Stabilising shifts of finite type with cellular automata

Joint work with

Nazim Fatès and Irène Marcovici

Siamak Taati

Department of Mathematics, American University of Beirut

Complexity of Simple Dynamical Systems
CIRM, Marseille, February 2024

Time lapse of a wound healing

Day 1

Day 16

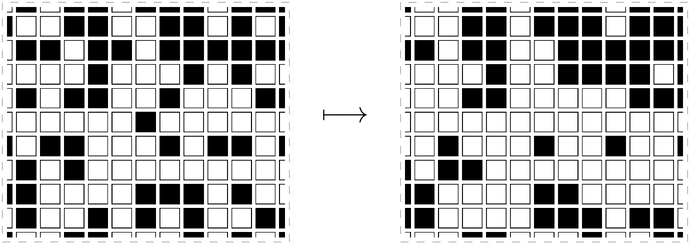
Day 33

Source: `https://youtu.be/YDmn0iZ5vhc`

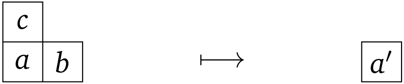
Primitive “healing” in a cellular automaton

Toom’s NEC-majority CA

A two-dimensional binary CA



Local rule:

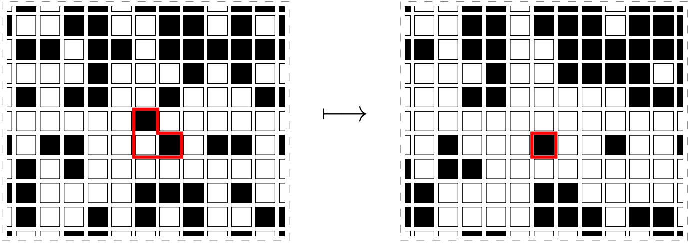


$$a' := \mathrm{maj}(a, b, c)$$

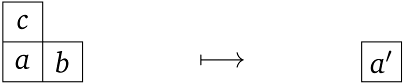
Primitive “healing” in a cellular automaton

Toom’s NEC-majority CA

A two-dimensional binary CA



Local rule:

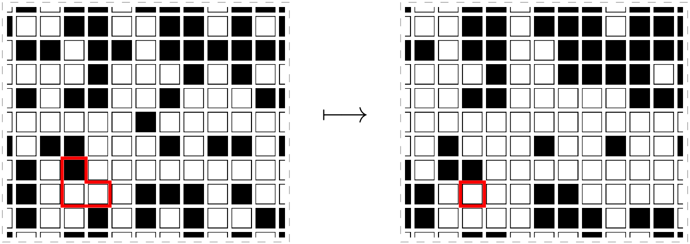


$$a' := \mathrm{maj}(a, b, c)$$

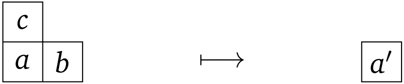
Primitive “healing” in a cellular automaton

Toom’s NEC-majority CA

A two-dimensional binary CA



Local rule:

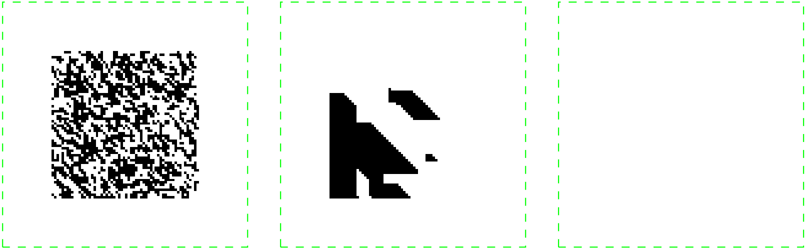


$a' := \mathrm{maj}(a, b, c)$

Primitive “healing” in a cellular automaton

Toom’s NEC-majority CA

Time lapse of Toom’s CA “healing”

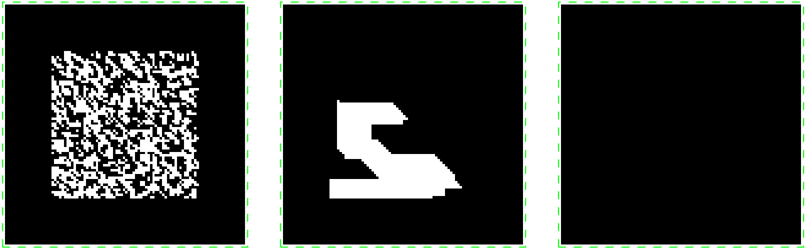


A finite perturbation of all-□ | After 30 iterations | After 120 iterations

Primitive “healing” in a cellular automaton

Toom’s NEC-majority CA

Time lapse of Toom’s CA “healing”

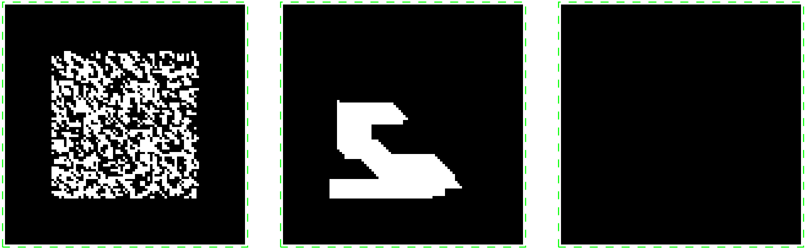


A finite perturbation of all-■ | After 30 iterations | After 120 iterations

Primitive "healing" in a cellular automaton

Toom's NEC-majority CA

Time lapse of Toom's CA "healing"



A finite perturbation of all-■ | After 30 iterations | After 120 iterations

Toom's CA is self-stabilising:

- Two "legal" configurations: all-□ and all-■
- The "legal" configurations remain unchanged.
- Finite perturbations of "legal" configurations rapidly "heal".

