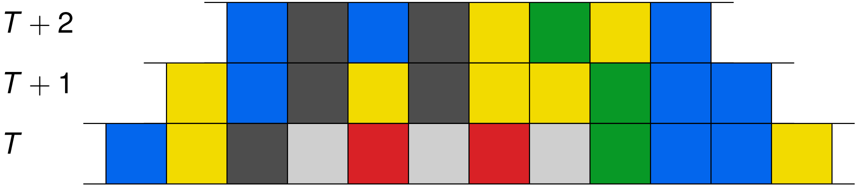
# Cellular Automata

- a lattice of cells (in this talk: $\mathbb{Z}$)
- each cell has a state among a finite set
- cells update their state by looking at a finite neighbourhood
- same neighbourhood and same updating rule for all cells
- global behaviour: synchronous update at each discrete time



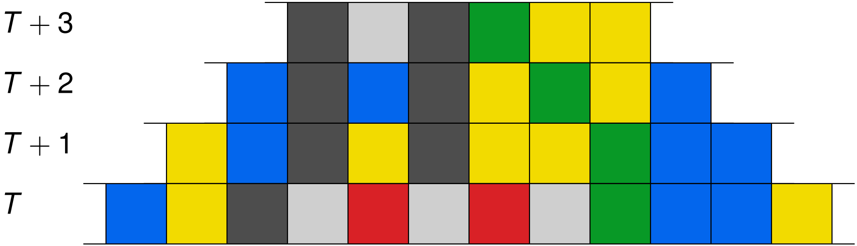
# Cellular Automata

- a lattice of cells (in this talk: $\mathbb{Z}$)
- each cell has a state among a finite set
- cells update their state by looking at a finite neighbourhood
- same neighbourhood and same updating rule for all cells
- global behaviour: synchronous update at each discrete time



# Cellular Automata

- a lattice of cells (in this talk: $\mathbb{Z}$)
- each cell has a state among a finite set
- cells update their state by looking at a finite neighbourhood
- same neighbourhood and same updating rule for all cells
- global behaviour: synchronous update at each discrete time



# Cellular Automata

- given: $\mathcal{A}$, local description of a CA
  - lattice $\mathbb{Z}$
  - states set $\Sigma$
  - size of neighbourhood $k$
  - local updating rule $\delta_{\mathcal{A}} : \Sigma^k \to \Sigma$

# Cellular Automata

- given: $\mathcal{A}$, local description of a CA
  - lattice $\mathbb{Z}$
  - states set $\Sigma$
  - size of neighbourhood $k$
  - local updating rule $\delta_{\mathcal{A}} : \Sigma^k \to \Sigma$
- induced: $G_{\mathcal{A}}$, global mapping from configurations to configurations

**definition:** *a configuration c is a mapping from cells to states* ($c : \mathbb{Z} \to \Sigma$)

# Cellular Automata

- given: $\mathcal{A}$, local description of a CA
  - lattice $\mathbb{Z}$
  - states set $\Sigma$
  - size of neighbourhood $k$
  - local updating rule $\delta_{\mathcal{A}} : \Sigma^k \to \Sigma$
- induced: $G_{\mathcal{A}}$, global mapping from configurations to configurations

**definition:** *a configuration $c$ is a mapping from cells to states ($c : \mathbb{Z} \to \Sigma$)*

## Main question

Knowing $\mathcal{A}$, what is the long term behaviour of $G_{\mathcal{A}}$?

# Classical tool: limit sets

- "configurations that may appear arbitrarily late in the evolution"

# Classical tool: limit sets

- "configurations that may appear arbitrarily late in the evolution"
- $\Omega_{\mathcal{A}}$ is defined by:

$$c \in \Omega_{\mathcal{A}} \overset{\text{def}}{\iff} \forall t, \exists c_0 : {G_{\mathcal{A}}}^t(c_0) = c$$

# Classical tool: limit sets

- "configurations that may appear arbitrarily late in the evolution"
- $\Omega_{\mathcal{A}}$ is defined by:

$$c \in \Omega_{\mathcal{A}} \overset{\text{def}}{\iff} \forall t, \exists c_0 : G_{\mathcal{A}}{}^{t}(c_0) = c$$

- $\mathcal{A}$ is nilpotent $\overset{\text{def}}{\iff}$ $\Omega_{\mathcal{A}}$ is a singleton.

