Tilings and undecidability in cellular automata

Jarkko Kari

Department of Mathematics, University of Turku, Finland

TUCS(Turku Centre for Computer Science), Turku, Finland

Outline of the talk

- Cellular Automata: definitions
- Topolgy & Curtis-Hedlund-Lyndom -theorem
- Reversible CA
- Wang tiles and decidability questions
- Aperiodicity and undecidability
- NW-determinism & one-dimensional CA
- Snake tiles & two-dimensional CA

Cellular Automata (CA): definition

The d-dimensional cellular space is the infinite d-dimensional grid $\mathbb{Z}^d$ of **cells**.

Each cell stores a finite amount of information (represented as its **state**).

The cells change their states synchronously according to a **local update rule** that provides the next state of a cell, depending on the present state of the neighboring cells.

Precisely speaking, we have

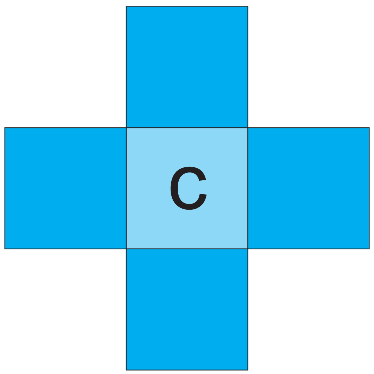
- Finite **state set** S.
- **Configurations** are elements of $S^{\mathbb{Z}^d}$, i.e., functions $\mathbb{Z}^d \longrightarrow S$ assigning states to cells,
- A **neighborhood vector**

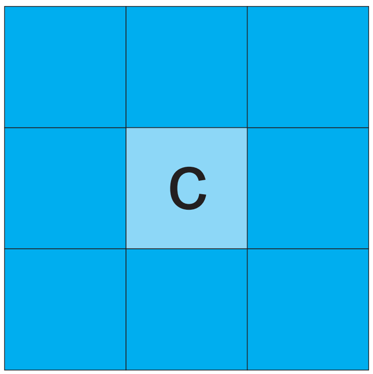
  $$N = (\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n)$$

  is a vector of n distinct elements of $\mathbb{Z}^d$ that provide the relative offsets to neighbors.
- The **neighbors** of a cell at location $\vec{x} \in \mathbb{Z}^d$ are the n cells at locations

  $$\vec{x} + \vec{x}_i, \text{ for } i = 1, 2, \ldots, n.$$

Typical two-dimensional neighborhoods:





Von Neumann neighborhood

$\{(0,0),(\pm 1,0),(0,\pm 1)\}$

Moore neighborhood

$\{-1,0,1\}\times\{-1,0,1\}$

The **local rule** is a function

$$f : S^n \longrightarrow S$$

where n is the size of the neighborhood.

State $f(a_1, a_2, \ldots, a_n)$ is the new state of a cell whose n neighbors were at states $a_1, a_2, \ldots, a_n$ one time step before.

The local update rule determines the global dynamics of the CA: Configuration c becomes in one time step the configuration e where, for all $\vec{x} \in \mathbb{Z}^d$,

$$e(\vec{x}) = f(c(\vec{x} + \vec{x}_1), c(\vec{x} + \vec{x}_2), \ldots, c(\vec{x} + \vec{x}_n)).$$

The transformation

$$G : S^{\mathbb{Z}^d} \longrightarrow S^{\mathbb{Z}^d}$$

that maps $c \mapsto e$ is the **global transition function** of the CA.

Curtis-Hedlund-Lyndom -theorem

It is convenient to endow $S^{\mathbb{Z}^d}$ with the product topology. The topology is compact and induced by a metric.

Curtis-Hedlund-Lyndom -theorem

It is convenient to endow $S^{\mathbb{Z}^d}$ with the product topology. The topology is compact and induced by a metric.

The topology is generated by the **cylinder sets**

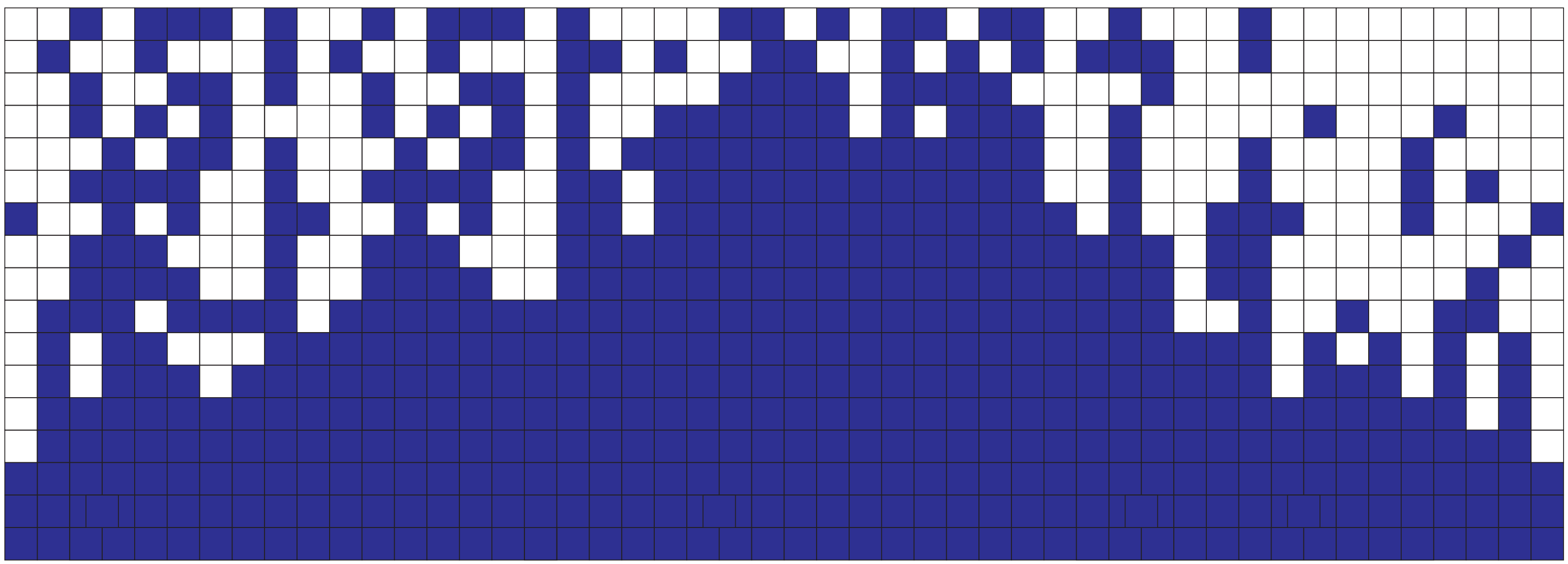
$$\mathrm{Cyl}(c, M) = \{e \in S^{\mathbb{Z}^d} \mid e(\vec{x}) = c(\vec{x}) \text{ for all } \vec{x} \in M\}$$

for $c \in S^{\mathbb{Z}^d}$ and finite $M \subset \mathbb{Z}^d$.

All cylinder sets are clopen, i.e. closed and open. Cylinders for fixed finite $M \subseteq \mathbb{Z}^d$ form a finite partitioning of $S^{\mathbb{Z}^d}$.

Under this topology, a sequence $c_1, c_2, \ldots$ of configurations **converges** to $c \in S^{\mathbb{Z}^d}$ if and only if for all cells $\vec{x} \in \mathbb{Z}^d$ and for all sufficiently large i holds

$$c_i(\vec{x}) = c(\vec{x}).$$



Compactness of the topology means that all infinite sequences $c_1, c_2, \ldots$ of configurations have converging subsequences.

All cellular automata are **continuous** transformations

$$S^{\mathbb{Z}^d} \longrightarrow S^{\mathbb{Z}^d}$$

under the topology. Indeed, locality of the update rule means that if

$$c_1, c_2, \ldots$$

is a converging sequence of configurations then

$$G(c_1), G(c_2), \ldots$$

converges as well, and

$$\lim_{i \longrightarrow \infty} G(c_i) = G(\lim_{i \longrightarrow \infty} c_i).$$

The **translation** τ determined by vector $\vec{r} \in \mathbb{Z}^d$ is the transformation

$$S^{\mathbb{Z}^d} \longrightarrow S^{\mathbb{Z}^d}$$

that maps $c \mapsto e$ where

$$e(\vec{x}) = c(\vec{x} - \vec{r}) \text{ for all } \vec{x} \in \mathbb{Z}^d.$$

(It is the CA whose local rule is the identity function and whose neighborhood consists of $-\vec{r}$ alone.)

Translations determined by unit coordinate vectors $(0, \ldots, 0, 1, 0 \ldots, 0)$ are called **shifts**

Since all cells of a CA use the same local rule, the CA commutes with all translations:

$$G \circ \tau = \tau \circ G.$$

We have seen that all CA are continuous, translation commuting maps $S^{\mathbb{Z}^d} \longrightarrow S^{\mathbb{Z}^d}$.

The **Curtis-Hedlund- Lyndon theorem** from 1969 states that also the converse is true:

Theorem: A function $G : S^{\mathbb{Z}^d} \longrightarrow S^{\mathbb{Z}^d}$ is a CA function if and only if

(i) G is continuous, and

(ii) G commutes with translations.

- The set $S^{\mathbb{Z}^d}$, together with the shift maps, is the d-dimensional **full shift**.
- Topologically closed, shift invariant subsets of $S^{\mathbb{Z}^d}$ are called **subshifts**.
- Cellular automata are the endomorphisms of the full shift.

Reversible CA

A CA is called

- **injective** if G is one-to-one,
- **surjective** if G is onto,
- **bijective** if G is both one-to-one and onto.

Reversible CA

A CA is called

- **injective** if G is one-to-one,
- **surjective** if G is onto,
- **bijective** if G is both one-to-one and onto.

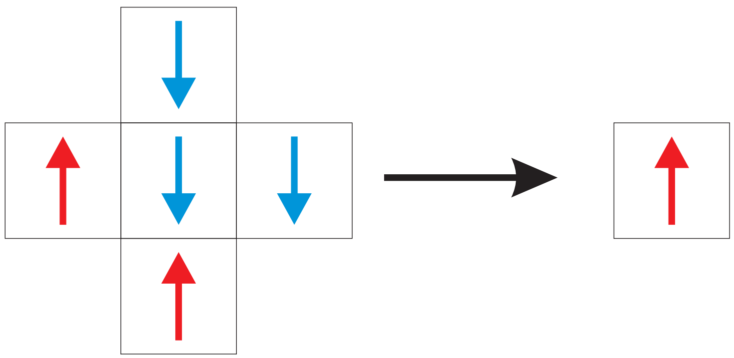
A CA G is a **reversible** (RCA) if there is another CA function F that is its inverse, i.e.

$$G \circ F = F \circ G = \text{identity function.}$$

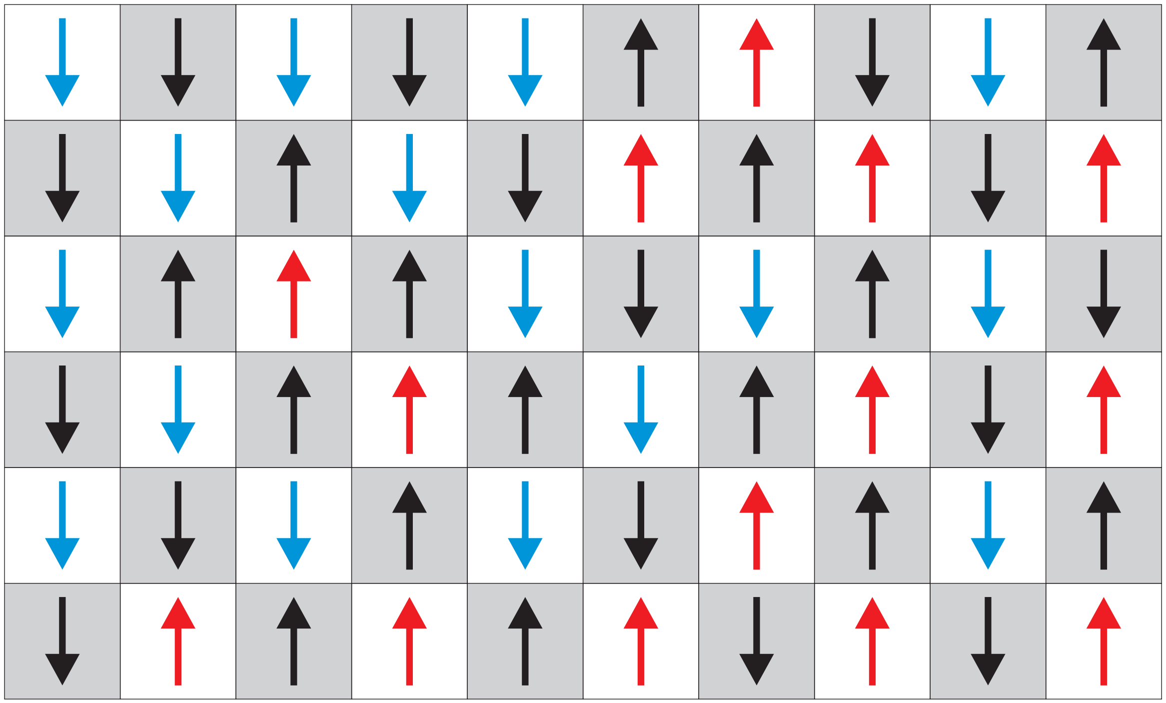
RCA G and F are called the **inverse automata** of each other.

Two-dimensional **Q2R** Ising model by G.Vichniac (1984) is an example of a reversible cellular automaton.

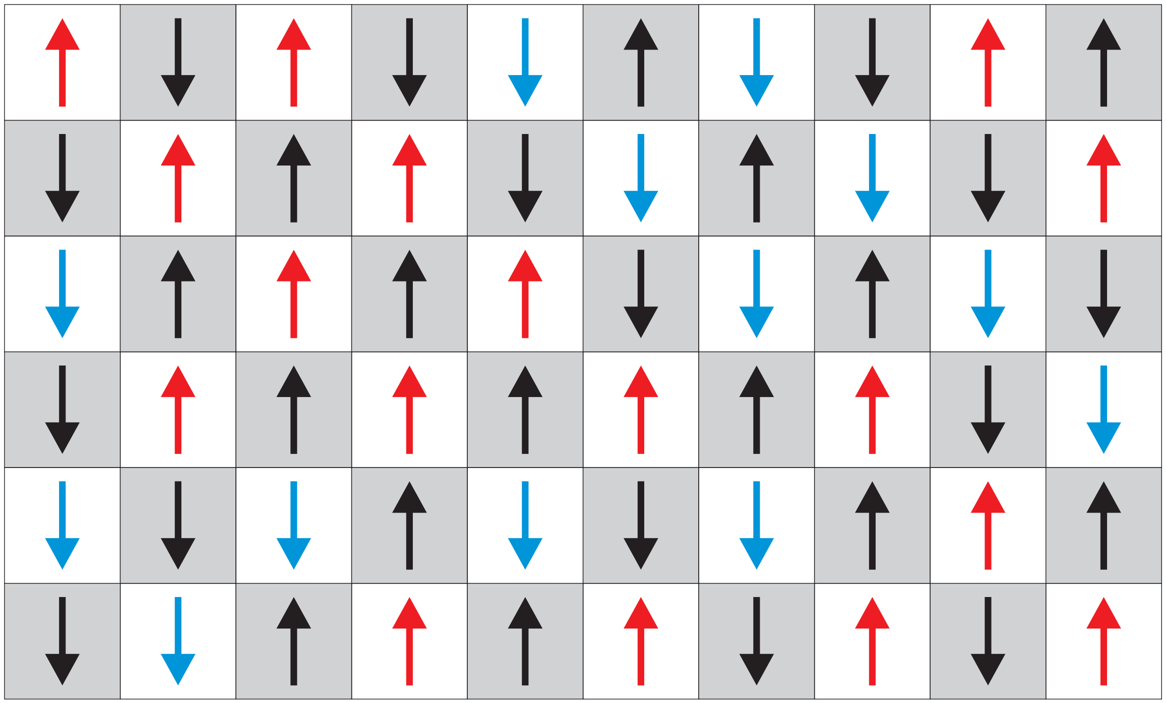
Each cell has a spin that is directed either up or down. The direction of a spin is swapped if and only if among the four immediate neighbors there are exactly two cells with spin up and two cells with spin down:



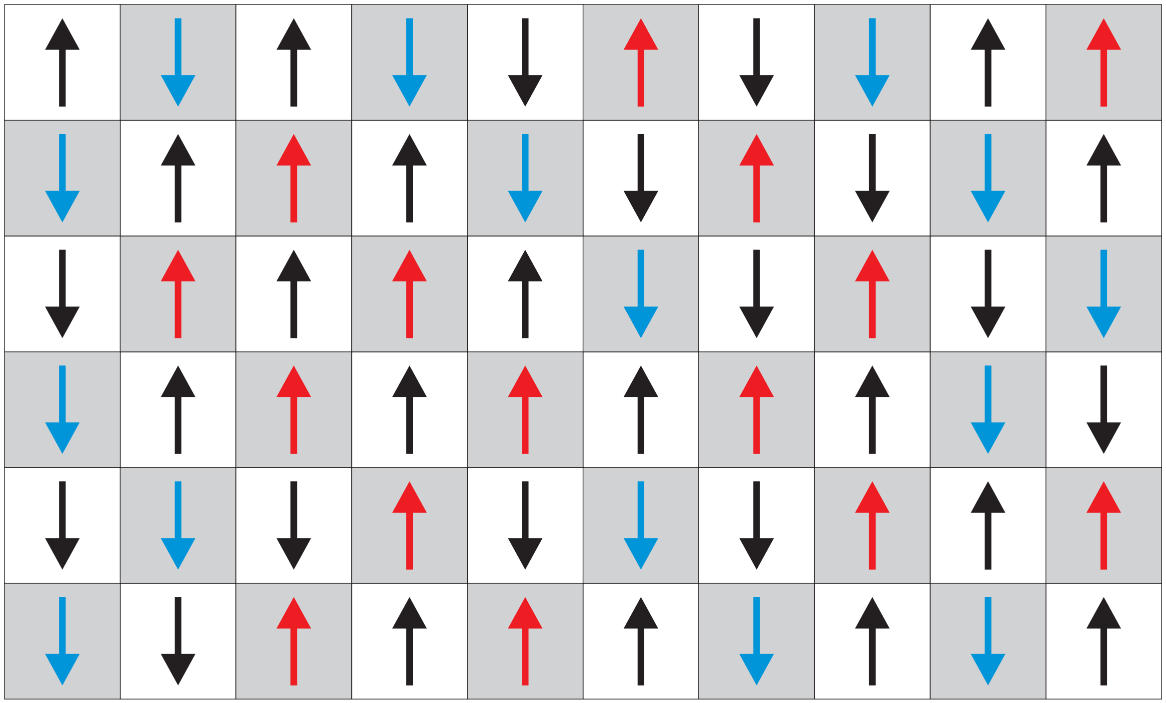
The twist that makes the Q2R rule reversible: Color the space as a checker-board. On even time steps only update the spins of the white cells and on odd time steps update the spins of the black cells.