Self-stabilisation

Question

Can we design self-stabilising CA with more complex sets of legal configurations?*

* prescribed using finitely many local constraints (i.e., an SFT)

Motivation

- Fault-tolerance (robustness against random noise)
- Robustness against tampering by an adversary
- Self-healing materials (?)
- Symbolic dynamics [a notion of "complexity" for SFTs]

Outline

- Formulation
- Efficient solutions for some examples of local constraints
  - Deterministic solutions
  - (Probabilistic solutions)
  - An example which appears difficult
- Time complexity
  - Invariance under conjugacy
  - An example with hard self-stabilisation
- (Self-stabilisation starting from random perturbations)

Formulation

Self-stabilisation

space of legal configurations

We say that a CA F stabilises an SFT X if

(i) Every element of X is a fixed point of F.

[i.e., the CA keeps each legal configuration unchanged.]

$$x \in X \qquad \Longrightarrow \qquad F(x) = x$$

(ii) Starting from any finite perturbation of an element of X, the CA returns to X in finitely many steps.

[i.e., the CA "heals" any finite perturbation of a legal configuration.]

$$\tilde{x} \sim x \in X \qquad \Longrightarrow \qquad F^t(\tilde{x}) \in X \quad \text{for some } t \in \mathbb{N}$$

$\tilde{x}$ is a finite perturbation of x

The smallest such t is called the recovery time of $\tilde{x}$.

Formulation

Self-stabilisation

We say that a CA F stabilises an SFT X if

(i) Every element of X is a fixed point of F.

(ii) Starting from any finite perturbation of an element of X, the CA returns to X in finitely many steps.

Example (Toom's NEC-majority CA)

$$X = \{\text{all-}\square, \text{all-}\blacksquare\}$$

Formulation

Self-stabilisation

We say that a CA F stabilises an SFT X if

(i) Every element of X is a fixed point of F.

(ii) Starting from any finite perturbation of an element of X, the CA returns to X in finitely many steps.

Example (Toom's NEC-majority CA)

$$X = \{\text{all-}\square, \text{all-}\blacksquare\}$$

Remark

The alphabet of F may be strictly larger than the alphabet of X. The perturbations are in the alphabet of F.

Formulation

Self-stabilisation

We say that a CA F stabilises an SFT X if

(i) Every element of X is a fixed point of F.

(ii) Starting from any finite perturbation of an element of X, the CA returns to X in finitely many steps.

Example (Toom's NEC-majority CA)

$$X = \{\text{all-}\square, \text{all-}\blacksquare\}$$

Question

Which SFTs can be (efficiently) stabilised by CAs?

Formulation

Question

Which SFTs can be (efficiently) stabilised by CAs?

Efficiency

What counts as "efficiency"?

- Speed of stabilisation [i.e., recovery time]
- Number of extra symbols
- Neighbourhood radius [linear trade-off with speed]

Example

Toom's CA stabilises $X = \{\text{all-}\square, \text{all-}\blacksquare\}$ very efficiently:

- Linear recovery time [... in the diameter of the perturbed region]
- No extra symbols
- Neighbourhood radius 1

Formulation

Question

Which SFTs can be (efficiently) stabilised by CAs?

Efficiency

What counts as "efficiency"?

- Speed of stabilisation [i.e., recovery time]
- Number of extra symbols
- Neighbourhood radius [linear trade-off with speed]

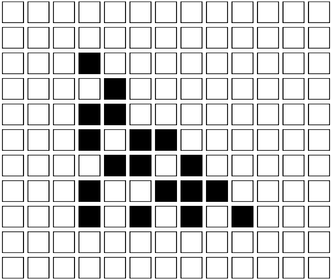
Example

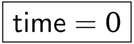
Toom's CA stabilises $X = \{\text{all-}\square, \text{all-}\blacksquare\}$ very efficiently:

- Linear recovery time [... in the diameter of the perturbed region]
- No extra symbols
- Neighbourhood radius 1