**Theorems (J. Kari, 90s)**

1. *Nilpotency is undecidable.*
2. *Any property of limit sets is either trivial or undecidable.*

# Classical tool: limit sets

- "configurations that may appear arbitrarily late in the evolution"
- $\Omega_{\mathcal{A}}$ is defined by:

$$c \in \Omega_{\mathcal{A}} \overset{\text{def}}{\iff} \forall t, \exists c_0 : {G_{\mathcal{A}}}^t(c_0) = c$$

- $\mathcal{A}$ is nilpotent $\overset{\text{def}}{\iff}$ $\Omega_{\mathcal{A}}$ is a singleton.

**Theorems (J. Kari, 90s)**

1. *Nilpotency is undecidable.*
2. *Any property of limit sets is either trivial or undecidable.*

- set-theoretic point of view
- complexity of $\Omega_{\mathcal{A}}$ can come from a negligible set of configurations

# Recent tool: $\mu$-limit sets (P. Kůrka, A. Maass, 2000)

- "typical configurations that appear arbitrarily late in the evolution"

# Recent tool: $\mu$-limit sets (P. Kůrka, A. Maass, 2000)

- "typical configurations that appear arbitrarily late in the evolution"
- fix a measure $\mu$ over $\Sigma^{\mathbb{Z}}$ (in this talk: a Bernouilli measure)

# Recent tool: $\mu$-limit sets (P. Kůrka, A. Maass, 2000)

- “typical configurations that appear arbitrarily late in the evolution”
- fix a measure $\mu$ over $\Sigma^{\mathbb{Z}}$ (in this talk: a Bernouilli measure)

## Definitions

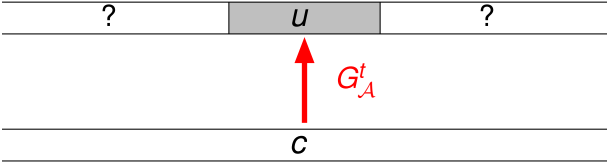
- $[u] \stackrel{def}{=} \{$*configurations with u as factor around the center cell*$\}$

| | | |
|---|---|---|
| ? | $u$ | ? |

- "typical configurations that appear arbitrarily late in the evolution"
- fix a measure $\mu$ over $\Sigma^{\mathbb{Z}}$ (in this talk: a Bernouilli measure)

## Definitions

- $[u] \stackrel{def}{=} \{$*configurations with u as factor around the center cell*$\}$
- $\mu_t(u) \stackrel{def}{=} \mu(\{c : G^t_{\mathcal{A}}(c) \in [u]\})$

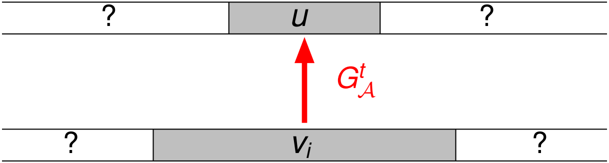


# Recent tool: $\mu$-limit sets (P. Kůrka, A. Maass, 2000)

- "typical configurations that appear arbitrarily late in the evolution"
- fix a measure $\mu$ over $\Sigma^{\mathbb{Z}}$ (in this talk: a Bernouilli measure)

## Definitions

- $[u] \stackrel{def}{=} \{\textit{configurations with } u \textit{ as factor around the center cell}\}$
- $\mu_t(u) \stackrel{def}{=} \sum_i \mu([v_i])$ *where* $\bigcup_i [v_i] = \{c : G^t_{\mathcal{A}}(c) \in [u]\}$