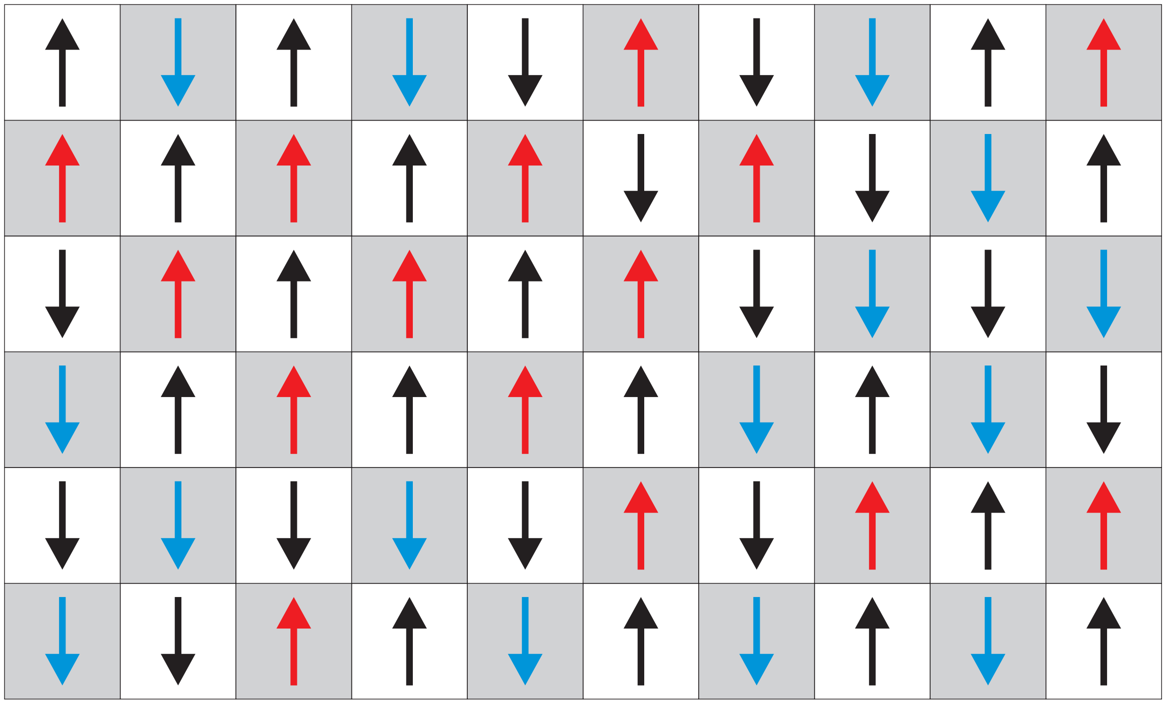
The twist that makes the Q2R rule reversible: Color the space as a checker-board. On even time steps only update the spins of the white cells and on odd time steps update the spins of the black cells.



The twist that makes the Q2R rule reversible: Color the space as a checker-board. On even time steps only update the spins of the white cells and on odd time steps update the spins of the black cells.

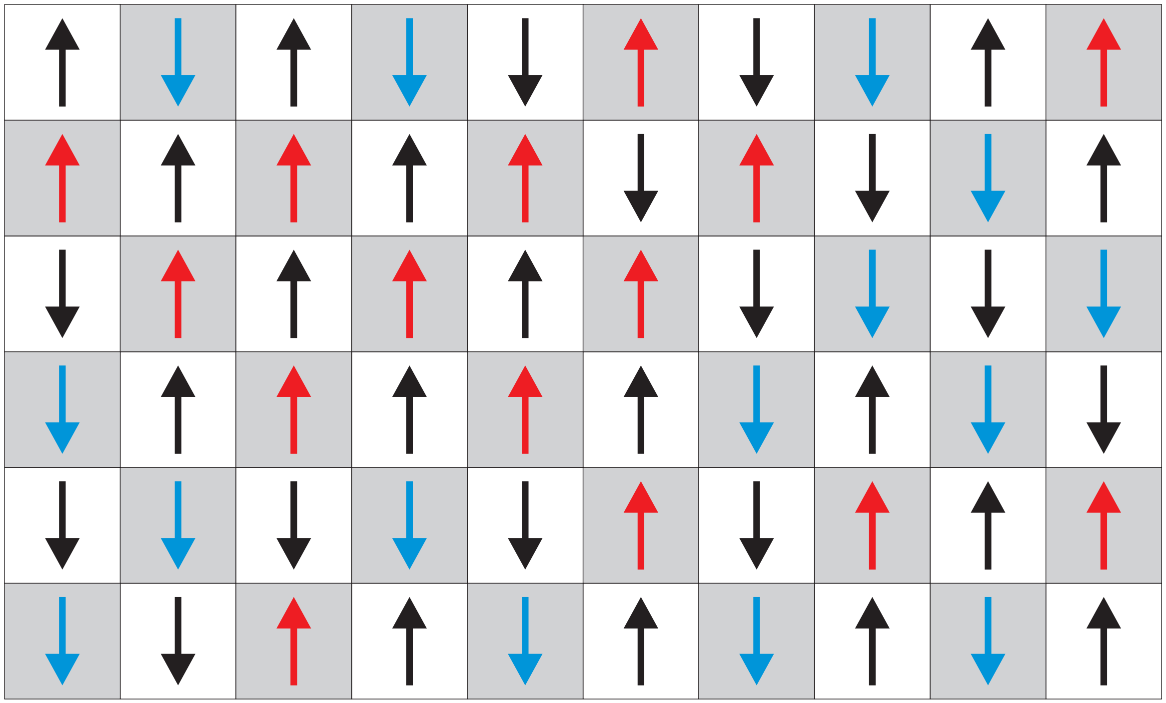


The twist that makes the Q2R rule reversible: Color the space as a checker-board. On even time steps only update the spins of the white cells and on odd time steps update the spins of the black cells.

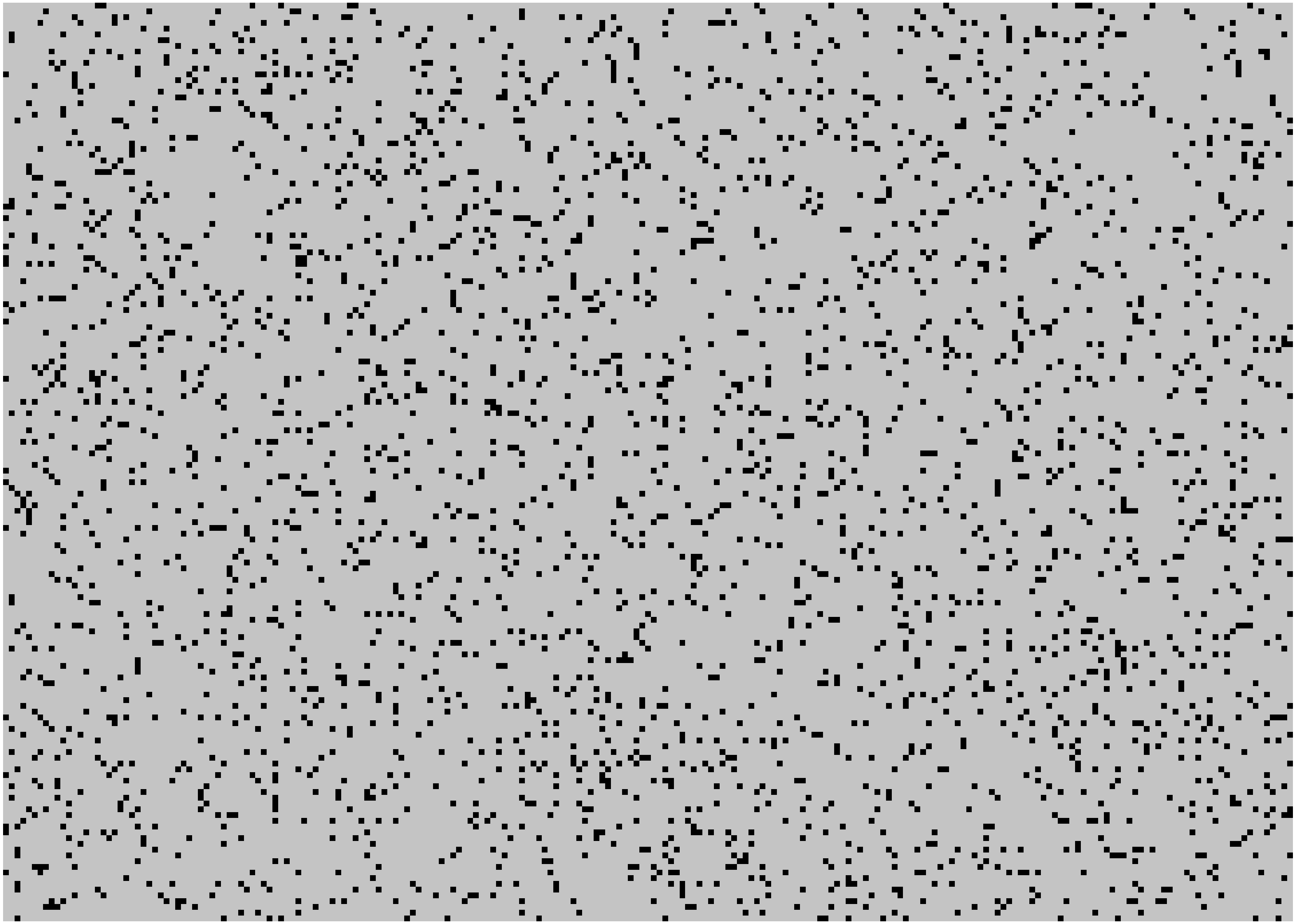


Q2R is **reversible**: The same rule (applied again on squares of the same color) reconstructs the previous generation.

Q2R rule also exhibits a local **conservation law**: The number of neighbors with opposite spins remains constant over time.

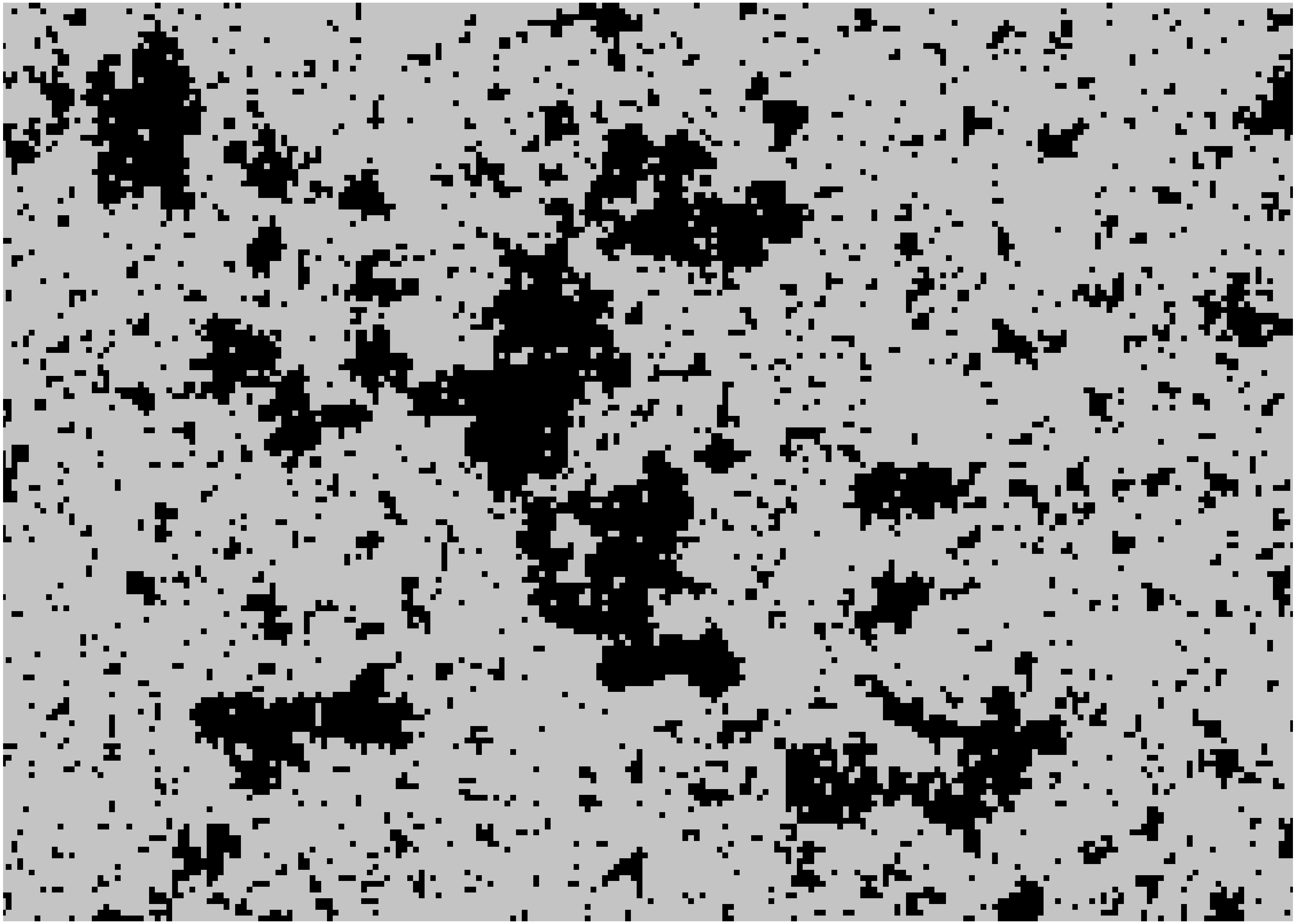


Evolution of Q2R from an uneven random distribution of spins:



Initial random configuration with 8% spins up.

Evolution of Q2R from an uneven random distribution of spins:



After approx. one million steps. Notice the clustering.

From the Curtis-Hedlund-Lyndom -theorem we get

Corollary: A cellular automaton G is reversible if and only if it is bijective.

From the Curtis-Hedlund-Lyndom -theorem we get

Corollary: A cellular automaton G is reversible if and only if it is bijective.

Proof: If G is a reversible CA function then G is by definition bijective.

Conversely, suppose that G is a bijective CA function. Then G has an inverse function G^{-1} that clearly commutes with the shifts. The inverse function G^{-1} is also continuous because the space $S^{\mathbb{Z}^d}$ is compact. It now follows from the Curtis-Hedlund-Lyndon theorem that G^{-1} is a cellular automaton. $\square$

From the Curtis-Hedlund-Lyndom -theorem we get

Corollary: A cellular automaton G is reversible if and only if it is bijective.

The point of the corollary is that in bijective CA each cell can determine its previous state by looking at the current states in some bounded neighborhood around them.

Wang tiles and decidability questions

Suppose we are given a CA (in terms of its local update rule) and want to know if it is reversible or surjective ? Is there an algorithm to decide this ? Or is there an algorithm to determine if the dynamics of a given CA is trivial in the sense that after a while all activity has died ?

It turns out that many such algorithmic problems are **undecidable**. In some cases there is an algorithm for one-dimensional CA while the two-dimensional case is undecidable.

A useful tool to obtain undecidability results is the concept of **Wang tiles** and the undecidable **tiling problem**.

A **Wang tile** is a unit square tile with colored edges. A tile set T is a finite collection of such tiles. A valid tiling is an assignment

$$\mathbb{Z}^2 \longrightarrow T$$

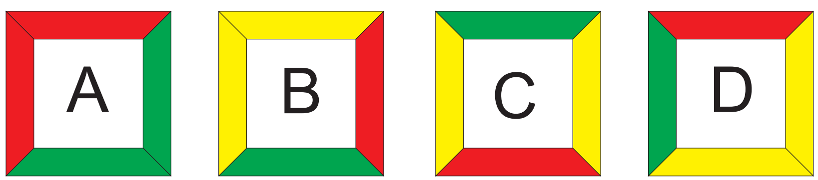
of tiles on infinite square lattice so that the abutting edges of adjacent tiles have the same color.