Back to Toom's CA

Mechanism of stabilisation

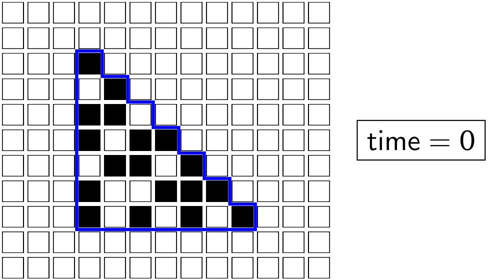




A finite perturbation of the all-□ configuration

Back to Toom's CA

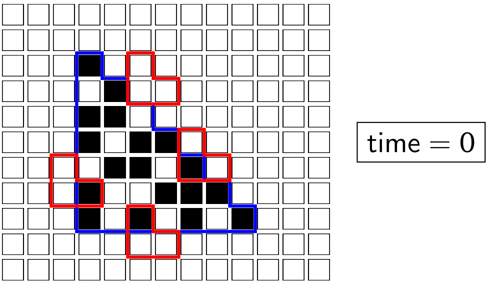
Mechanism of stabilisation



A finite perturbation of the all-□ configuration

Back to Toom's CA

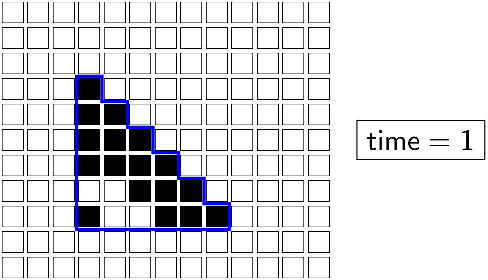
Mechanism of stabilisation



A finite perturbation of the all-□ configuration

Back to Toom's CA

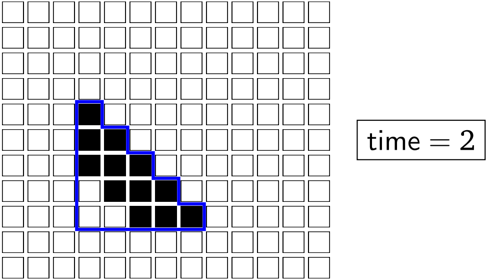
Mechanism of stabilisation



A finite perturbation of the all-□ configuration

Back to Toom's CA

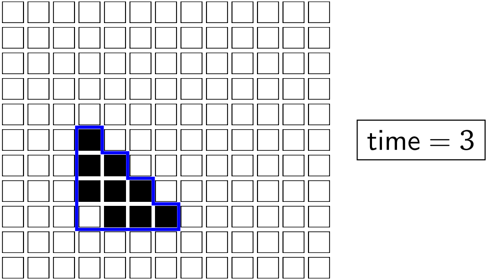
Mechanism of stabilisation



A finite perturbation of the all-□ configuration

Back to Toom's CA

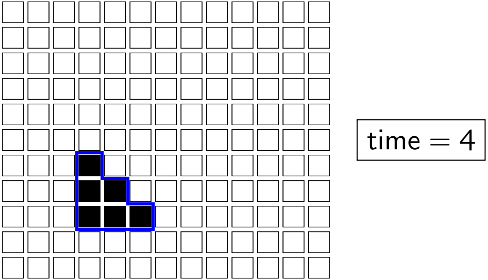
Mechanism of stabilisation



A finite perturbation of the all-□ configuration

Back to Toom's CA

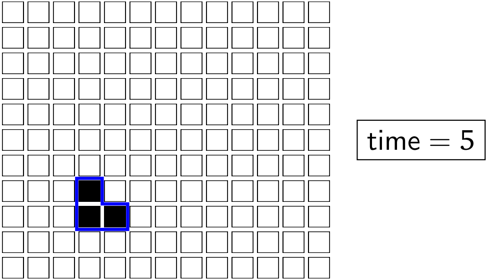
Mechanism of stabilisation



A finite perturbation of the all-□ configuration

Back to Toom's CA

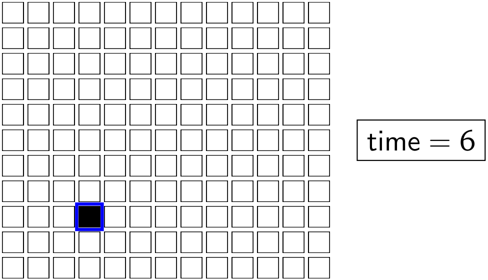
Mechanism of stabilisation



A finite perturbation of the all-□ configuration

Back to Toom's CA

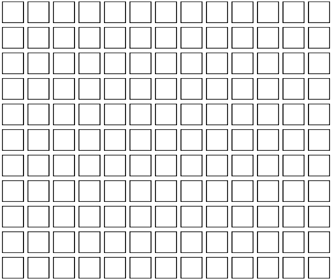
Mechanism of stabilisation



A finite perturbation of the all-□ configuration

Back to Toom's CA

Mechanism of stabilisation

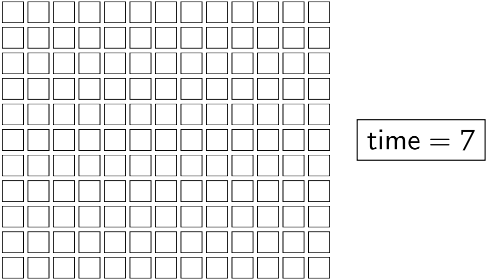


time = 7

A legal configuration is reached!

Back to Toom's CA

Mechanism of stabilisation



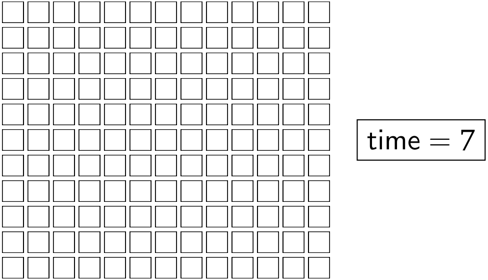
A legal configuration is reached!

Proposition (Linear recovery)

If the perturbed region fits in a triangle of size ℓ*, then the recovery time is at most* ℓ*.*

Back to Toom's CA

Mechanism of stabilisation



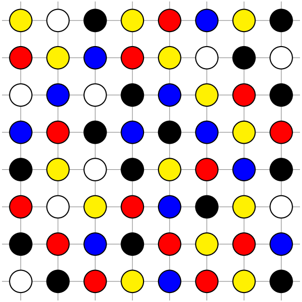
A legal configuration is reached!

Proposition (Linear recovery)

If the perturbed region fits in a triangle of size ℓ, then the recovery time is at most ℓ.

By symmetry: the same holds for any finite perturbation of the all-■ configuration.