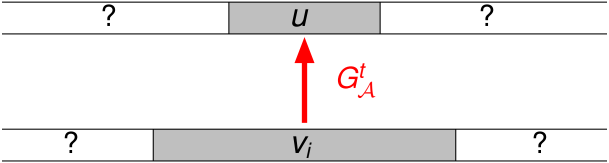


# Recent tool: $\mu$-limit sets (P. Kůrka, A. Maass, 2000)

- "typical configurations that appear arbitrarily late in the evolution"
- fix a measure $\mu$ over $\Sigma^{\mathbb{Z}}$ (in this talk: a Bernouilli measure)

## Definitions

- $[u] \stackrel{def}{=} \{\textit{configurations with } u \textit{ as factor around the center cell}\}$
- $\mu_t(u) \stackrel{def}{=} \sum_i \mu([v_i])$ *where* $\bigcup_i [v_i] = \{c : G^t_{\mathcal{A}}(c) \in [u]\}$
- $u \in L_\mu(\mathcal{A}) \stackrel{def}{\Longleftrightarrow} \mu_t(u) \not\to 0$ *as* $t \to \infty$

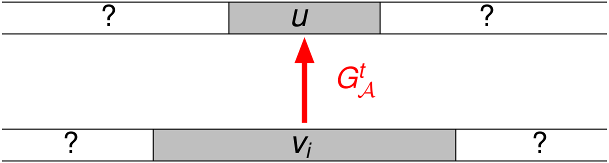


# Recent tool: $\mu$-limit sets (P. Kůrka, A. Maass, 2000)

- "typical configurations that appear arbitrarily late in the evolution"
- fix a measure $\mu$ over $\Sigma^{\mathbb{Z}}$ (in this talk: a Bernouilli measure)

## Definitions

- $[u] \overset{def}{=} \{$*configurations with u as factor around the center cell*$\}$
- $\mu_t(u) \overset{def}{=} \sum_i \mu([v_i])$ *where* $\bigcup_i [v_i] = \{c : G^t_{\mathcal{A}}(c) \in [u]\}$
- $u \in L_\mu(\mathcal{A}) \overset{def}{\iff} \mu_t(u) \not\to 0$ *as* $t \to \infty$
- $c \in \Upsilon_\mu(\mathcal{A}) \overset{def}{\iff}$ *all finite factors of c are in* $L_\mu(\mathcal{A})$



# Example 1: the 'max' automaton

- $\Sigma = \{\blacksquare, \blacksquare, \blacksquare, \blacksquare\}$
- $\blacksquare < \blacksquare < \blacksquare < \blacksquare$
- $\delta_{\mathcal{A}}(x, y, z) = \max(x, y, z)$

# Example 1: the 'max' automaton

- $\Sigma = \{\blacksquare, \blacksquare, \blacksquare, \blacksquare\}$
- $\blacksquare < \blacksquare < \blacksquare < \blacksquare$
- $\delta_{\mathcal{A}}(x, y, z) = \max(x, y, z)$

## ▶ Properties of $\mathcal{A}$:

- $\Omega_{\mathcal{A}} = \{$decreasing then increasing configurations$\}$ $(\searrow\nearrow)$

# Example 1: the 'max' automaton

- $\Sigma = \{\color{blue}\blacksquare\color{black}, \color{gold}\blacksquare\color{black}, \color{red}\blacksquare\color{black}, \color{green}\blacksquare\color{black}\}$
- $\color{blue}\blacksquare \color{black} < \color{gold}\blacksquare \color{black} < \color{red}\blacksquare \color{black} < \color{green}\blacksquare$
- $\delta_{\mathcal{A}}(x, y, z) = \max(x, y, z)$

▶ **Properties of $\mathcal{A}$:**

- $\Omega_{\mathcal{A}} = \{$decreasing then increasing configurations$\}$ ($\searrow\nearrow$)
- $\Upsilon_{\mu}(\mathcal{A}) = \{{}^{\omega}\color{green}\blacksquare\color{black}^{\omega}\}$ for any (complete) Bernouilli measure $\mu$

# Example 1: the 'max' automaton

- $\Sigma = \{{\color{blue}\blacksquare}, {\color{yellow}\blacksquare}, {\color{red}\blacksquare}, {\color{green}\blacksquare}\}$
- ${\color{blue}\blacksquare} < {\color{yellow}\blacksquare} < {\color{red}\blacksquare} < {\color{green}\blacksquare}$
- $\delta_{\mathcal{A}}(x, y, z) = \max(x, y, z)$