A **Wang tile** is a unit square tile with colored edges. A tile set T is a finite collection of such tiles. A valid tiling is an assignment

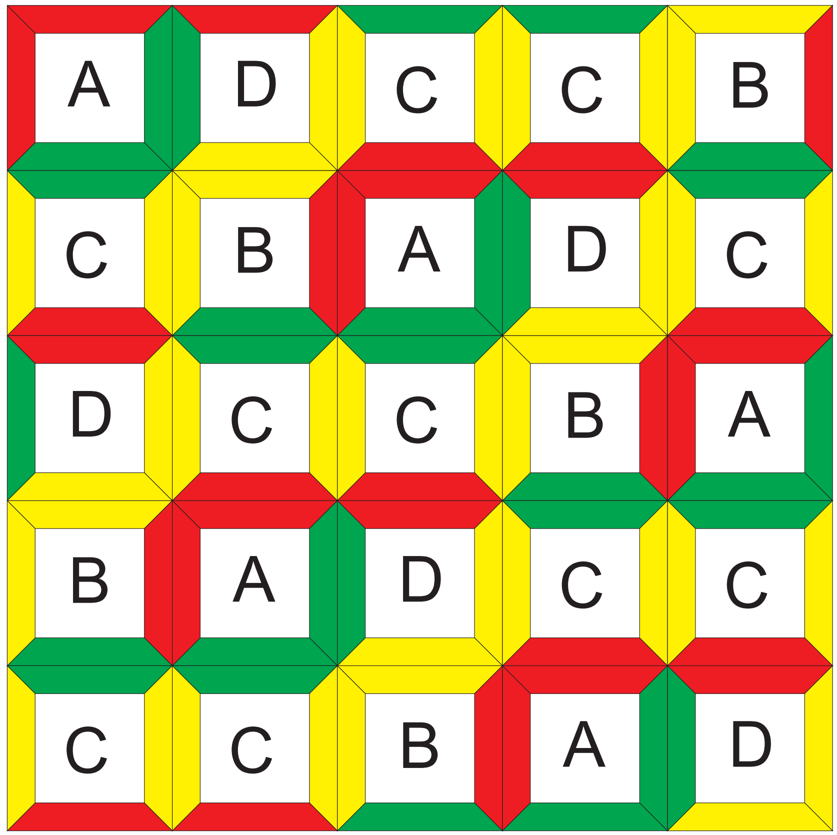
$$\mathbb{Z}^2 \longrightarrow T$$

of tiles on infinite square lattice so that the abutting edges of adjacent tiles have the same color.

For example, consider Wang tiles



With copies of the given four tiles we can properly tile a 5×5 square. . .



. . . and since the colors on the borders match this square can be repeated to form a valid periodic tiling of the plane.

Configuration $c \in T^{\mathbb{Z}^2}$ is **valid inside** $M \subseteq \mathbb{Z}^2$ if the colors match between any two neighboring cells, both of which are inside region M.

If here $M = \mathbb{Z}^2$, the configuration is a **valid tiling** of the plane.

The set of valid tilings over T is a translation invariant, compact subset of the configuration space $T^{\mathbb{Z}^2}$, i.e., it is a subshift.

The set of valid tilings over T is a translation invariant, compact subset of the configuration space $T^{\mathbb{Z}^2}$, i.e., it is a subshift.

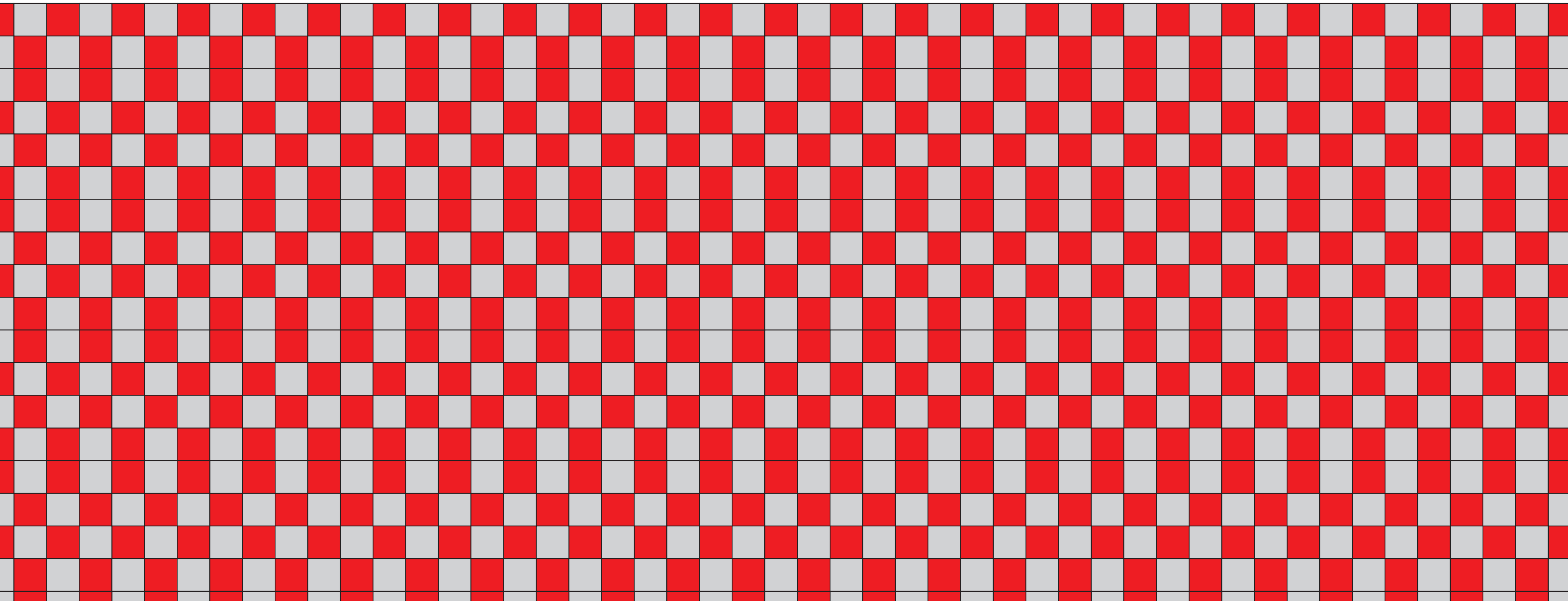
More precisely, valid tilings form a **subshift of finite type** because they are defined via a finite collection of patterns that are not allowed in any valid tiling.

Moreover, any two-dimensional subshift of finite type is conjugate to the set of valid tilings under a suitable Wang tile set.

A configuration $c \in T^{\mathbb{Z}^2}$ **(doubly) periodic** if there are two linearly independent translations τ_1 and τ_2 that keep c invariant:

$$\tau_1(c) = \tau_2(c) = c.$$

Then c is also invariant under some horizontal and vertical translations.



A configuration $c \in T^{\mathbb{Z}^2}$ **(doubly) periodic** if there are two linearly independent translations τ_1 and τ_2 that keep c invariant:

$$\tau_1(c) = \tau_2(c) = c.$$

Then c is also invariant under some horizontal and vertical translations.

Proposition: If a tile set admits a tiling that is invariant under some non-zero translation then it admits a valid doubly periodic tiling.

More generally, a d-dimensional configuration $c \in S^{\mathbb{Z}^d}$ is **(d-ways) periodic** if it is invariant under d linearly independent translations.

The **tiling problem** of Wang tiles is the decision problem to determine if a given finite set of Wang tiles admits a valid tiling of the plane.

Theorem (R.Berger 1966): The tiling problem of Wang tiles is undecidable.

Observations:

(1) If T admits valid tilings inside squares of arbitrary size then it admits a valid tiling of the whole plane.

Observations:

(1) If T admits valid tilings inside squares of arbitrary size then it admits a valid tiling of the whole plane.

Follows from compactness: Let $t_1, t_2, \ldots$ be a sequence of configurations $t_n \in T^{\mathbb{Z}^2}$ where t_n is a valid tiling inside the $(2n+1) \times (2n+1)$ square centered at the origin.

By compactness, the sequence has a converging subsequence. The limit $t \in T^{\mathbb{Z}^2}$ of the subsequence is clearly a valid tiling of the plane.

Observations:

(1) If T admits valid tilings inside squares of arbitrary size then it admits a valid tiling of the whole plane.

(2) There is a **semi-algorithm** to recursively enumerate tile sets that do not admit valid tilings of the plane.

Observations:

(1) If T admits valid tilings inside squares of arbitrary size then it admits a valid tiling of the whole plane.

(2) There is a **semi-algorithm** to recursively enumerate tile sets that do not admit valid tilings of the plane.

Follows from (1): Just try tiling larger and larger squares until (if ever) a square is found that can not be tiled.

Observations:

(1) If T admits valid tilings inside squares of arbitrary size then it admits a valid tiling of the whole plane.

(2) There is a **semi-algorithm** to recursively enumerate tile sets that do not admit valid tilings of the plane.

(3) There is a **semi-algorithm** to recursively enumerate tile sets that admit a valid periodic tiling.

Observations:

(1) If T admits valid tilings inside squares of arbitrary size then it admits a valid tiling of the whole plane.

(2) There is a **semi-algorithm** to recursively enumerate tile sets that do not admit valid tilings of the plane.

(3) There is a **semi-algorithm** to recursively enumerate tile sets that admit a valid periodic tiling.

Reason: Just try tiling rectangles until (if ever) a valid tiling is found where colors on the top and the bottom match, and left and the right sides match as well.

Observations:

(1) If T admits valid tilings inside squares of arbitrary size then it admits a valid tiling of the whole plane.

(2) There is a **semi-algorithm** to recursively enumerate tile sets that do not admit valid tilings of the plane.

(3) There is a **semi-algorithm** to recursively enumerate tile sets that admit a valid periodic tiling.

(4) There exist **aperiodic** sets of Wang tiles. These

- admit valid tilings of the plane, but
- do not admit any periodic tiling

Observations:

(1) If T admits valid tilings inside squares of arbitrary size then it admits a valid tiling of the whole plane.

(2) There is a **semi-algorithm** to recursively enumerate tile sets that do not admit valid tilings of the plane.

(3) There is a **semi-algorithm** to recursively enumerate tile sets that admit a valid periodic tiling.

(4) There exist **aperiodic** sets of Wang tiles. These

- admit valid tilings of the plane, but
- do not admit any periodic tiling

Follows from (2), (3) and undecidability of the tiling problem.

The tiling problem can be reduced to various decision problems concerning (two-dimensional) cellular automata, so that the undecidability of these problems then follows from Berger's result.

This is not so surprising since Wang tilings are "static" versions of "dynamic" cellular automata.

Example: Let us prove that it is undecidable whether a given two-dimensional CA G has any fixed point configurations, that is, configurations c such that $G(c) = c$.

Proof: Reduction from the tiling problem. For any given Wang tile set T (with at least two tiles) we effectively construct a two-dimensional CA with state set T, the von Neumann -neighborhood and a local update rule that keeps a tile unchanged if and only if its colors match with the neighboring tiles.

Trivially, $G(c) = c$ if and only if c is a valid tiling. □

Example: Let us prove that it is undecidable whether a given two-dimensional CA G has any fixed point configurations, that is, configurations c such that $G(c) = c$.

Proof: Reduction from the tiling problem. For any given Wang tile set T (with at least two tiles) we effectively construct a two-dimensional CA with state set T, the von Neumann -neighborhood and a local update rule that keeps a tile unchanged if and only if its colors match with the neighboring tiles.

Trivially, $G(c) = c$ if and only if c is a valid tiling. $\square$

Note: For one-dimensional CA it is decidable whether fixed points exist. Fixed points form a subshift of finite type that can be effectively constructed.

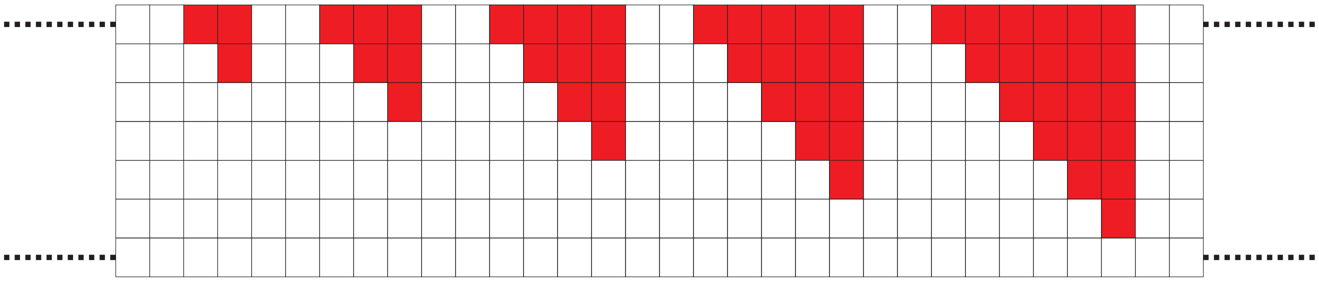
More interesting reduction: A CA is called **nilpotent** if all configurations eventually evolve into the quiescent configuration.

Observation: In a nilpotent CA all configurations must become quiescent within a bounded time, that is, there is number n such that $G^n(c)$ is quiescent, for all $c \in S^{\mathbb{Z}^d}$.

More interesting reduction: A CA is called **nilpotent** if all configurations eventually evolve into the quiescent configuration.

Observation: In a nilpotent CA all configurations must become quiescent within a bounded time, that is, there is number n such that $G^n(c)$ is quiescent, for all $c \in S^{\mathbb{Z}^d}$.

Proof: Suppose contrary: for every n there is a configuration c_n such that $G^n(c_n)$ is not quiescent. Then c_n contains a finite pattern p_n that evolves in n steps into some non-quiescent state. A configuration c that contains a copy of every p_n never becomes quiescent, contradicting nilpotency.



Theorem (Culik, Pachl, Yu, 1989): It is undecidable whether a given two-dimensional CA is nilpotent.

Theorem (Culik, Pachl, Yu, 1989): It is undecidable whether a given two-dimensional CA is nilpotent.

Proof: For any given set T of Wang tiles the goal is to construct a two-dimensional CA that is nilpotent if and only if T does not admit a tiling.

For tile set T we make the following CA:

- State set is $S = T \cup \{q\}$ where q is a new symbol $q \notin T$,

For tile set T we make the following CA:

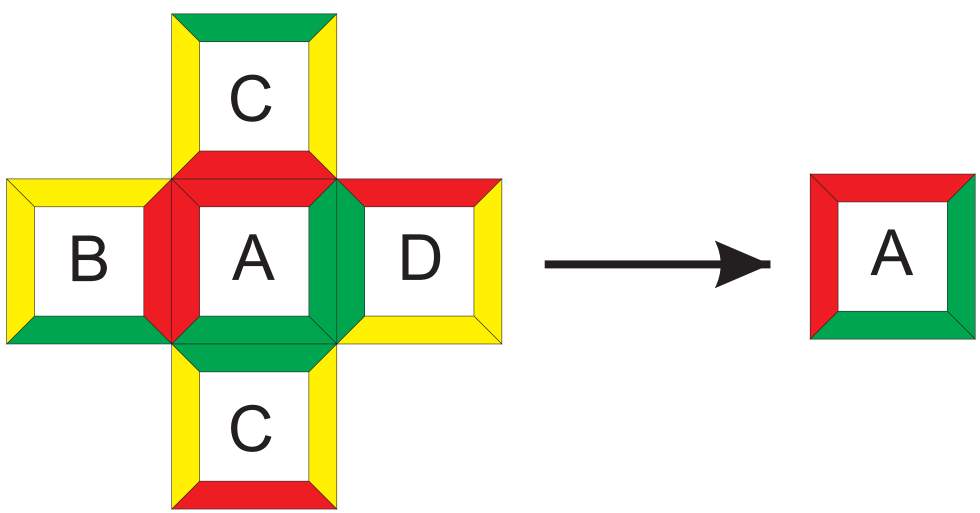
- State set is $S = T \cup \{q\}$ where q is a new symbol $q \notin T$,
- Von Neumann neighborhood,

For tile set T we make the following CA:

- State set is $S = T \cup \{q\}$ where q is a new symbol $q \notin T$,
- Von Neumann neighborhood,
- The local rule keeps state unchanged if all states in the neighborhood are tiles and the tiling constraint is satisfied. In all other cases the new state is q.

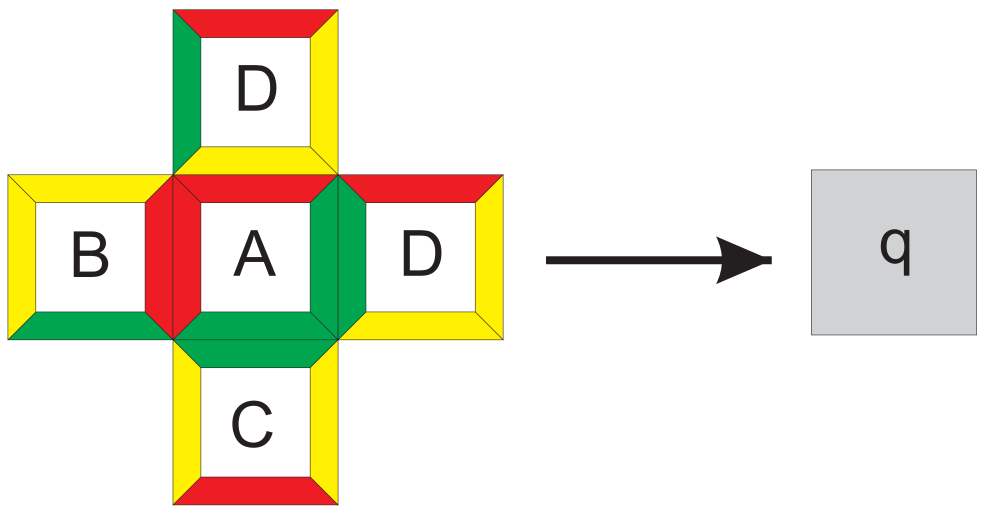
For tile set T we make the following CA:

- State set is $S = T \cup \{q\}$ where q is a new symbol $q \notin T$,
- Von Neumann neighborhood,
- The local rule keeps state unchanged if all states in the neighborhood are tiles and the tiling constraint is satisfied. In all other cases the new state is q.