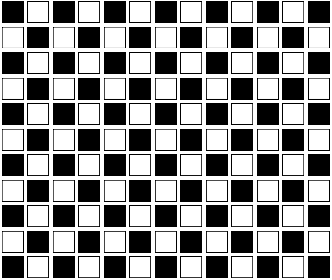
A prototypical example: k-colourings



$X =$ all valid k-colourings of the lattice

A prototypical example: k-colourings

Case: $k = 2$

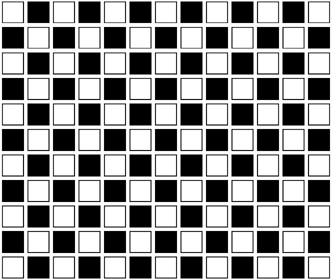


The even checkerboard

- Only two legal configurations: the even and odd checkerboards

A prototypical example: k-colourings

Case: $k = 2$

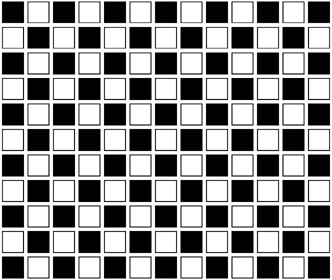


The odd checkerboard

- Only two legal configurations: the even and odd checkerboards

A prototypical example: k-colourings

Case: $k = 2$

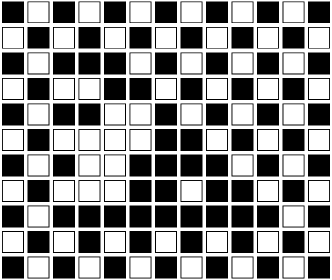


The even checkerboard

- Only two legal configurations: the even and odd checkerboards

A prototypical example: k-colourings

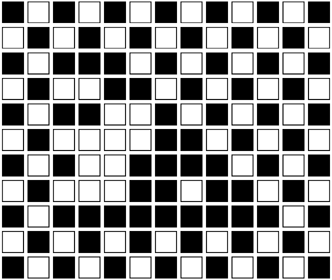
Case: $k = 2$



A finite perturbation of the even checkerboard

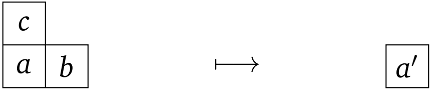
A prototypical example: k-colourings

Case: $k = 2$



A finite perturbation of the even checkerboard

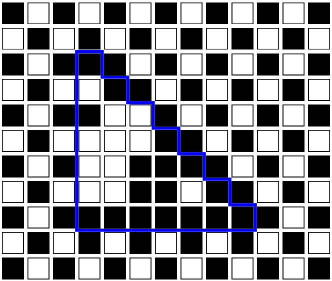
A simple solution based on Toom's CA



$$a' := \mathrm{maj}(a, \bar{b}, \bar{c})$$

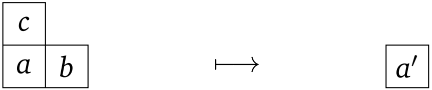
A prototypical example: k-colourings

Case: $k = 2$



A finite perturbation of the even checkerboard

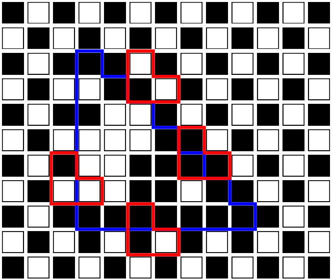
A simple solution based on Toom's CA



$$a' := \mathrm{maj}(a, \bar{b}, \bar{c})$$

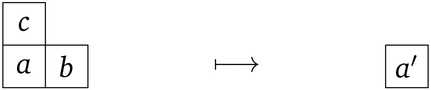
A prototypical example: k-colourings

Case: $k = 2$



A finite perturbation of the even checkerboard

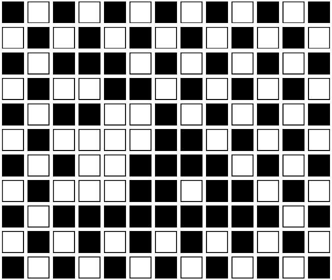
A simple solution based on Toom's CA



$$a' := \mathrm{maj}(a, \bar{b}, \bar{c})$$

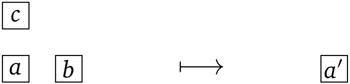
A prototypical example: k-colourings

Case: $k = 2$



A finite perturbation of the even checkerboard

An alternative simple solution based on Toom's CA



$$a' := \mathrm{maj}(a, b, c)$$

[Apply Toom's CA on four sublattices separately!]

Inspired by 2-colourings

More generally:

Proposition

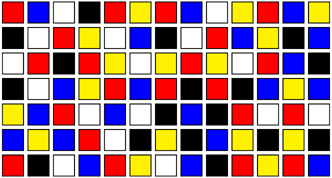
Let X be a finite two-dimensional SFT.
There exists a CA without additional symbols that stabilises X in linear time.

Idea: Pick $p, q \in \mathbb{N}$ such that X is horizontally p-periodic and vertically q-periodic. Apply Toom's CA* on each (p, q)-sublattice.

* If all three symbols are different, leave unchanged.

A prototypical example: k-colourings

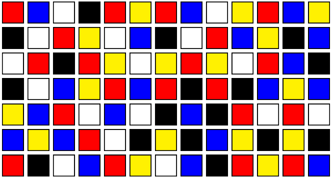
Case: $k \geq 5$



A valid 5-colouring

A prototypical example: k-colourings

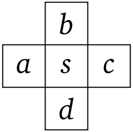
Case: $k \geq 5$



A valid 5-colouring

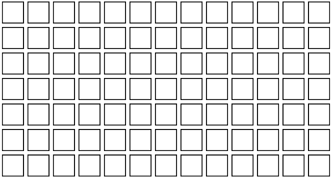
Key property: single-site fillability

For every choice of colours a, b, c, d, there is a matching colour $s := \psi(a, b, c, d)$ for the center.