▸ **Properties of $\mathcal{A}$:**

- $\Omega_{\mathcal{A}} = \{$decreasing then increasing configurations$\}$ ($\searrow\nearrow$)
- $\Upsilon_\mu(\mathcal{A}) = \{{}^\omega{\color{green}\blacksquare}^\omega\}$ for any (complete) Bernouilli measure $\mu$

▸ **Definition:** a CA is $\mu$-quasi-nilpotent if its $\mu$-limit set is a singleton

- $\Sigma = \{\blacksquare, \blacksquare, \blacksquare, \blacksquare\}$
- $\blacksquare < \blacksquare < \blacksquare < \blacksquare$
- $\delta_{\mathcal{A}}(x, y, z) = \max(x, y, z)$

## ▶ Properties of $\mathcal{A}$:

- $\Omega_{\mathcal{A}} = \{$decreasing then increasing configurations$\}$ ($\searrow\nearrow$)
- $\Upsilon_{\mu}(\mathcal{A}) = \{{}^{\omega}\blacksquare^{\omega}\}$ for any (complete) Bernouilli measure $\mu$

▶ **Definition:** a CA is $\mu$-quasi-nilpotent if its $\mu$-limit set is a singleton

## ▶ Limit sets *vs.* $\mu$-limit sets:

- there is a $\mu$-quasi-nilpotent CA with a non recursive limit set
- it is undecidable to know whether a $\mu$-quasi-nilpotent CA is nilpotent

# A partial characterisation

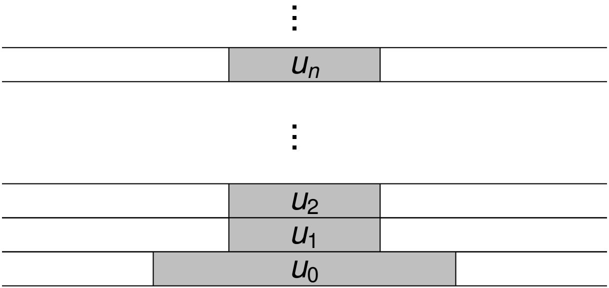
## Definitions (walls and bricks)

- *A* *wall* *is a sequence* $W = (u_i)$ *of words such that :*
  1. $|u_0| \geq |u_1| = |u_2| = \cdots$

# A partial characterisation

## Definitions (walls and bricks)

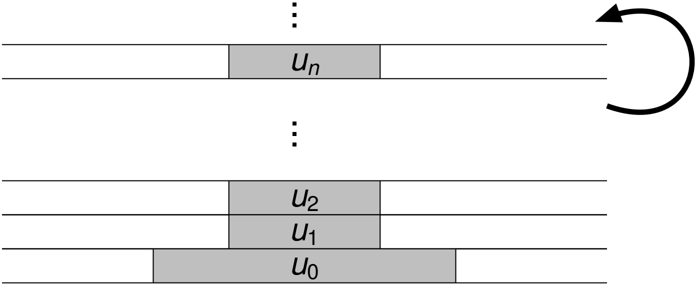
- *A wall is a sequence $W = (u_i)$ of words such that :*
  1. $|u_0| \geq |u_1| = |u_2| = \cdots$
  2. $\forall c \in [u_0], \forall t : G^t_{\mathcal{A}}(c) \in [u_t]$



# A partial characterisation

## Definitions (walls and bricks)

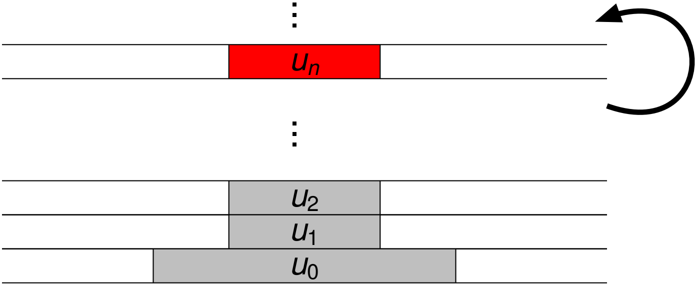
- *A wall is a sequence $W = (u_i)$ of words such that :*
  1. $|u_0| \geq |u_1| = |u_2| = \cdots$
  2. $\forall c \in [u_0], \forall t : G^t_{\mathcal{A}}(c) \in [u_t]$
- *This definition implies that $W$ is ultimately periodic.*