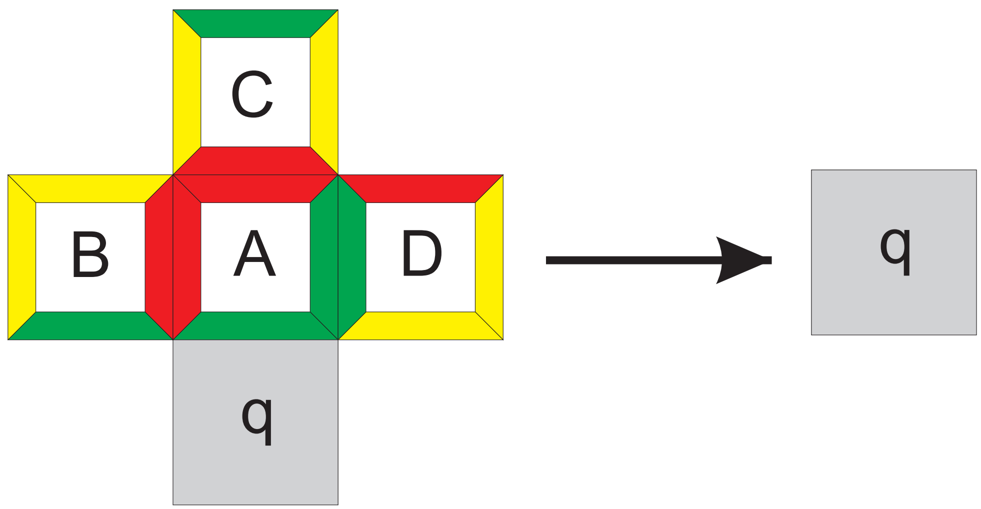
For tile set T we make the following CA:

- State set is $S = T \cup \{q\}$ where q is a new symbol $q \notin T$,
- Von Neumann neighborhood,
- The local rule keeps state unchanged if all states in the neighborhood are tiles and the tiling constraint is satisfied. In all other cases the new state is q.



For tile set T we make the following CA:

- State set is $S = T \cup \{q\}$ where q is a new symbol $q \notin T$,
- Von Neumann neighborhood,
- The local rule keeps state unchanged if all states in the neighborhood are tiles and the tiling constraint is satisfied. In all other cases the new state is q.



For tile set T we make the following CA:

- State set is $S = T \cup \{q\}$ where q is a new symbol $q \notin T$,
- Von Neumann neighborhood,
- The local rule keeps state unchanged if all states in the neighborhood are tiles and the tiling constraint is satisfied. In all other cases the new state is q.

For tile set T we make the following CA:

- State set is $S = T \cup \{q\}$ where q is a new symbol $q \notin T$,
- Von Neumann neighborhood,
- The local rule keeps state unchanged if all states in the neighborhood are tiles and the tiling constraint is satisfied. In all other cases the new state is q.

$\Longrightarrow$ If T admits a tiling c then c is a non-quiescent fixed point of the CA. So the CA is not nilpotent.

For tile set T we make the following CA:

- State set is $S = T \cup \{q\}$ where q is a new symbol $q \notin T$,
- Von Neumann neighborhood,
- The local rule keeps state unchanged if all states in the neighborhood are tiles and the tiling constraint is satisfied. In all other cases the new state is q.

$\Longrightarrow$ If T admits a tiling c then c is a non-quiescent fixed point of the CA. So the CA is not nilpotent.

$\Longleftarrow$ If T does not admit a valid tiling then every $n \times n$ square contains a tiling error, for some n. State q propagates, so in at most $2n$ steps all cells are in state q. The CA is nilpotent. $\square$

If we do the previous construction for an aperiodic tile set T we obtain a two-dimensional CA in which every periodic configuration becomes eventually quiescent, but there are some non-periodic fixed points.

If we do the previous construction for an aperiodic tile set T we obtain a two-dimensional CA in which every periodic configuration becomes eventually quiescent, but there are some non-periodic fixed points.

Another interesting observation is that while in nilpotent CA all configurations become quiescent within bounded time, that transient time can be very long: one cannot compute any upper bound on it as otherwise nilpotency could be effectively checked.

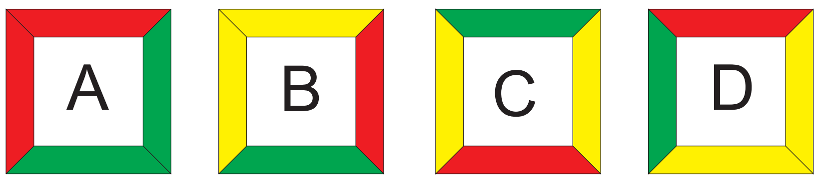
NW-deterministic tiles

While tilings relate naturally to two-dimensional CA, one can strengthen Berger's result so that the nilpotency can be proved undecidable for one-dimensional CA as well.

The basic idea is to view space-time diagrams of one-dimensional CA as tilings: they are two-dimensional subshifts of finite type.

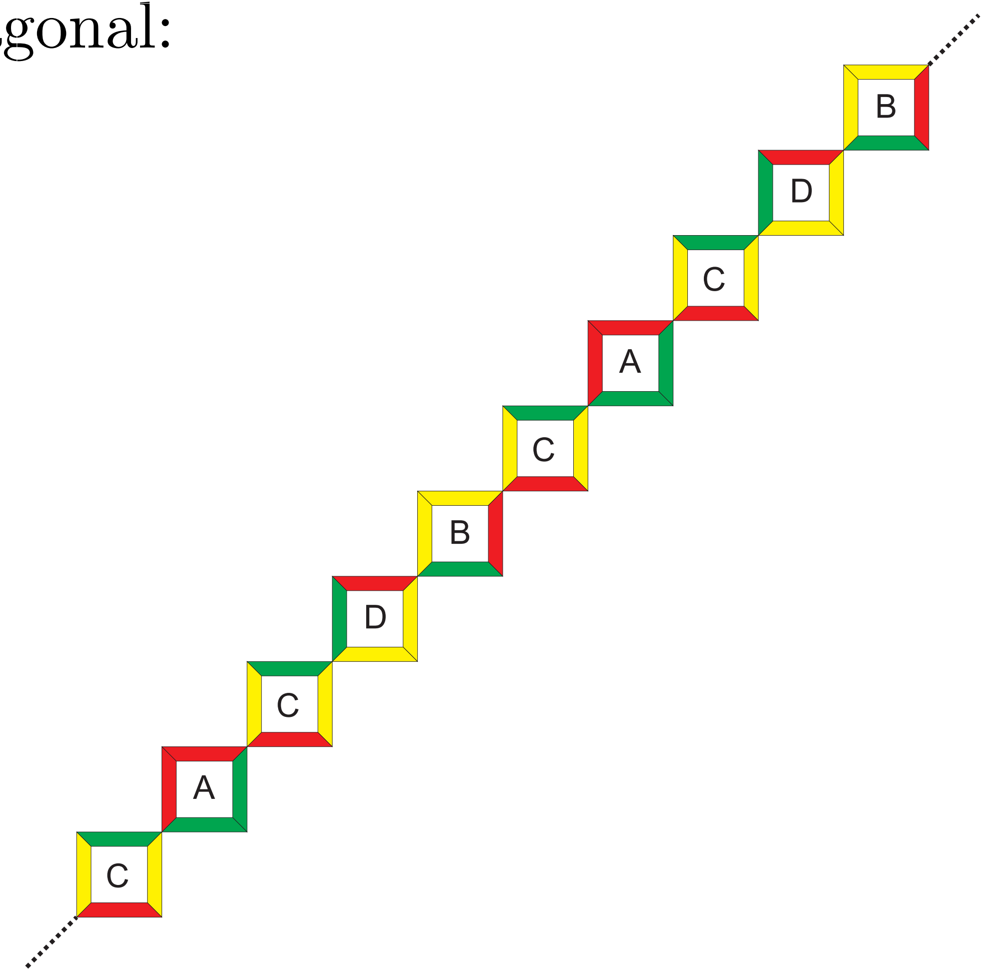
Tile set T is **NW-deterministic** if no two tiles have identical colors on their top edges and on their left edges. In a valid tiling the left and the top neighbor of a tile uniquely determine the tile.

For example, our sample tile set

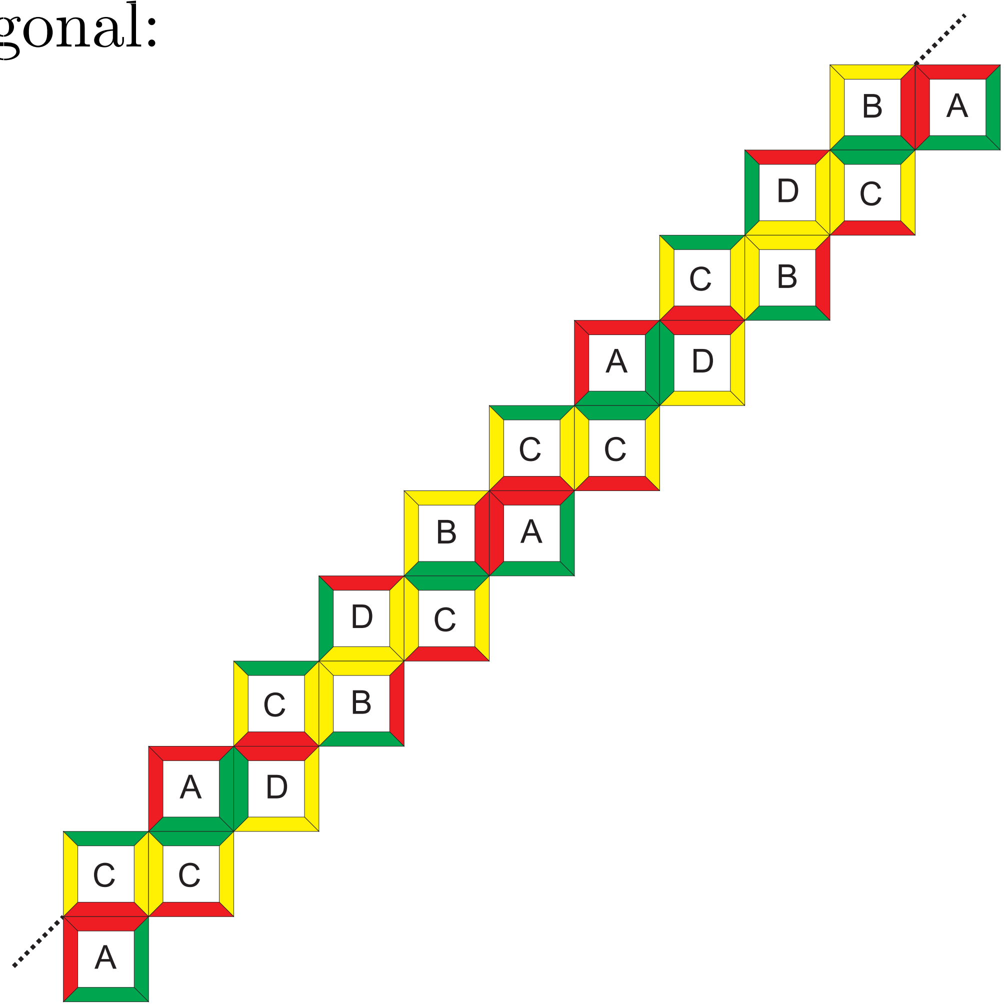


is NW-deterministic.

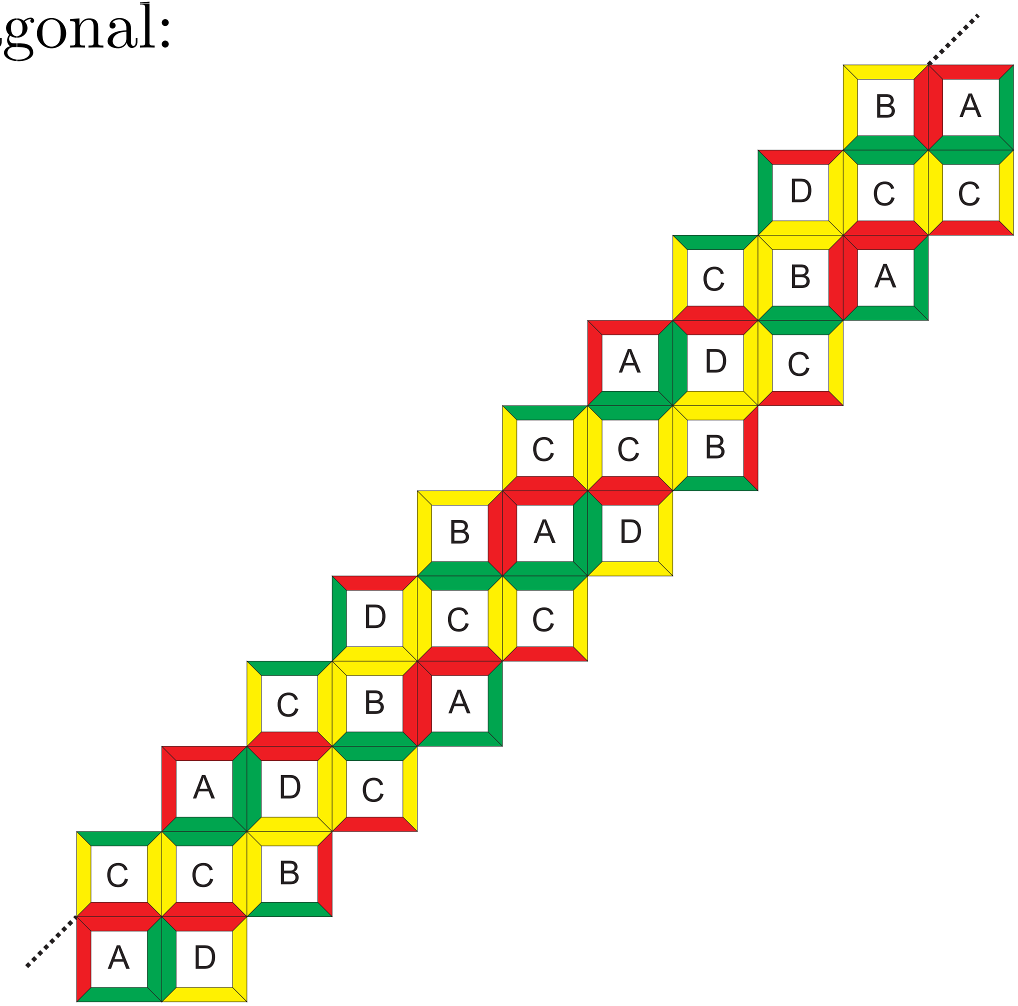
In any valid tiling by NW-deterministic tiles, NE-to-SW diagonals uniquely determine the next diagonal below them. The tiles of the next diagonal are determined locally from the previous diagonal:



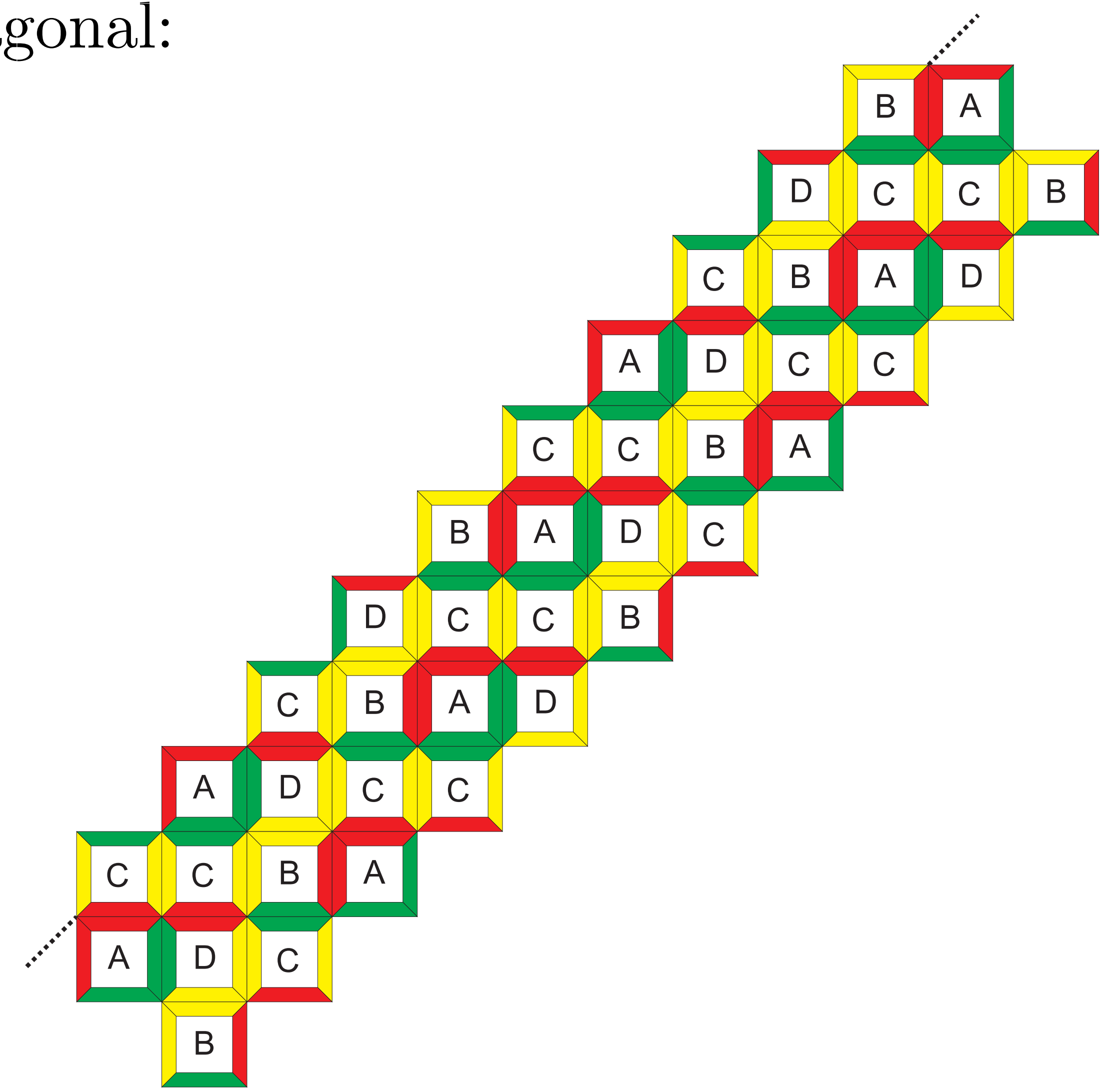
In any valid tiling by NW-deterministic tiles, NE-to-SW diagonals uniquely determine the next diagonal below them. The tiles of the next diagonal are determined locally from the previous diagonal:



In any valid tiling by NW-deterministic tiles, NE-to-SW diagonals uniquely determine the next diagonal below them. The tiles of the next diagonal are determined locally from the previous diagonal:



In any valid tiling by NW-deterministic tiles, NE-to-SW diagonals uniquely determine the next diagonal below them. The tiles of the next diagonal are determined locally from the previous diagonal:



If diagonals are interpreted as configurations of a one-dimensional CA, valid tilings represent space-time diagrams.