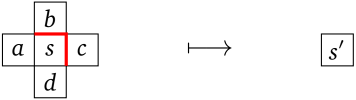
A prototypical example: k-colourings

Case: $k \geq 5$



A finite perturbation of a 5-colouring

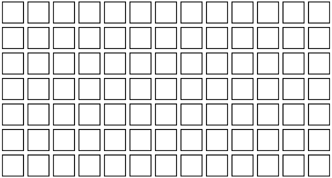
A solution based on Toom's CA



$$s' := \begin{cases} \psi(a,b,c,d) & \text{if } s \text{ does not match upwards or rightwards,} \\ s & \text{otherwise.} \end{cases}$$

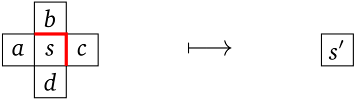
A prototypical example: k-colourings

Case: $k \geq 5$



A finite perturbation of a 5-colouring

A solution based on Toom's CA

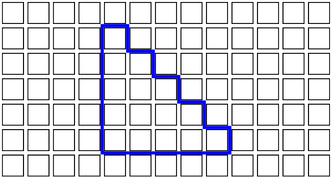


$$s' := \begin{cases} \psi(a,b,c,d) & \text{if } s \text{ does not match upwards or rightwards,} \\ s & \text{otherwise.} \end{cases}$$

Note: No new *NE*-defects are created!

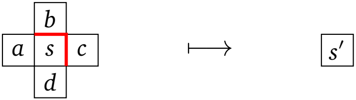
A prototypical example: k-colourings

Case: $k \geq 5$



A finite perturbation of a 5-colouring

A solution based on Toom's CA

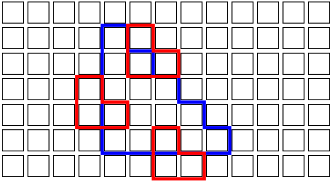


$$s' := \begin{cases} \psi(a,b,c,d) & \text{if } s \text{ does not match upwards or rightwards,} \\ s & \text{otherwise.} \end{cases}$$

<u>Note</u>: No new *NE*-defects are created!

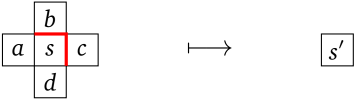
A prototypical example: k-colourings

Case: $k \geq 5$



A finite perturbation of a 5-colouring

A solution based on Toom's CA



$$s' := \begin{cases} \psi(a,b,c,d) & \text{if } s \text{ does not match upwards or rightwards,} \\ s & \text{otherwise.} \end{cases}$$

Note: No new *NE*-defects are created!

Inspired by k-colourings for $k \geq 5$

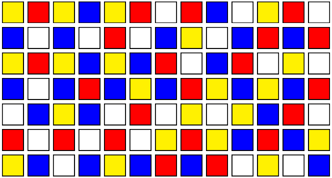
More generally:

Proposition

Let X be a single-site fillable two-dimensional n.n. SFT. There exists a CA without additional symbols that stabilises X in linear time.

A prototypical example: k-colourings

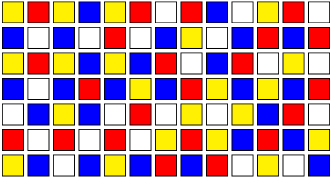
Case: $k = 4$



A valid 4-colouring

A prototypical example: k-colourings

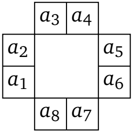
Case: $k = 4$



A valid 4-colouring

Key property: strong 2-fillability

For every (not necessarily valid) choice of $a_1, a_2, \ldots, a_8$, there is a matching colouring of the central 2×2 block.



Inspired by 4-colourings

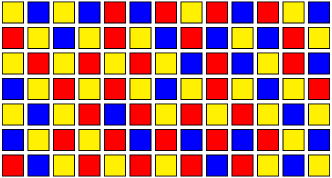
Proposition

Let X be a strongly ℓ-fillable two-dimensional n.n. SFT. There exists a CA without additional symbols that stabilises X in quadratic time.

Idea: The CA *locally* identifies a non-empty subset of non-adjacent faulty $\ell \times \ell$ blocks and corrects them.
In this fashion, at every step, the number of faulty $\ell \times \ell$ blocks decreases by at least 1.

A prototypical example: k-colourings

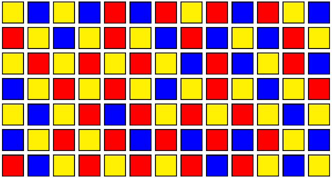
Case: $k = 3$



A valid 3-colouring

A prototypical example: k-colourings

Case: $k = 3$

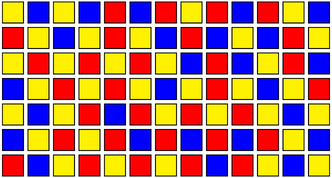


A valid 3-colouring

We are stuck!!

A prototypical example: k-colourings

Case: $k = 3$



A valid 3-colouring

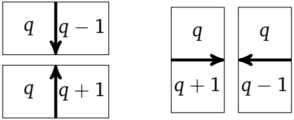
We are stuck!!

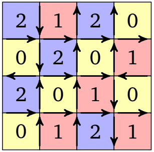
Question

Is there a CA that stabilises 3-colourings?

Why are 3-colourings difficult to stabilise?