# A partial characterisation

## Definitions (walls and bricks)

- *A wall is a sequence* $W = (u_i)$ *of words such that :*
  1. $|u_0| \geq |u_1| = |u_2| = \cdots$
  2. $\forall c \in [u_0], \forall t : G^t_{\mathcal{A}}(c) \in [u_t]$
- *This definition implies that* $W$ *is ultimately periodic.*
- *A brick of* $W$ *is any word* $u_n$ *in the period.*



# A partial characterisation

## Definitions (walls and bricks)

- *A wall is a sequence $W = (u_i)$ of words such that :*
  1. $|u_0| \geq |u_1| = |u_2| = \cdots$
  2. $\forall c \in [u_0], \forall t : G^t_{\mathcal{A}}(c) \in [u_t]$
- *This definition implies that $W$ is ultimately periodic.*
- *A brick of $W$ is any word $u_n$ in the period.*

## Theorem (Bricks theorem)

***If** $\mathcal{A}$ is a CA with neighbourhood size $k$ and having a brick of size $\geq k$ **then** $L_\mu(\mathcal{A}) = \{$bricks of $\mathcal{A}\}$.*

# A partial characterisation

## Definitions (walls and bricks)

- *A wall is a sequence $W = (u_i)$ of words such that :*
  1. $|u_0| \geq |u_1| = |u_2| = \cdots$
  2. $\forall c \in [u_0], \forall t : G^t_{\mathcal{A}}(c) \in [u_t]$
- *This definition implies that $W$ is ultimately periodic.*
- *A brick of $W$ is any word $u_n$ in the period.*

## Theorem (Bricks theorem)

***If*** *$\mathcal{A}$ is a CA with neighbourhood size $k$ and having a brick of size $\geq k$* ***then*** *$L_\mu(\mathcal{A}) = \{\text{bricks of } \mathcal{A}\}$.*

▶ **Corollary:**

- $\mu$-limit sets of such CA do not depend on $\mu$ (within Bernouilli $\mu$).

# A partial characterisation

## Definitions (walls and bricks)

- *A wall is a sequence $W = (u_i)$ of words such that :*
  1. $|u_0| \geq |u_1| = |u_2| = \cdots$
  2. $\forall c \in [u_0], \forall t : G^t_{\mathcal{A}}(c) \in [u_t]$
- *This definition implies that $W$ is ultimately periodic.*
- *A brick of $W$ is any word $u_n$ in the period.*

## Theorem (Bricks theorem)

***If*** *$\mathcal{A}$ is a CA with neighbourhood size $k$ and having a brick of size $\geq k$* ***then*** *$L_\mu(\mathcal{A}) = \{$bricks of $\mathcal{A}\}$.*

- **Corollary:**
  - $\mu$-limit sets of such CA do not depend on $\mu$ (within Bernouilli $\mu$).
- **Known fact:** no brick of size $\geq k \iff$ sensitive to initial conditions

# A partial characterisation

## Definitions (walls and bricks)

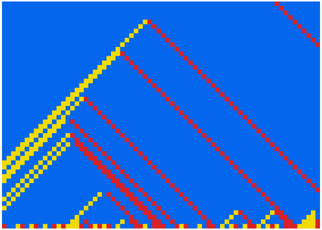
- *A wall is a sequence $W = (u_i)$ of words such that :*
  1. $|u_0| \geq |u_1| = |u_2| = \cdots$
  2. $\forall c \in [u_0], \forall t : G^t_{\mathcal{A}}(c) \in [u_t]$
- *This definition implies that $W$ is ultimately periodic.*
- *A brick of $W$ is any word $u_n$ in the period.*