If diagonals are interpreted as configurations of a one-dimensional CA, valid tilings represent space-time diagrams.

More precisely, for any given NW-deterministic tile set T we construct a one-dimensional CA whose

- state set is $S = T \cup \{q\}$ where q is a new symbol $q \notin T$,

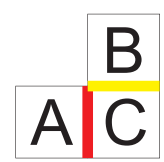
If diagonals are interpreted as configurations of a one-dimensional CA, valid tilings represent space-time diagrams.

More precisely, for any given NW-deterministic tile set T we construct a one-dimensional CA whose

- state set is $S = T \cup \{q\}$ where q is a new symbol $q \notin T$,
- neighborhood is $(0, 1)$,

If diagonals are interpreted as configurations of a one-dimensional CA, valid tilings represent space-time diagrams.

More precisely, for any given NW-deterministic tile set T we construct a one-dimensional CA whose

- state set is $S = T \cup \{q\}$ where q is a new symbol $q \notin T$,
- neighborhood is $(0, 1)$,
- local rule $f : S^2 \longrightarrow S$ is defined as follows:
  - $f(A, B) = C$ if the colors match in 

  - $f(A, B) = q$ if $A = q$ or $B = q$ or no matching tile C exists.

Claim: The CA is nilpotent if and only if T does not admit a tiling.

Claim: The CA is nilpotent if and only if T does not admit a tiling.

Proof:

$\Longrightarrow$ If T admits a tiling c then diagonals of c are configurations that never evolve into the quiescent configuration. So the CA is not nilpotent.

Claim: The CA is nilpotent if and only if T does not admit a tiling.

Proof:

$\Longrightarrow$ If T admits a tiling c then diagonals of c are configurations that never evolve into the quiescent configuration. So the CA is not nilpotent.

$\Longleftarrow$ If T does not admit a valid tiling then every $n \times n$ square contains a tiling error, for some n. Hence state q is created inside every segment of length n. Since q starts spreading once it has been created, the whole configuration becomes eventually quiescent.

Now we just need the following strengthening of Berger's theorem:

Theorem: The tiling problem is undecidable among NW-deterministic tile sets.

and we have

Theorem: It is undecidable whether a given one-dimensional CA (with spreading state q) is nilpotent. $\square$

NW-deterministic aperiodic tile sets exist.

If we do the previous construction using an aperiodic set then we have an interesting one-dimensional CA:

- all periodic configurations eventually die, but
- there are non-periodic configurations that never create a quiescent state in any cell.

NW-deterministic aperiodic tile sets exist.

If we do the previous construction using an aperiodic set then we have an interesting one-dimensional CA:

- all periodic configurations eventually die, but
- there are non-periodic configurations that never create a quiescent state in any cell.

As in the two-dimensional case, the transient time before a one-dimensional nilpotent CA dies can be very long: it cannot be bounded by any computable function.

The construction also provides the following result (due to Culik, Hurd, Kari):

Theorem: The topological entropy of a one-dimensional CA cannot be effectively computed, or even approximated.

The construction also provides the following result (due to Culik, Hurd, Kari):

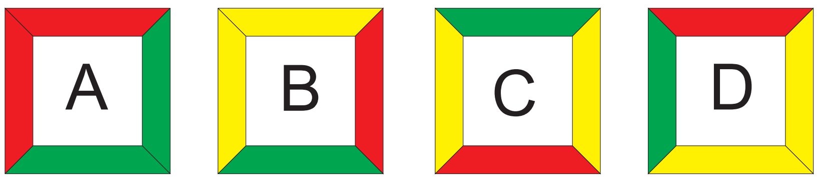
Theorem: The topological entropy of a one-dimensional CA cannot be effectively computed, or even approximated.

Proof: Add to the previous construction as a second layer a CA A with positive entropy $h(A)$. States of the new layer are killed whenever the tiling layer enters state q.

This CA is still nilpotent (and has zero entropy) if the tiles do not admit a tiling, but otherwise contain all orbits of A and hence have entropy at least as high as $h(A)$. $\square$

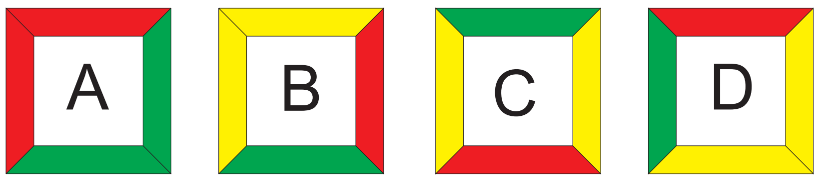
Analogously we can define NE-, SW- and SE-determinism of tile sets. A tile set is called **4-way** deterministic if it is deterministic in all four corners.

Our sample tile set is 4-way deterministic



Analogously we can define NE-, SW- and SE-determinism of tile sets. A tile set is called **4-way** deterministic if it is deterministic in all four corners.

Our sample tile set is 4-way deterministic



Ville Lukkarila has shown the following:

Theorem: The tiling problem is undecidable among 4-way deterministic tile sets.

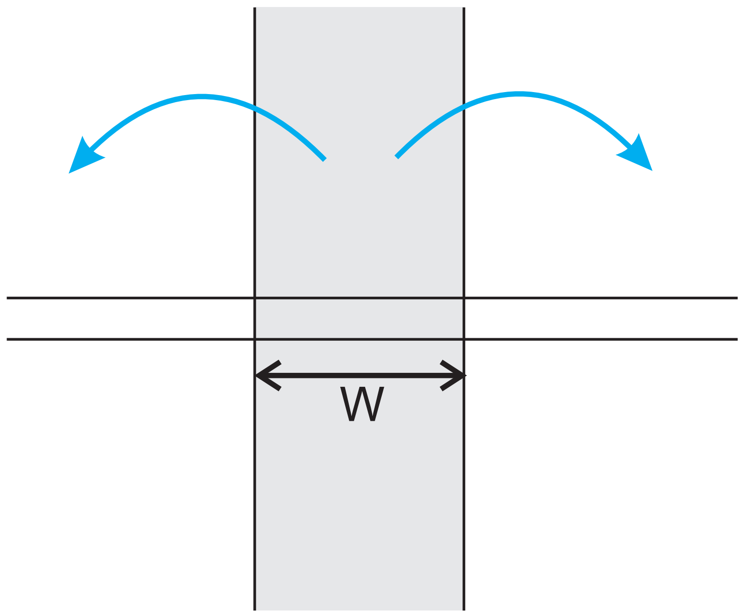
This result provides some undecidability results for dynamics of reversible one-dimensional CA.

Expansivity is a strong form of sensitivity to initial conditions.

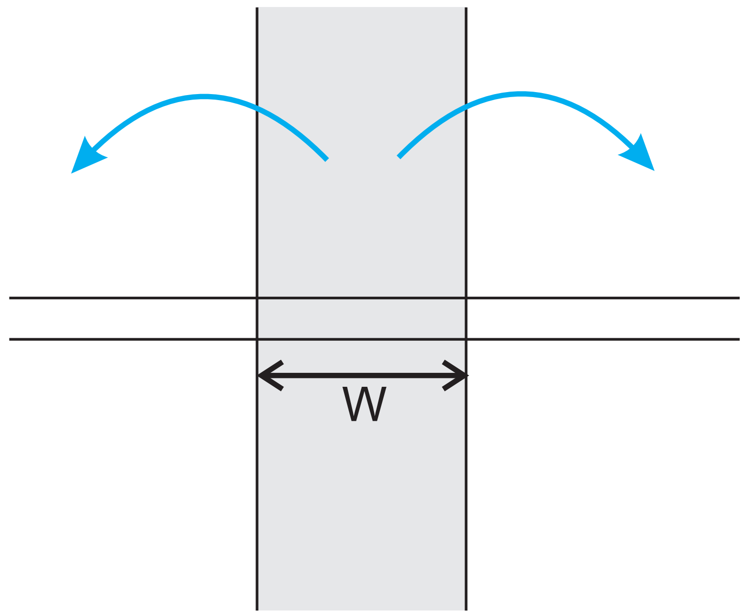
A one-dimensional reversible CA is expansive if there is a finite observation window $W \subset \mathbb{Z}^2$ such that

- knowing the states of the cells inside W at all times uniquely determines the configuration.

Expansivity: there is a vertical strip in space-time whose content uniquely identifies the entire space-time diagram:



Expansivity: there is a vertical strip in space-time whose content uniquely identifies the entire space-time diagram:

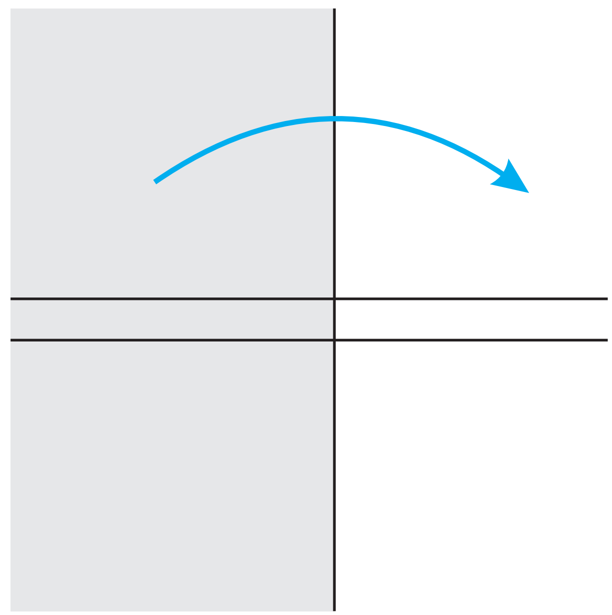


We would like to know which reversible CA are expansive.

Open problem: Is expansivity decidable ?

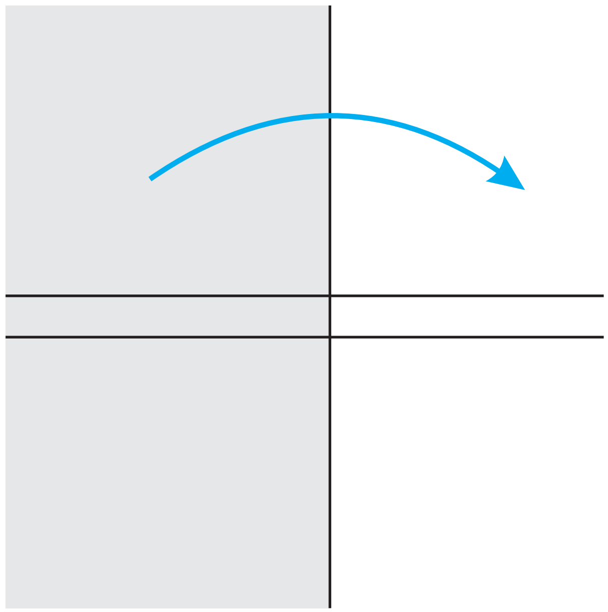
Let us call a one-dimensional reversible CA **left-expansive** if

- knowing the states of the cells $x < 0$ at all times uniquely determines the configuration.



Let us call a one-dimensional reversible CA **left-expansive** if

- knowing the states of the cells $x < 0$ at all times uniquely determines the configuration.

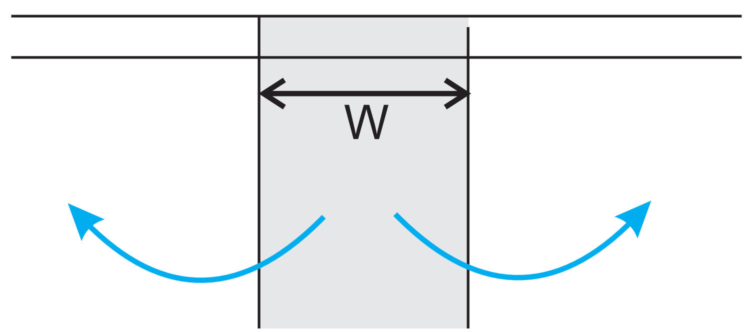


A reduction from the 4-way deterministic tiling problem proves

Theorem: It is undecidable if a given reversible 1D CA is left-expansive.

A (necessarily surjective) cellular automaton is **positively expansive** if there is a finite window $W \subset \mathbb{Z}^2$ such that

- knowing the states of the cells inside W at all **positive** times uniquely determines the initial configuration.

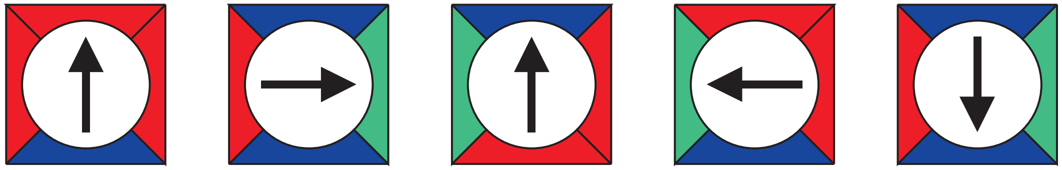


Open problem: Is positive expansivity decidable ?

SNAKES

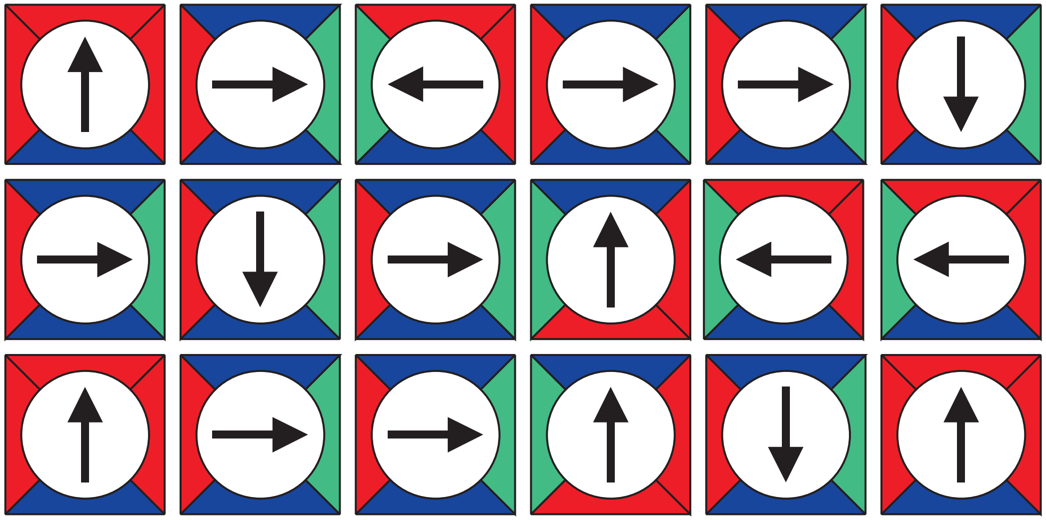
SNAKES is a tile set with some interesting (and useful) properties.