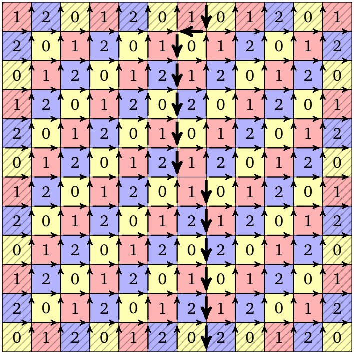
Connection with the six-vertex model





Six-vertex model: Each vertex will have exactly two incoming arrows and two outgoing arrows.

Why are 3-colourings difficult to stabilise?



A finite perturbation of a valid 3-colouring

The difficulty:

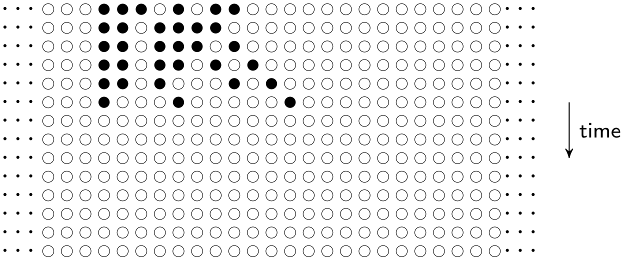
There are only two defects, but correcting them requires changing the colour of a large number of sites.

One-dimensional SFTs

One-dimensional SFTs

Example (GKL)

$$F(x)_i := \begin{cases} \mathrm{maj}(x_{i-3}, x_{i-1}, x_i) & \text{if } x_i = \circ, \\ \mathrm{maj}(x_i, x_{i+1}, x_{i+3}) & \text{if } x_i = \bullet, \end{cases}$$



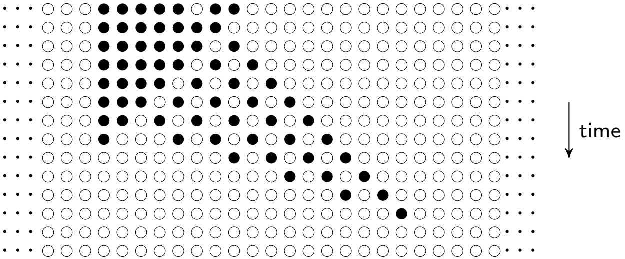
Proposition (Gács, Kurdyumov, Levin, 1977)

The GKL CA stabilises $X = \{$all-$\circ$, all-$\bullet\}$ in linear time.

One-dimensional SFTs

Example (GKL)

$$F(x)_i := \begin{cases} \mathrm{maj}(x_{i-3}, x_{i-1}, x_i) & \text{if } x_i = \circ, \\ \mathrm{maj}(x_i, x_{i+1}, x_{i+3}) & \text{if } x_i = \bullet, \end{cases}$$



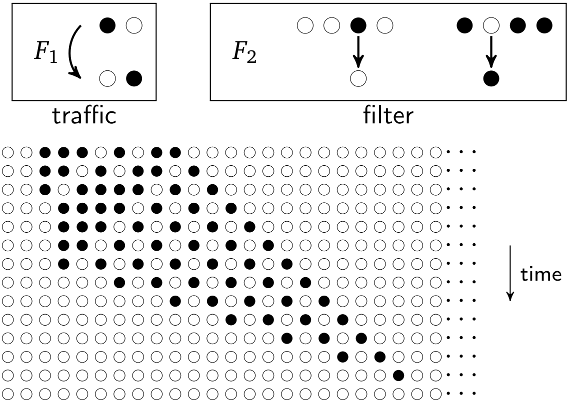
Proposition (Gács, Kurdyumov, Levin, 1977)

The GKL CA stabilises $X = \{$all-$\circ$, all-$\bullet\}$ in linear time.

One-dimensional SFTs

Example (Modified Traffic)

$F = F_2 F_1$



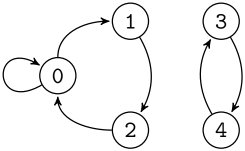
Proposition (Kari and Le Gloanec, 2012)

The modified traffic CA stabilises $X = \{$*all-*$\circ$*, all-*$\bullet\}$ *in linear time.*

One-dimensional SFTs

Theorem

For every non-wandering one-dimensional SFT X, there exists a CA F (with extra symbols) that stabilises X in linear time.

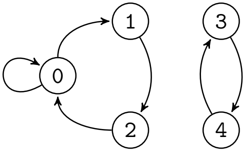


An example of a non-wandering SFT

One-dimensional SFTs

Theorem

For every non-wandering one-dimensional SFT X, there exists a CA F (with extra symbols) that stabilises X in linear time.



An example of a non-wandering SFT

Remark

There is a more sophisticated solution by Ilkka Törmä which does not require extra symbols and works for every (not just non-wandering) SFT.

One-dimensional SFTs

Theorem

For every non-wandering one-dimensional SFT X, there exists a CA F (with extra symbols) that stabilises X in linear time.

Idea: There is a simple sequential procedure for correcting defects from left to right.

Difficulty: The CA cannot identify the left-most defect to start such a procedure.

. . .

Back to two dimensions

Back to two dimensions

Question

Can a CA stabilise an aperiodic SFT?

Back to two dimensions

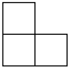
Question

Can a CA stabilise an aperiodic SFT?

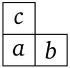
Answer: Yes!

Deterministic two-dimensional SFTs

NE-deterministic SFTs



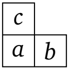
shape of forbidden patterns



at most one symbol a consistent with each pair b, c

Example (Ledrappier's SFT)

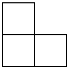
There are two symbols 0 and 1. The forbidden patterns are



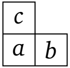
where $a \neq b + c \pmod{2}$.