## Theorem (Bricks theorem)

***If*** *$\mathcal{A}$ is a CA with neighbourhood size $k$ and having a brick of size $\geq k$* ***then*** *$L_\mu(\mathcal{A}) = \{$bricks of $\mathcal{A}\}$.*

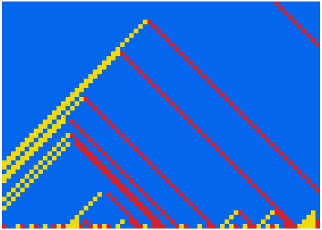
- **Corollary:**
  - $\mu$-limit sets of such CA do not depend on $\mu$ (within Bernouilli $\mu$).
- **Known fact:** no brick of size $\geq k \iff$ sensitive to initial conditions
- *Bricks theorem true for sensitive CA?*

# Example 2: the 'Just Glider' automaton



- **Definition:**
  - states move right at full speed in a background
  - states move left at full speed in a background
  - when and meet they disappear

# Example 2: the 'Just Glider' automaton



- **Definition:**
  - states move right at full speed in a background
  - states move left at full speed in a background
  - when and meet they disappear
- **Properties:**
  1. $\Omega_{\mathcal{A}} = \{$configurations without any on the left of a $\}$
  2. $\mu(\ ) = \mu(\ ) \iff \mathcal{A}$ is $\mu$-quasi-nilpotent

# Complexity of $\mu$-limit sets?

# Complexity of $\mu$-limit sets?

- **4 evidences of high complexity:**

**Theorem (Not r.e.)**

*The set of $\mu$-quasi-nilpotent CA is not recursively enumerable.*

# Complexity of $\mu$-limit sets?

- **4 evidences of high complexity:**

**Theorem (Not r.e.)**

*The set of $\mu$-quasi-nilpotent CA is not recursively enumerable.*

**Theorem (Not co-r.e.)**

*The set of $\mu$-quasi-nilpotent CA is not co-recursively enumerable.*

# Complexity of $\mu$-limit sets?

- **4 evidences of high complexity:**

**Theorem (Not r.e.)**

*The set of $\mu$-quasi-nilpotent CA is not recursively enumerable.*

**Theorem (Not co-r.e.)**

*The set of $\mu$-quasi-nilpotent CA is not co-recursively enumerable.*

**Theorem**

*It is undecidable to know whether a given word is persistent for a given CA.*

# Complexity of $\mu$-limit sets?

- **4 evidences of high complexity:**

**Theorem (Not r.e.)**

*The set of $\mu$-quasi-nilpotent CA is not recursively enumerable.*

**Theorem (Not co-r.e.)**

*The set of $\mu$-quasi-nilpotent CA is not co-recursively enumerable.*

**Theorem**

*It is undecidable to know whether a given word is persistent for a given CA.*

**Theorem**

*There is a CA with a non recursive persistent language.*

# Proof sketch of theorem **"Not r.e."**

- **Basic idea:**
  - For any Turing machine $M$, we construct a CA $\mathcal{A}$ such that:
    1. $\mathcal{A}$ simulates $M$
    2. $\mathcal{A}$ has an inalterable state '#'
    3. $M$ doesn't halt on the empty input **iff** the set of bricks of $\mathcal{A}$ is $\#^*$.

# Proof sketch of theorem "**Not r.e.**"

- **Basic idea:**
  - For any Turing machine *M*, we construct a CA $\mathcal{A}$ such that:
    1. $\mathcal{A}$ simulates *M*
    2. $\mathcal{A}$ has an inalterable state '#'
    3. *M* doesn't halt on the empty input **iff** the set of bricks of $\mathcal{A}$ is $\#^*$.
  - The bricks theorem concludes the proof

# Proof sketch of theorem "**Not r.e.**"

- **Basic idea:**
  - For any Turing machine $M$, we construct a CA $\mathcal{A}$ such that:
    1. $\mathcal{A}$ simulates $M$
    2. $\mathcal{A}$ has an inalterable state '#'
    3. $M$ doesn't halt on the empty input **iff** the set of bricks of $\mathcal{A}$ is $\#^*$.
  - The bricks theorem concludes the proof