In addition to colored edges, these tiles also have an arrow printed on them. The arrow is horizontal or vertical and it points to one of the four neighbors of the tile:

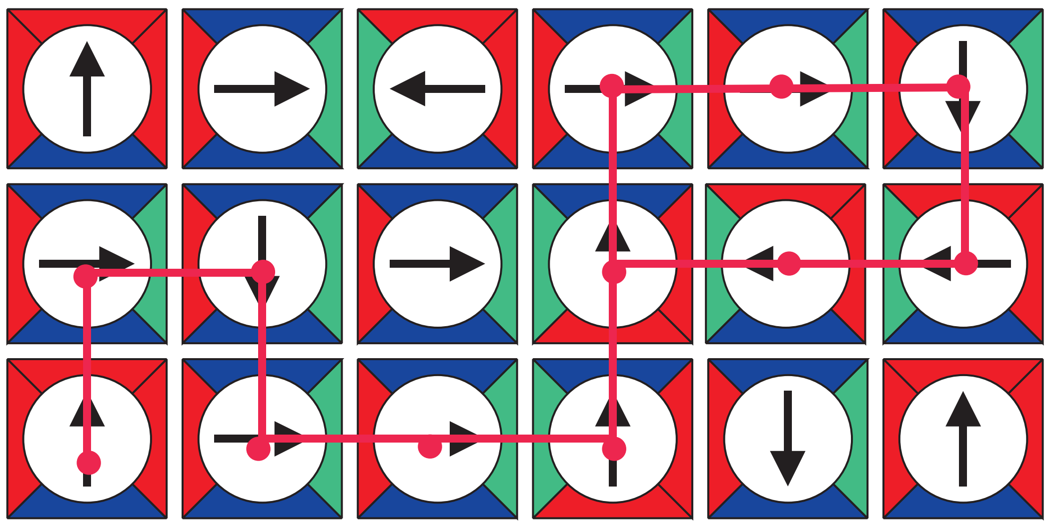


Such tiles with arrows are called **directed tiles**.

Given a configuration (valid tiling or not!) and a starting position, the arrows specify a path on the plane. Each position is followed by the neighboring position indicated by the arrow of the tile:

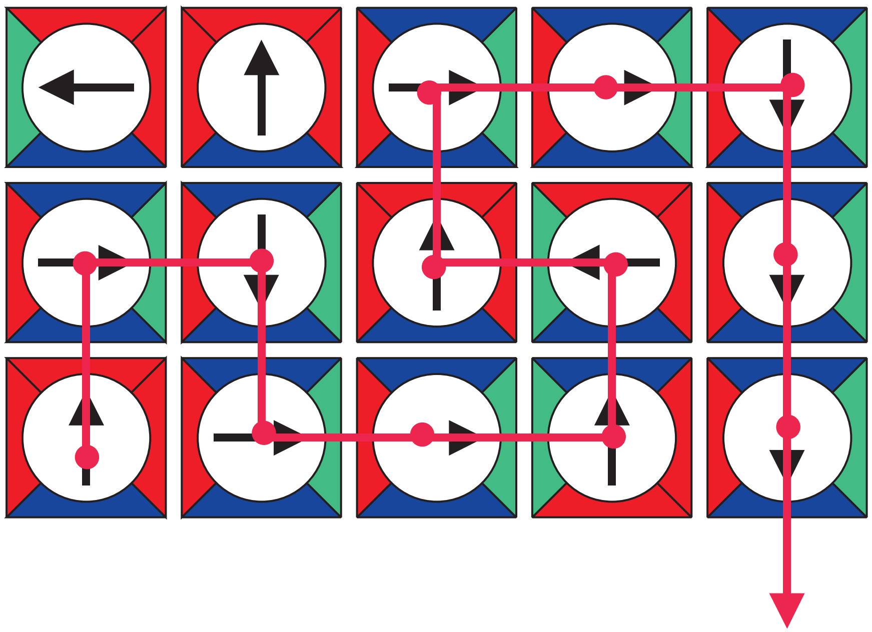


Given a configuration (valid tiling or not!) and a starting position, the arrows specify a path on the plane. Each position is followed by the neighboring position indicated by the arrow of the tile:



The path may enter a loop…

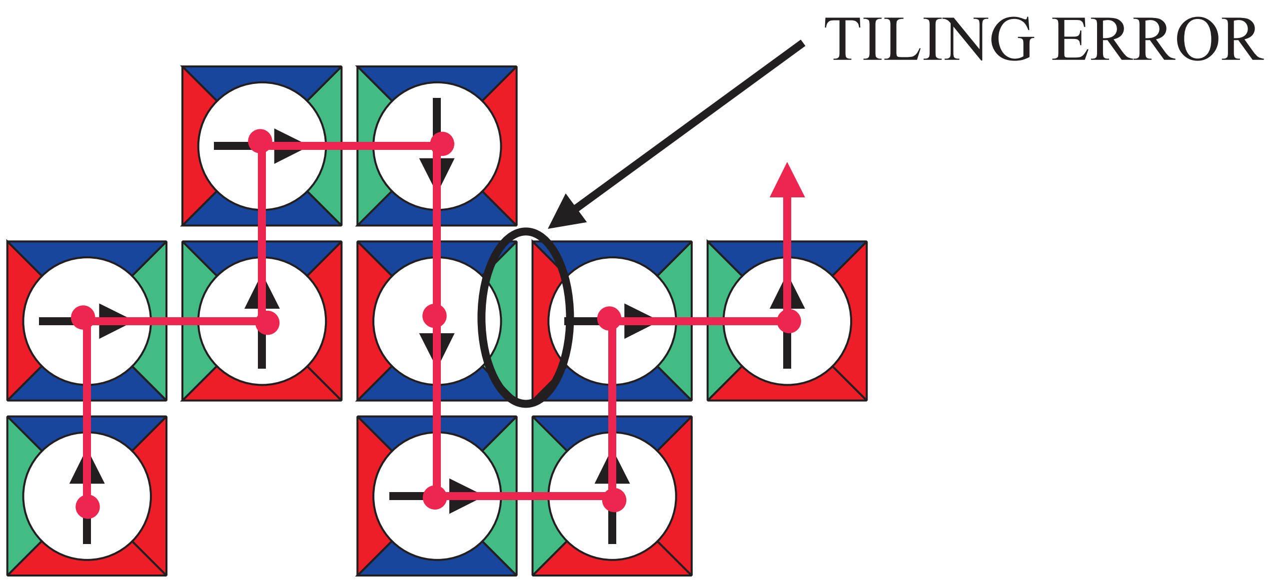
Given a configuration (valid tiling or not!) and a starting position, the arrows specify a path on the plane. Each position is followed by the neighboring position indicated by the arrow of the tile:



. . . or the path may be infinite and never return to a tile visited before.

The directed tile set SNAKES has the following property: On any configuration (valid tiling or not) and on any path that follows the arrows one of the following two things happens:

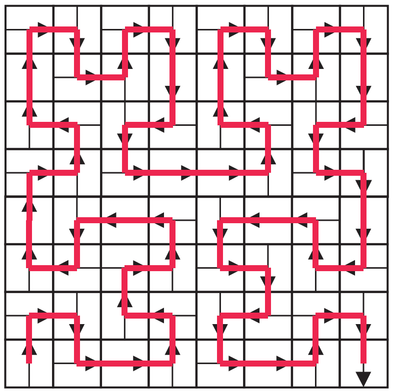
(1) Either there is a tiling error between two tiles both of which are on the path,



The directed tile set SNAKES has the following property: On any configuration (valid tiling or not) and on any path that follows the arrows one of the following two things happens:

(1) Either there is a tiling error between two tiles both of which are on the path,

(2) or the path is a plane-filling path, that is, for every positive integer n there exists an $n \times n$ square all of whose positions are visited by the path.



The directed tile set SNAKES has the following property: On any configuration (valid tiling or not) and on any path that follows the arrows one of the following two things happens:

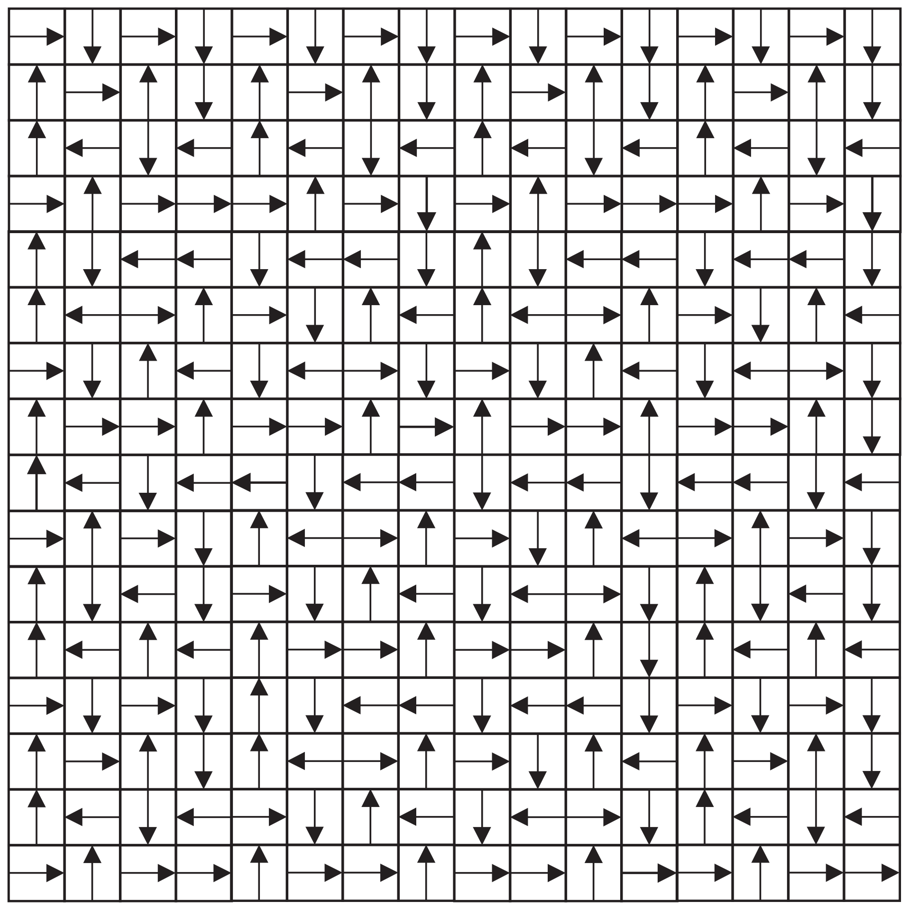
(1) Either there is a tiling error between two tiles both of which are on the path,

(2) or the path is a plane-filling path, that is, for every positive integer n there exists an $n \times n$ square all of whose positions are visited by the path.

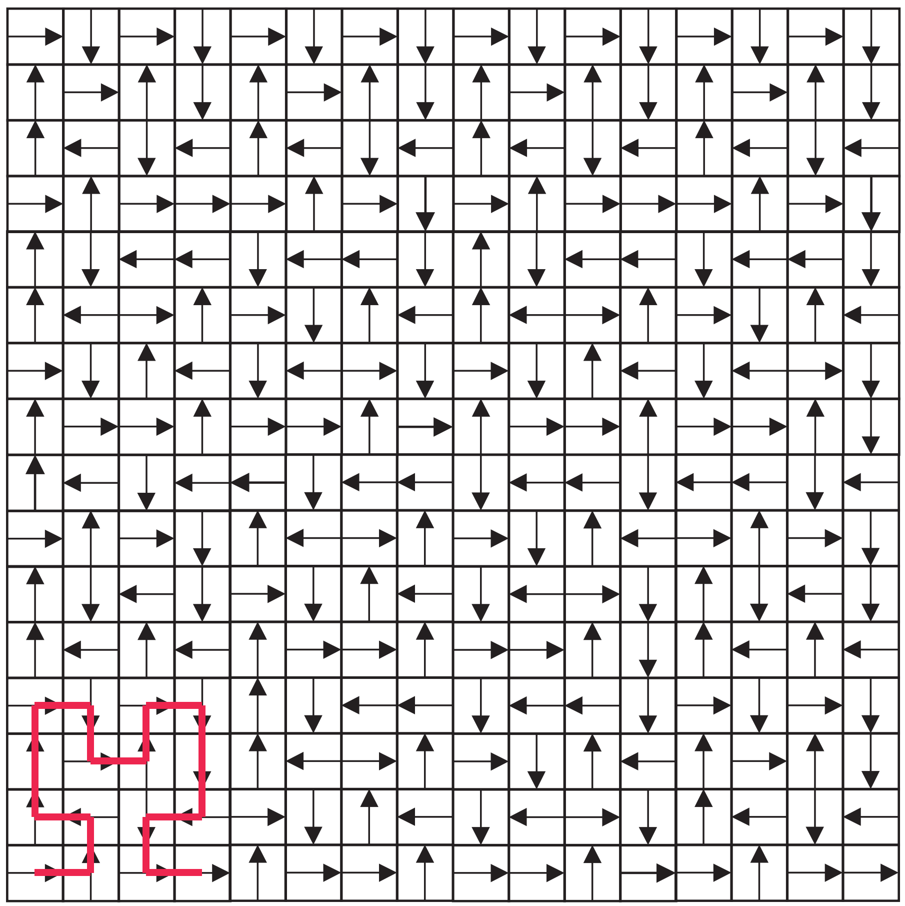
Note that the tiling may be invalid outside path P, yet the path is forced to snake through larger and larger squares.

SNAKES also has the property that it admits a valid tiling.

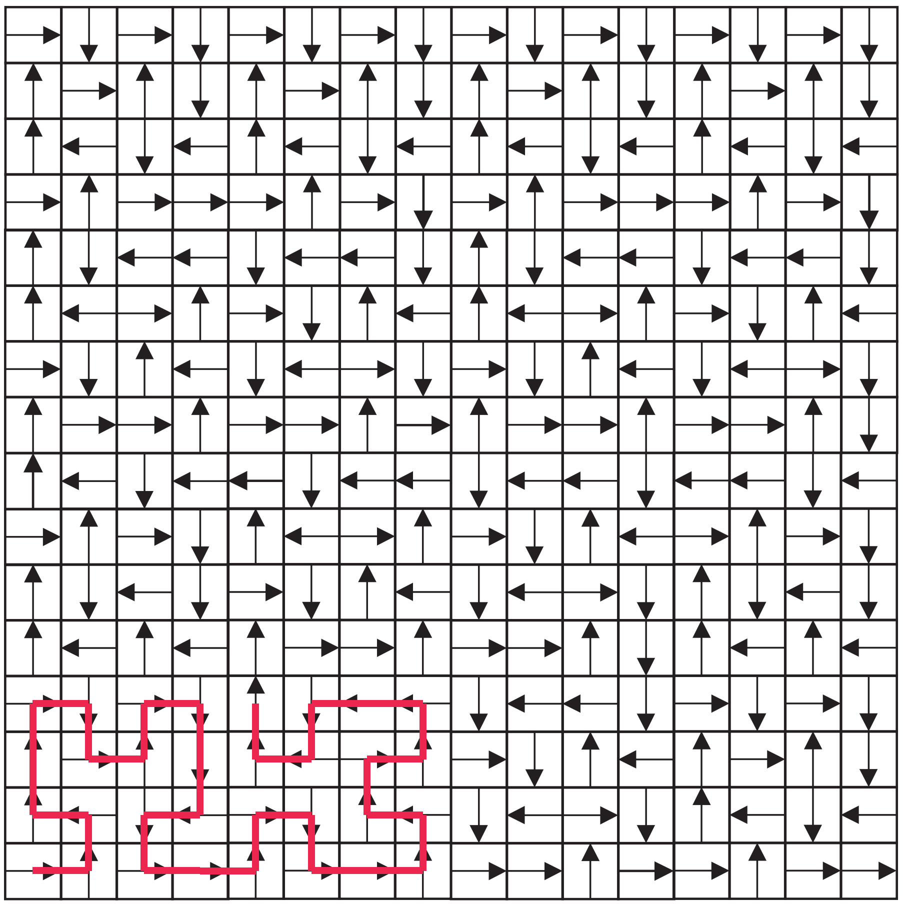
The construction of SNAKES is fairly complex and will be skipped. The paths that SNAKES forces when no tiling error is encountered have the shape of the well known plane-filling Hilbert-curve



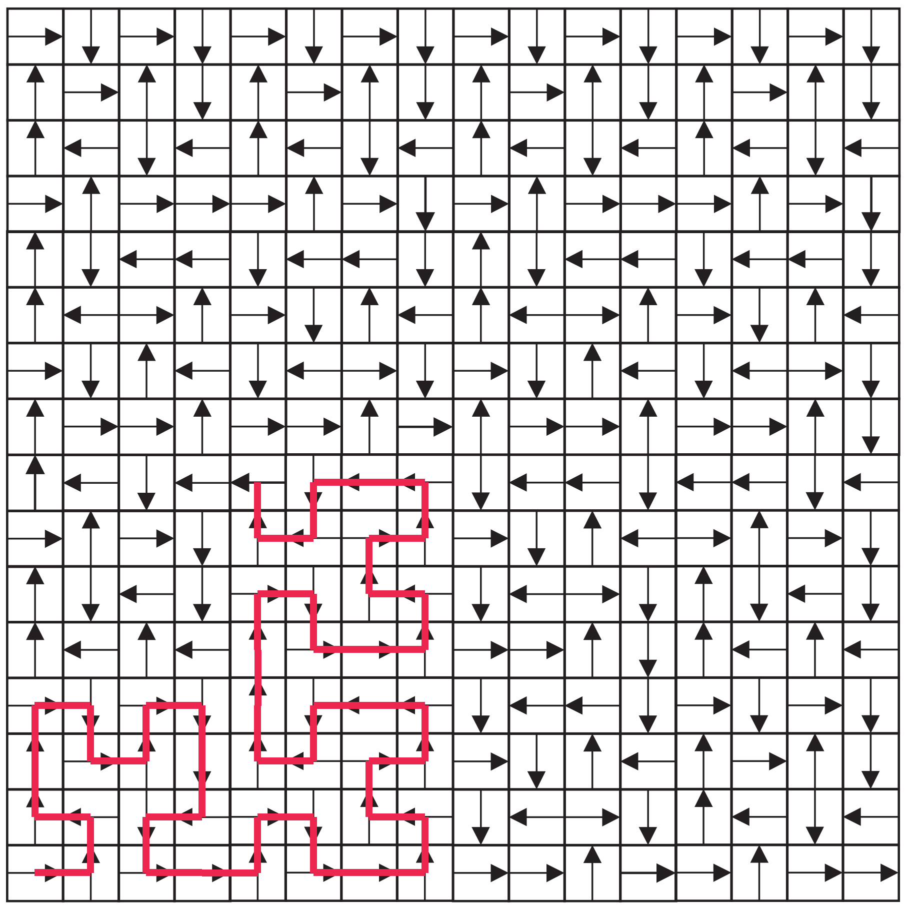
The construction of Snakes is fairly complex and will be skipped. The paths that Snakes forces when no tiling error is encountered have the shape of the well known plane-filling Hilbert-curve