Deterministic two-dimensional SFTs

NE-deterministic SFTs

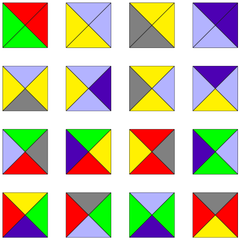


shape of forbidden patterns



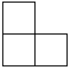
at most one symbol a consistent with each pair b, c

Example (Ammann's aperiodic tile set)

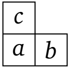


Deterministic two-dimensional SFTs

NE-deterministic SFTs



shape of forbidden patterns



at most one symbol a
consistent with each pair b, c

Theorem

For every two-dimensional NE-deterministic SFT X, there exists a CA F (with extra symbols) that stabilises X in linear time.

Difficulty: Naïvely applying the deterministic rule doesn't work.

Idea: Similar to the one-dimensional SFT.

Time complexity of stabilisation

Theorem (Invariance under conjugacy)

Suppose X and Y are conjugate SFTs. If there is a CA that stabilises X in time $\tau(n)$, then there also exists a CA that stabilises Y in time $\tau\big(n + O(1)\big)$.

Time complexity of stabilisation

Theorem (Invariance under conjugacy)

Suppose X *and* Y *are conjugate SFTs. If there is a CA that stabilises* X *in time* $\tau(n)$*, then there also exists a CA that stabilises* Y *in time* $\tau\big(n + O(1)\big)$*.*

The "best" recovery time for an SFT X can be thought of as a measure of the "local complexity" of X.

[Reminiscent of logical depth (Bennett, 1982)?]

Convention

If an SFT has no stabilising CA, we define its "best" recover time to be ∞.

Time complexity of self-stabilisation

Example

The "best" recovery time of some classes of SFTs:

- 1d SFT: (at most) linear.
- 2d k-colourings with $k = 2$ or $k \geq 5$: linear.
- 2d 4-colourings: (at most) quadratic.
- 2d 3-colourings: unknown
- Deterministic SFT: (at most) linear

Time complexity of self-stabilisation

Example

The "best" recovery time of some classes of SFTs:

- 1d SFT: (at most) linear.
- 2d k-colourings with $k = 2$ or $k \geq 5$: linear.
- 2d 4-colourings: (at most) quadratic.
- 2d 3-colourings: unknown
- Deterministic SFT: (at most) linear

Any negative result?

Time complexity of self-stabilisation

Example

The "best" recovery time of some classes of SFTs:

- 1d SFT: (at most) linear.
- 2d k-colourings with $k = 2$ or $k \geq 5$: linear.
- 2d 4-colourings: (at most) quadratic.
- 2d 3-colourings: unknown
- Deterministic SFT: (at most) linear

Any negative result?

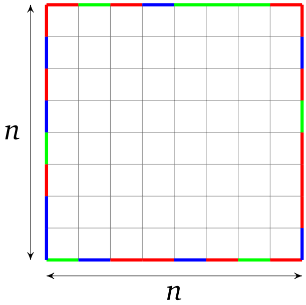
Theorem (Super-polynomial hardness)

Unless $\mathbf{P} = \mathbf{NP}$*, there exists a two-dimensional SFT* X *which cannot be stabilised by any CA in polynomial time.*

Super-polynomial hardness

Square tiling problem of a set Θ of Wang tiles

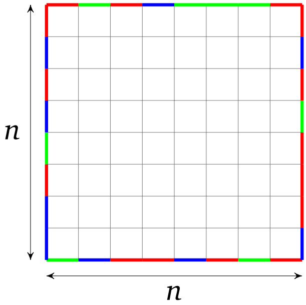
Given n and a prescribed colouring of the boundary of an $n \times n$ square, is there an admissible colouring of the square?



Super-polynomial hardness

Square tiling problem of a set Θ of Wang tiles

Given n and a prescribed colouring of the boundary of an $n \times n$ square, is there an admissible colouring of the square?



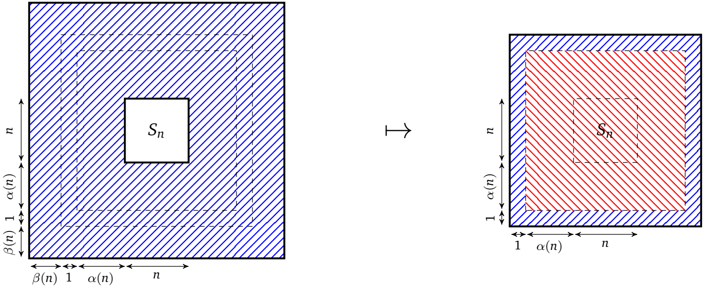
Proposition (Folklore)

There exists a tile set for which the square tiling problem is $\mathbf{NP}$*-complete.*

Super-polynomial hardness

A CA stabilising X_Θ can be used to solve a variant of the square tiling problem (with only polynomial overhead):

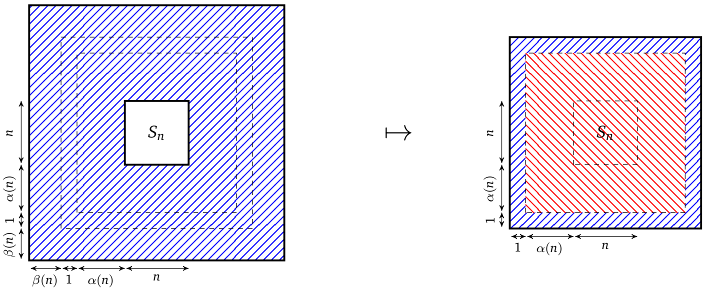
Global tiling patching problem (associated to Θ, α, β)



Super-polynomial hardness

A CA stabilising X_Θ can be used to solve a variant of the square tiling problem (with only polynomial overhead):

Global tiling patching problem (associated to Θ, α, β)



Proposition

There exists a tile set Θ such that for every $\alpha, \beta : \mathbb{N} \to \mathbb{N}$ with polynomial growth, the global tiling patching problem associated to Θ, α, β is $\mathbf{NP}$*-hard.*

Self-stabilisation starting from random perturbations

Self-stabilisation starting from random perturbations

Formulation

#*&!@??!*#&???! ...