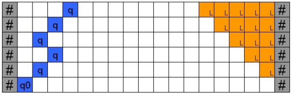
- **Simulation mechanism:**
  1. correct simulations occur on segments delimited by two '#'
  2. simulation for a limited duration
  3. a final state restart the simulation on the same segment
  4. the segment is destroyed (turn into #s) when:
     - the simulation time is over
     - an incorrect encoding is detected

# If *M* halts...

- **start:** sufficiently large segment

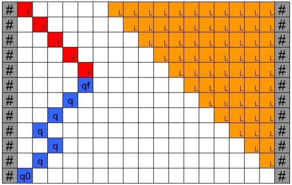
# If *M* halts...

- **start:** sufficiently large segment
- **during simulation:** a *L* is generated permanently on the rightmost cell and moves left.

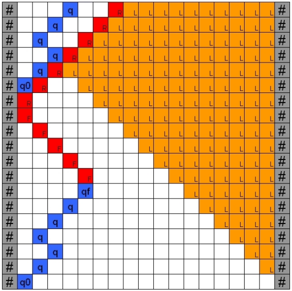


# If *M* halts...

- **start:** sufficiently large segment
- **during simulation:** a $L$ is generated permanently on the rightmost cell and moves left.
- **a final state appears:** it turns into $F$ which moves left

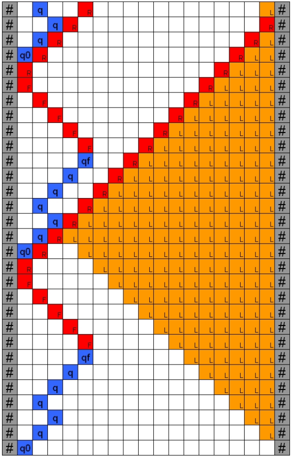


# If *M* halts...



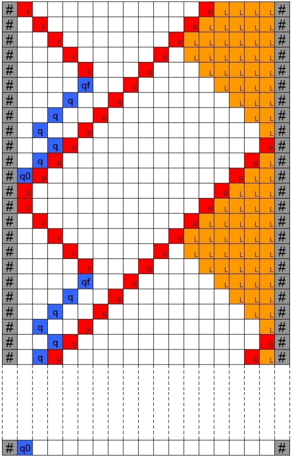
- **start:** sufficiently large segment
- **during simulation:** a *L* is generated permanently on the rightmost cell and moves left.
- **a final state appears:** it turns into *F* which moves left
- *F* **reaches the left #:** it turns into *R* which moves right to restart the computation and destroy the *L* signals.

# If *M* halts...



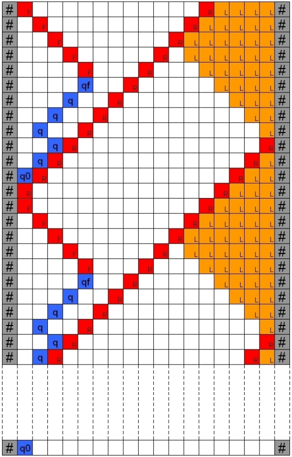
- **start:** sufficiently large segment
- **during simulation:** a *L* is generated permanently on the rightmost cell and moves left.
- **a final state appears:** it turns into *F* which moves left
- *F* **reaches the left #:** it turns into *R* which moves right to restart the computation and destroy the *L* signals.
- *R* **reaches the right #:** destroyed.