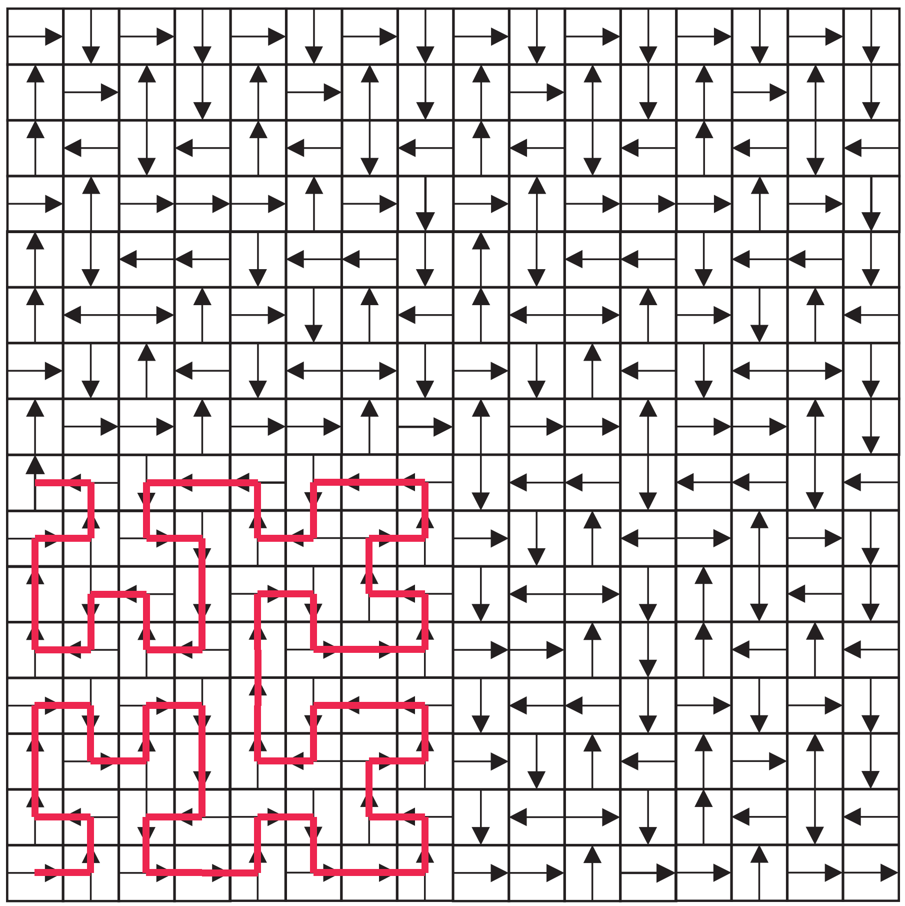
The construction of SNAKES is fairly complex and will be skipped. The paths that SNAKES forces when no tiling error is encountered have the shape of the well known plane-filling Hilbert-curve



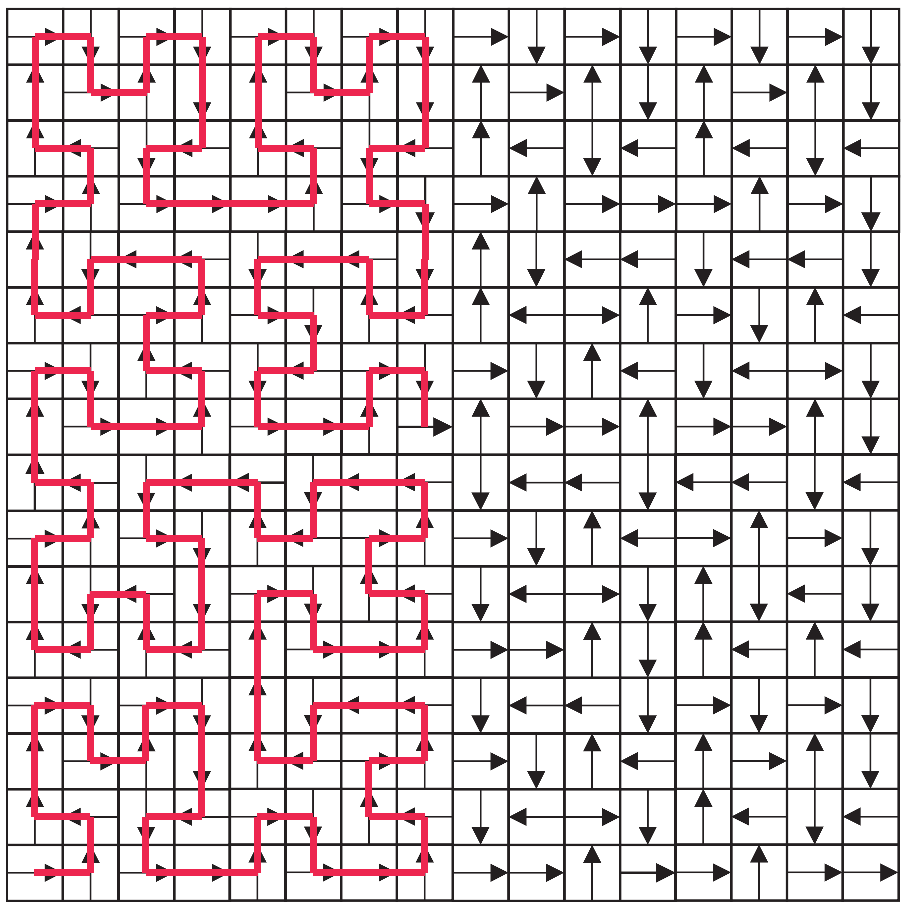
The construction of SNAKES is fairly complex and will be skipped. The paths that SNAKES forces when no tiling error is encountered have the shape of the well known plane-filling Hilbert-curve



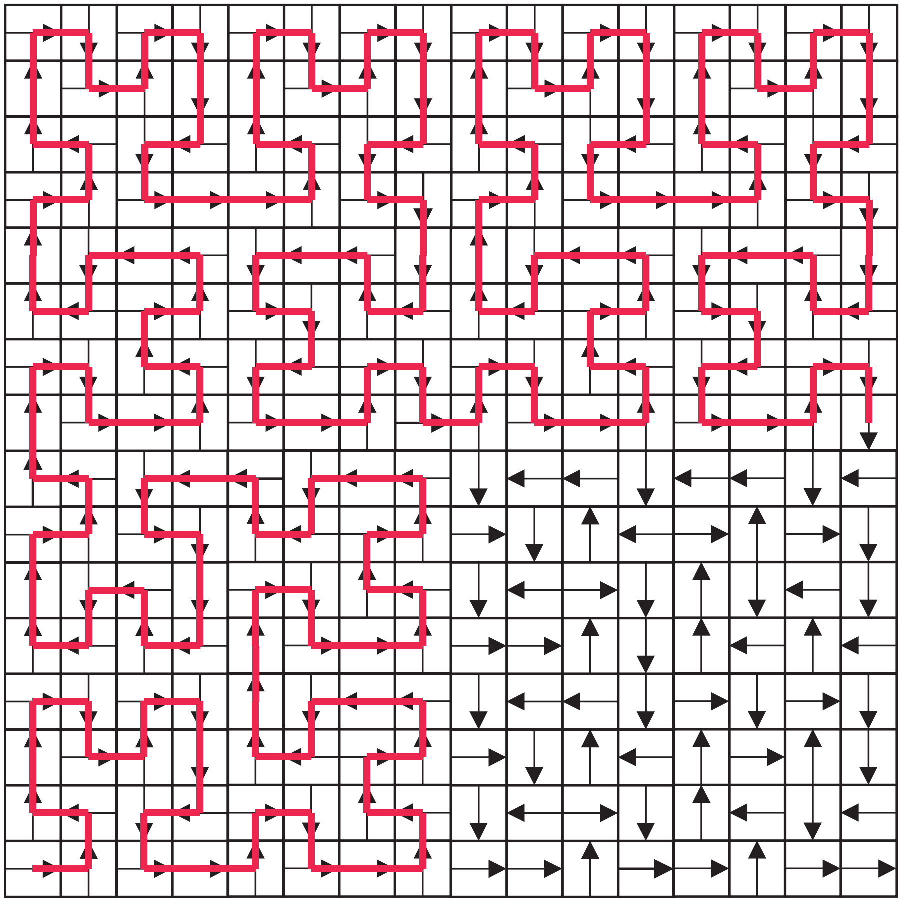
The construction of SNAKES is fairly complex and will be skipped. The paths that SNAKES forces when no tiling error is encountered have the shape of the well known plane-filling Hilbert-curve



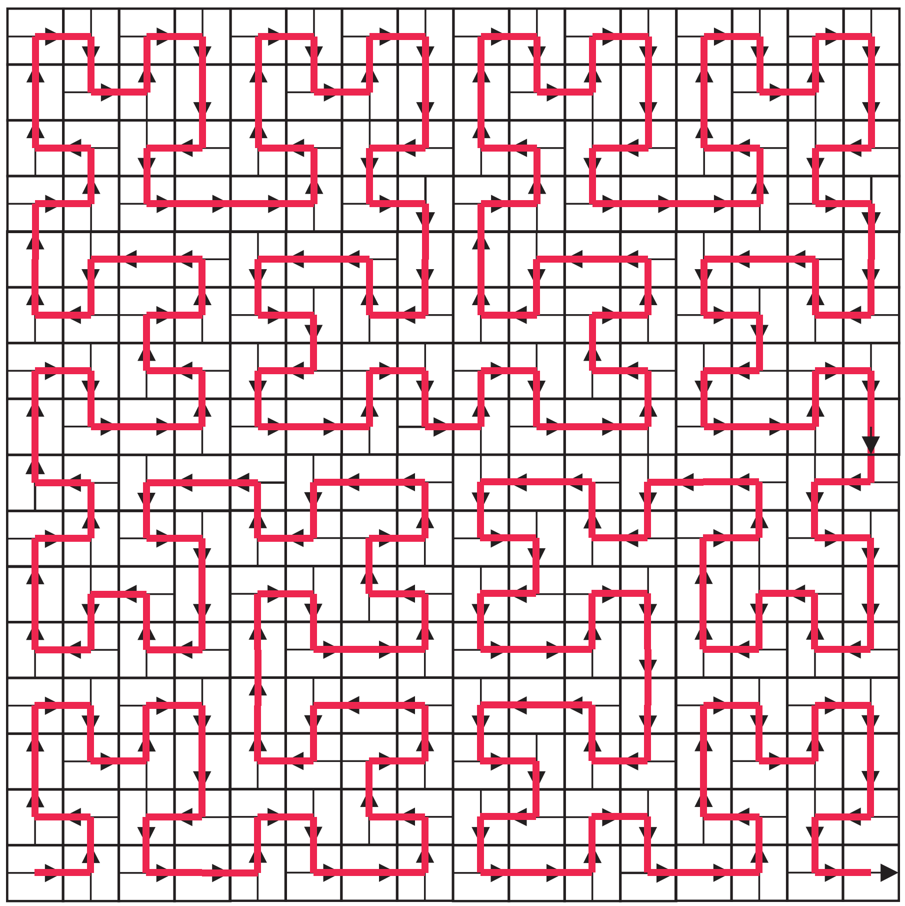
The construction of SNAKES is fairly complex and will be skipped. The paths that SNAKES forces when no tiling error is encountered have the shape of the well known plane-filling Hilbert-curve



The construction of SNAKES is fairly complex and will be skipped. The paths that SNAKES forces when no tiling error is encountered have the shape of the well known plane-filling Hilbert-curve



The construction of SNAKES is fairly complex and will be skipped. The paths that SNAKES forces when no tiling error is encountered have the shape of the well known plane-filling Hilbert-curve



Applications of Snakes

First application of Snakes: An example of a two-dimensional CA that is injective on periodic configurations but is not injective on all configurations.

Applications of SNAKES

First application of SNAKES: An example of a two-dimensional CA that is injective on periodic configurations but is not injective on all configurations.

Let G_P denote the restriction of CA function G into periodic configurations.

Among one-dimensional CA the following facts hold:

$$\begin{aligned} G \text{ injective} \quad &\Longleftrightarrow \quad G_P \text{ injective}, \\ G \text{ surjective} \quad &\Longleftrightarrow \quad G_P \text{ surjective}. \end{aligned}$$

Among one-dimensional CA the following facts hold:

$$G \text{ injective} \iff G_P \text{ injective},$$
$$G \text{ surjective} \iff G_P \text{ surjective}.$$

Among two-dimensional CA only these implications are easy:

$$G \text{ injective} \implies G_P \text{ injective},$$
$$G \text{ surjective} \impliedby G_P \text{ surjective}.$$

Among one-dimensional CA the following facts hold:

$$G \text{ injective} \iff G_P \text{ injective},$$

$$G \text{ surjective} \iff G_P \text{ surjective}.$$

Among two-dimensional CA only these implications are easy:

$$G \text{ injective} \implies G_P \text{ injective},$$

$$G \text{ surjective} \impliedby G_P \text{ surjective}.$$

The **Snake XOR** CA confirms that in 2D

$$G \text{ injective} \not\Longleftarrow G_P \text{ injective}.$$

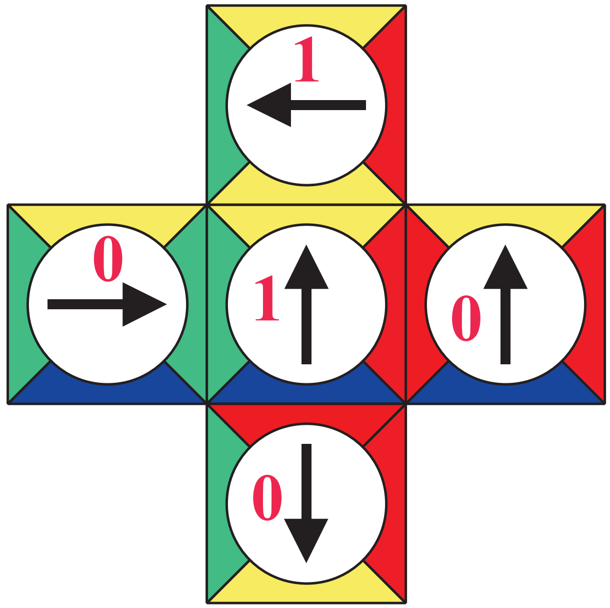
The state set of the CA is

$$S = \textsc{Snakes} \times \{0, 1\}.$$

(Each snake tile is attached a red bit.)

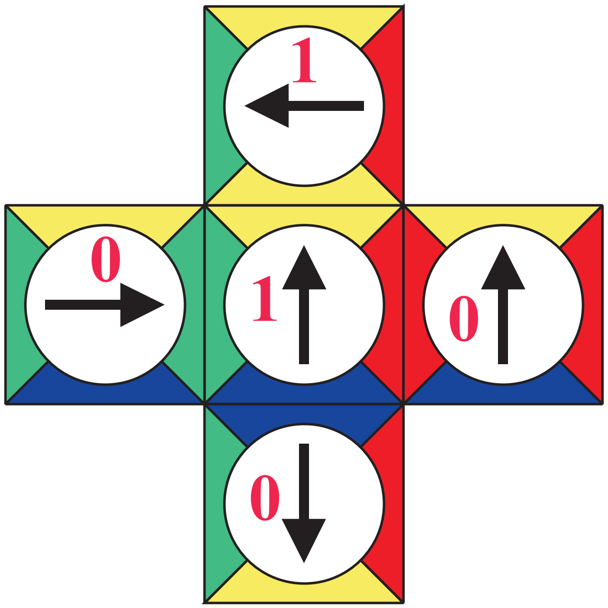
The local rule checks whether the tiling is valid at the cell:

- If there is a tiling error, no change in the state.



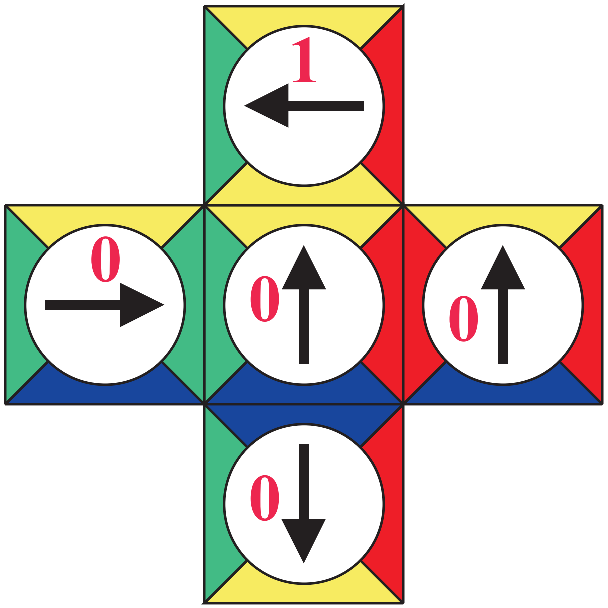
The local rule checks whether the tiling is valid at the cell:

- If there is a tiling error, no change in the state.
- If the tiling is valid, the cell is **active**: the bit of the neighbor next on the path is XOR'ed to the bit of the cell.