Self-stabilisation starting from random perturbations

Formulation

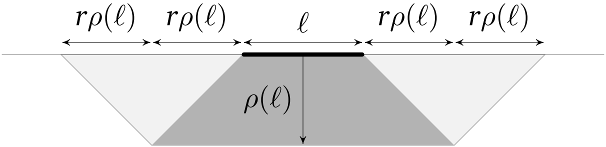
#*&!@??!*#&???! ...

Theorem

Suppose that a CA F stabilises an SFT X in sub-quadratic time. Then, F also stabilises X starting from (sufficiently weak) random perturbations.

Self-stabilisation starting from random perturbations

Proof idea.

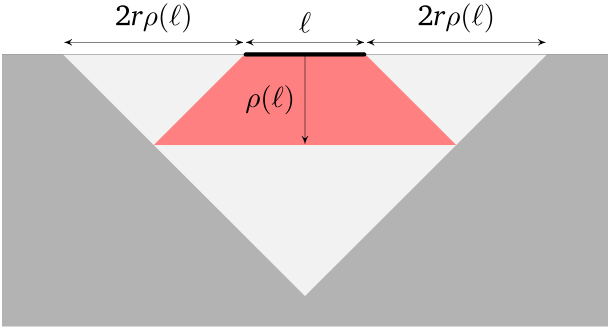


Correcting an island of length ℓ in $\rho(\ell)$ steps

r: neighbourhood radius of the CA

Self-stabilisation starting from random perturbations

Proof idea.



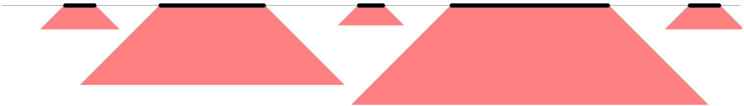
An isolated island has a sufficiently wide margin without errors

Observation

An isolated island disappears before sensing or affecting the rest of the configuration.

Self-stabilisation starting from random perturbations

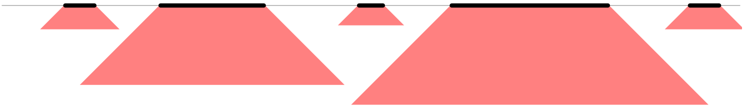
Proof idea.



A sparse set of errors can be decomposed into non-interacting islands

Self-stabilisation starting from random perturbations

Proof idea.



A sparse set of errors can be decomposed into non-interacting islands

Thus, the notion of sparseness is the key!

Self-stabilisation starting from random perturbations

Sparseness

[Gács, 1986, . . .]

Let $\rho : \mathbb{N} \to \mathbb{N}$ be a non-decreasing function.

The ρ-territory of a finite set $A \subseteq \mathbb{Z}^d$ is the set $N^\rho(A)$ of all sites that are within distance $\rho(\mathrm{diam}(A))$ from A.

A set $S \subseteq \mathbb{Z}^d$ is ρ-sparse if there is a partitioning $\mathcal{C}(S)$ of S into finite sets, called the ρ-islands of S, such that

(i) (separation) For every two distinct $A, B \in \mathcal{C}(S)$, either $A \cap N^\rho(B) = \varnothing$ or $N^\rho(A) \cap B = \varnothing$.

(ii) (thinness) Every site $k \in \mathbb{Z}^d$ is in the ρ-territory of at most finitely many ρ-islands.

Self-stabilisation starting from random perturbations

Theorem (Durand, Romashchenko, Shen, 2012)

Suppose that $\rho(\ell) = O(\ell)$. Let $\varepsilon > 0$ be sufficiently small.
Then, an ε-Bernoulli random set $\mathbf{S} \subseteq \mathbb{Z}^d$ is almost surely ρ-sparse.

Theorem (Gács, 2020)

Suppose that $\rho(\ell) = O(\ell^\beta)$ for some $\beta < 2$. Let $\varepsilon > 0$ be sufficiently small.
Then, an ε-Bernoulli random set $\mathbf{S} \subseteq \mathbb{Z}^d$ is almost surely ρ-sparse.

Open problems

Q1: Can every two-dimensional SFT be stabilised by a CA?

Q2: Is there a (polynomial-time) solution for 3-colourings?

Q3: Can 4-colourings be stabilised in sub-quadratic time?

Q4: Can a variant of the sparseness argument be applied to probabilistic self-stabilising CA?

Q5: Self-stabilisation in the presence of temporal noise

Q6: Self-organization . . . ?

. . .

Open problems

Q1: Can every two-dimensional SFT be stabilised by a CA?

Q2: Is there a (polynomial-time) solution for 3-colourings?

Q3: Can 4-colourings be stabilised in sub-quadratic time?

Q4: Can a variant of the sparseness argument be applied to probabilistic self-stabilising CA?

Q5: Self-stabilisation in the presence of temporal noise

Q6: Self-organization . . . ?

. . .

Happy 60th birthday, Jarkko!