# If $M$ halts...



- **start:** sufficiently large segment
- **during simulation:** a $L$ is generated permanently on the rightmost cell and moves left.
- **a final state appears:** it turns into $F$ which moves left
- $F$ **reaches the left #:** it turns into $R$ which moves right to restart the computation and destroy the $L$ signals.
- $R$ **reaches the right #:** destroyed.
- From now on, the behavior is periodic

# If $M$ halts...




- **start:** sufficiently large segment
- **during simulation:** a $L$ is generated permanently on the rightmost cell and moves left.
- **a final state appears:** it turns into $F$ which moves left
- $F$ **reaches the left #:** it turns into $R$ which moves right to restart the computation and destroy the $L$ signals.
- $R$ **reaches the right #:** destroyed.
- From now on, the behavior is periodic

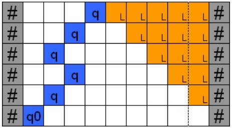
▶ **this gives a wall with a brick not in $\#^*$**

# If *M* doesn't halt...

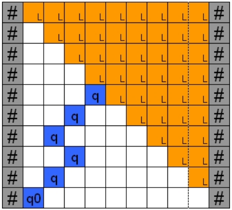
- **start:** any empty segment

# If *M* doesn't halt...

- **start:** any empty segment
- **during simulation:** a *L* is generated permanently on the rightmost cell and moves left.

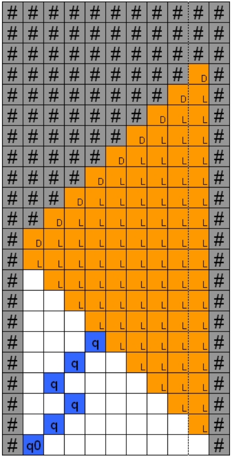


# If *M* doesn't halt...



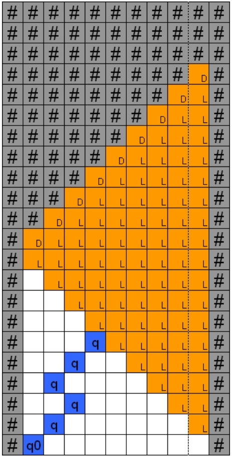

- **start:** any empty segment
- **during simulation:** a *L* is generated permanently on the rightmost cell and moves left.
- ***L* reaches the simulation head:** the head is erased.

# If *M* doesn't halt...



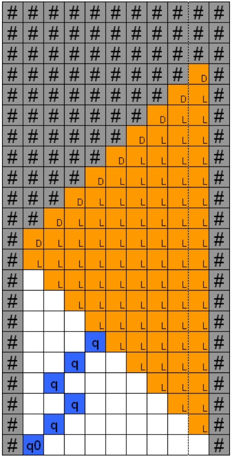
- **start:** any empty segment
- **during simulation:** a *L* is generated permanently on the rightmost cell and moves left.
- ***L* reaches the simulation head:** the head is erased.
- ***L* reaches the right #:** it turns into *D* which moves right to turn everything into #.

# If *M* doesn't halt...



- **start:** any empty segment
- **during simulation:** a *L* is generated permanently on the rightmost cell and moves left.
- ***L* reaches the simulation head:** the head is erased.
- ***L* reaches the right #:** it turns into *D* which moves right to turn everything into #.
- similar mechanism for non-empty segments (details skipped)

# If *M* doesn't halt...




- **start:** any empty segment
- **during simulation:** a *L* is generated permanently on the rightmost cell and moves left.
- ***L* reaches the simulation head:** the head is erased.
- ***L* reaches the right #:** it turns into *D* which moves right to turn everything into #.
- similar mechanism for non-empty segments (details skipped)

▶ **no other brick than #$^*$**

# About proofs of the other theorems

**Theorem (Not co-r.e.)**

*The set of $\mu$-quasi-nilpotent CA is not co-recursively enumerable.*

**Theorem**

*It is undecidable to know whether a given word is persistent for a given CA.*

**Theorem**

*There is a CA with a non recursive persistent language.*

---

- similar ingredients
- **"Not co-r.e."** is more tricky (details in article)
- results true in higher dimension
- proofs may be adapted for a fixed states set

# Future work

- is there a Rice theorem for $\mu$-limit sets?
- knowing more about the sequences $(\mu_t(u))$
  - oscillations?
  - convergence?
- are $\mu$-limit sets of $\mathcal{A}$ and $\mathcal{A}^t$ the same?
- cases of equality between limit set and $\mu$-limit set?
  - equality holds for surjective CA
- interesting sub-classes of CA where $\mu$-limit sets are tractable?
- generalising to a broader class of measures (e.g. Markov measures)