The local rule checks whether the tiling is valid at the cell:

- If there is a tiling error, no change in the state.
- If the tiling is valid, the cell is **active**: the bit of the neighbor next on the path is XOR'ed to the bit of the cell.



Snake XOR is not injective:

The following two configurations have the same successor: The SNAKES tilings of the configurations form the same valid tiling of the plane. In one of the configurations all bits are set to 0, and in the other configuration all bits are 1.

All cells are active because the tilings are correct. This means that all bits in both configurations become 0. So the two configurations become identical. The CA is not injective.

Snake XOR is injective on periodic configurations:

Suppose there are different periodic configurations c and d with the same successor. Since only bits may change, c and d must have identical SNAKES tiles everywhere. So they must have different bits 0 and 1 in some position $\vec{p}_1 \in \mathbb{Z}^2$.

Snake XOR is injective on periodic configurations:

Suppose there are different periodic configurations c and d with the same successor. Since only bits may change, c and d must have identical Snakes tiles everywhere. So they must have different bits 0 and 1 in some position $\vec{p}_1 \in \mathbb{Z}^2$.

Because c and d have identical successors:

- The cell in position $\vec{p}_1$ must be active, that is, the Snakes tiling is valid in position $\vec{p}_1$.
- The bits stored in the next position $\vec{p}_2$ (indicated by the direction) are different in c and d.

Snake XOR is injective on periodic configurations:

Suppose there are different periodic configurations c and d with the same successor. Since only bits may change, c and d must have identical SNAKES tiles everywhere. So they must have different bits 0 and 1 in some position $\vec{p}_1 \in \mathbb{Z}^2$.

Because c and d have identical successors:

- The cell in position $\vec{p}_1$ must be active, that is, the SNAKES tiling is valid in position $\vec{p}_1$.
- The bits stored in the next position $\vec{p}_2$ (indicated by the direction) are different in c and d.

Hence we can repeat the reasoning in position $\vec{p}_2$.

The same reasoning can be repeated over and over again. The positions $\vec{p}_1, \vec{p}_2, \vec{p}_3, \ldots$ form a path that follows the arrows on the tiles. There is no tiling error at any tile on this path.

But this contradicts the fact that the plane filling property of SNAKES guarantees that on periodic configuration every path encounters a tiling error. $\square$

Open problem: The implication

$$G \text{ surjective} \quad \stackrel{?}{\Longrightarrow} \quad G_P \text{ surjective}$$

is not known.

If every configuration has a pre-image, does every periodic configuration have a periodic pre-image ?

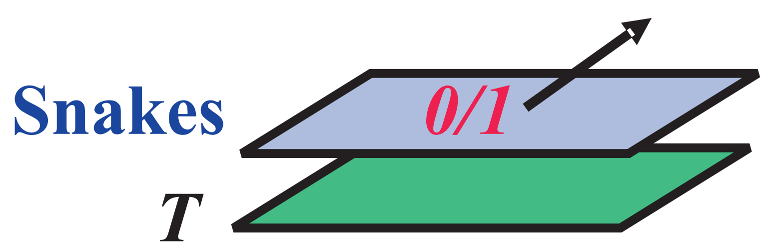
Second application of Snakes: It is undecidable to determine if a given two-dimensional CA is reversible.

Second application of SNAKES: It is undecidable to determine if a given two-dimensional CA is reversible.

The proof is a reduction from the tiling problem, using the tile set SNAKES.

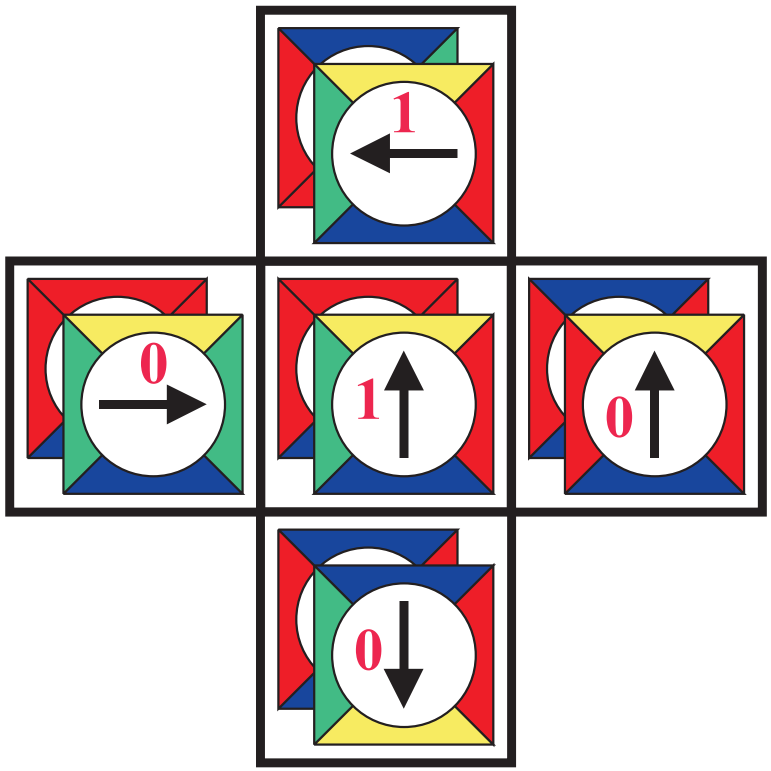
For any given tile set T we construct a CA with the state set

$$S = T \times \text{SNAKES} \times \{0, 1\}.$$



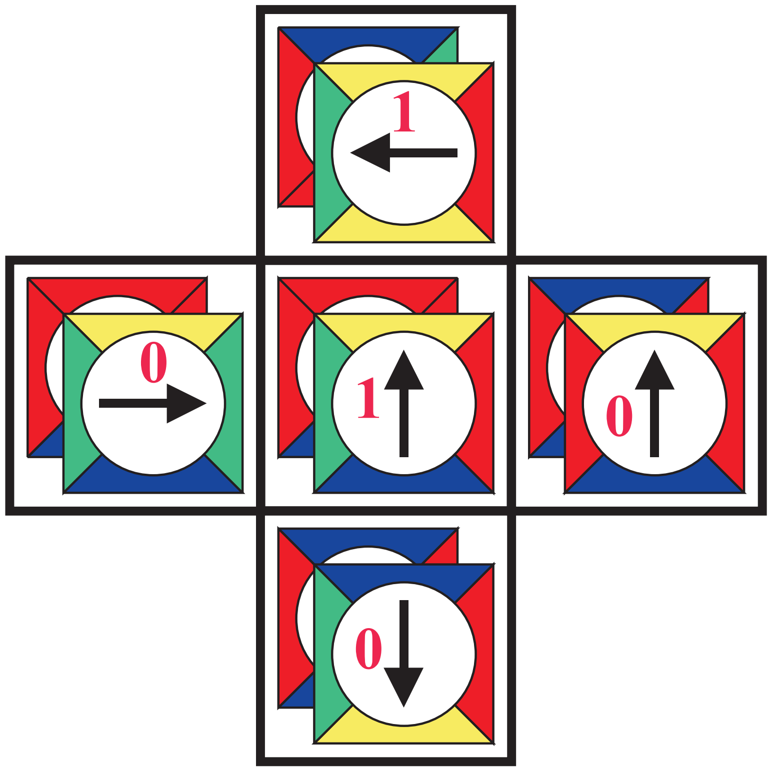
The local rule is analogous to **Snake XOR** with the difference that the correctness of the tiling is checked in both tile layers:

- If there is a tiling error then the cell is inactive.



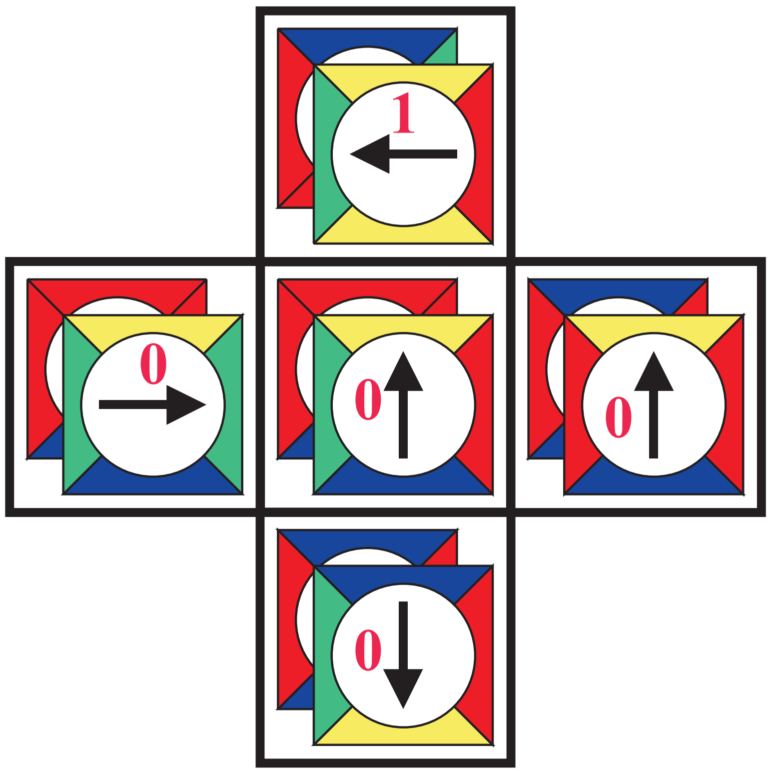
The local rule is analogous to **Snake XOR** with the difference that the correctness of the tiling is checked in both tile layers:

- If there is a tiling error then the cell is inactive.
- If both tilings are valid, the bit of the neighbor next on the path is XOR'ed to the bit of the cell.



The local rule is analogous to **Snake XOR** with the difference that the correctness of the tiling is checked in both tile layers:

- If there is a tiling error then the cell is inactive.
- If both tilings are valid, the bit of the neighbor next on the path is XOR'ed to the bit of the cell.



We can reason exactly as with **Snake XOR**, and show that the CA is reversible if and only if the tile set T does not admit a plane tiling:

We can reason exactly as with **Snake XOR**, and show that the CA is reversible if and only if the tile set T does not admit a plane tiling:

($\Longrightarrow$) If a valid tiling of the plane exists then we can construct two different configurations of the CA that have the same image under G. The SNAKES and the T layers of the configurations form the same valid tilings of the plane. In one of the configurations all bits are 0, and in the other configuration all bits are 1.

All cells are active because the tilings are correct. This means that all bits in both configurations become 0. So the two configurations become identical. The CA is not injective.

($\Longleftarrow$) Conversely, assume that the CA is not injective. Let c and d be two different configurations with the same successor. Since only bits may change, c and d must have identical SNAKES and T layers. So they must have different bits 0 and 1 in some position $\vec{p}_1 \in \mathbb{Z}^2$.

($\Longleftarrow$) Conversely, assume that the CA is not injective. Let c and d be two different configurations with the same successor. Since only bits may change, c and d must have identical SNAKES and T layers. So they must have different bits 0 and 1 in some position $\vec{p}_1 \in \mathbb{Z}^2$.

Because c and d have identical successors:

- The cell in position $\vec{p}_1$ must be active, that is, the SNAKES and T tilings are both valid in position $\vec{p}_1$.
- The bits stored in the next position $\vec{p}_2$ (indicated by the direction) are different in c and d.

($\Longleftarrow$) Conversely, assume that the CA is not injective. Let c and d be two different configurations with the same successor. Since only bits may change, c and d must have identical SNAKES and T layers. So they must have different bits 0 and 1 in some position $\vec{p}_1 \in \mathbb{Z}^2$.

Because c and d have identical successors:

- The cell in position $\vec{p}_1$ must be active, that is, the SNAKES and T tilings are both valid in position $\vec{p}_1$.
- The bits stored in the next position $\vec{p}_2$ (indicated by the direction) are different in c and d.

Hence we can repeat the reasoning in position $\vec{p}_2$.

The same reasoning can be repeated over and over again. The positions $\vec{p}_1, \vec{p}_2, \vec{p}_3, \ldots$ form a path that follows the arrows on the tiles. There is no tiling error at any tile on this path so the special property of SNAKES forces the path to cover arbitrarily large squares.

Hence T admits tilings of arbitrarily large squares, and consequently a tiling of the infinite plane. $\square$

Theorem: It is undecidable whether a given two-dimensional CA is injective.

Theorem: It is undecidable whether a given two-dimensional CA is injective.

An analogous (but simpler!) construction can be made for the surjectivity problem, based on the fact surjectivity is equivalent to pre-injectivity:

Theorem: It is undecidable whether a given two-dimensional CA is surjective.

Both problems are semi-decidable in one direction:

Injectivity is semi-decidable: Enumerate all CA G one-by one and check if G is the inverse of the given CA. Halt once (if ever) the inverse is found.

Non-surjectivity is semi-decidable: Enumerate all finite patterns one-by-one and halt once (if ever) an orphan is found.

Undecidability of injectivity implies the following:

There are some reversible CA that use von Neumann neighborhood but whose inverse automata use a very large neighborhood: There can be no computable upper bound on the extend of this inverse neighborhood.

Undecidability of injectivity implies the following:

There are some reversible CA that use von Neumann neighborhood but whose inverse automata use a very large neighborhood: There can be no computable upper bound on the extend of this inverse neighborhood.

Topological arguments $\Longrightarrow$ A finite neighborhood is enough to determine the previous state of a cell.

Computation theory $\Longrightarrow$ This neighborhood may be extremely large.

Undecidability of surjectivity implies the following:

There are non-surjective CA whose smallest orphan is very large: There can be no computable upper bound on the extend of the smallest orphan.

Undecidability of surjectivity implies the following:

There are non-surjective CA whose smallest orphan is very large: There can be no computable upper bound on the extend of the smallest orphan.

So while the smallest known orphan for Game-Of-Life is pretty big (109 cells), this pales in comparison with some other CA.

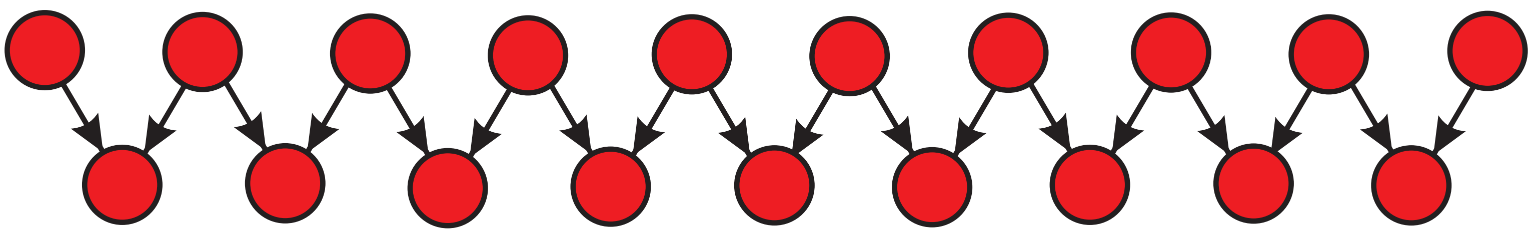
Both reversibility and surjectivity can be easily decided among one-dimensional CA:

Theorem (Amoroso, Patt 1972): It is decidable whether a given one-dimensional CA is injective (or surjective).

Best algorithms are based on de Bruijn -graphs.

We know the tight bound on the extend of the one-dimensional **inverse neighborhood**:

The neighborhood of a reversible CA with n states and the radius-$\frac{1}{2}$ neighborhood



consists of at most $n-1$ consecutive cells (Czeizler, Kari).

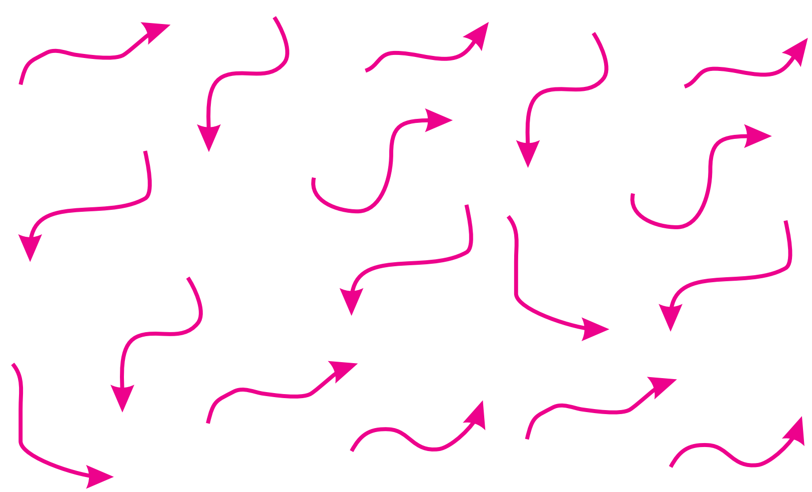
An upper bound on the length of the **smallest orphan** for a one-dimensional, radius-$\frac{1}{2}$, non-surjective CA with n states:

There is an orphan of length n^2. (Kari, Vanier, Zeume).

A CA G is called **periodic** if all configurations are temporally periodic. In this case, there is a positive integer n such that G^n is the identity function.

In the undecidability proof for reversibility we executed XOR along paths, and

- if tile set T does not admit a tiling then no infinite correctly tiled path exists.

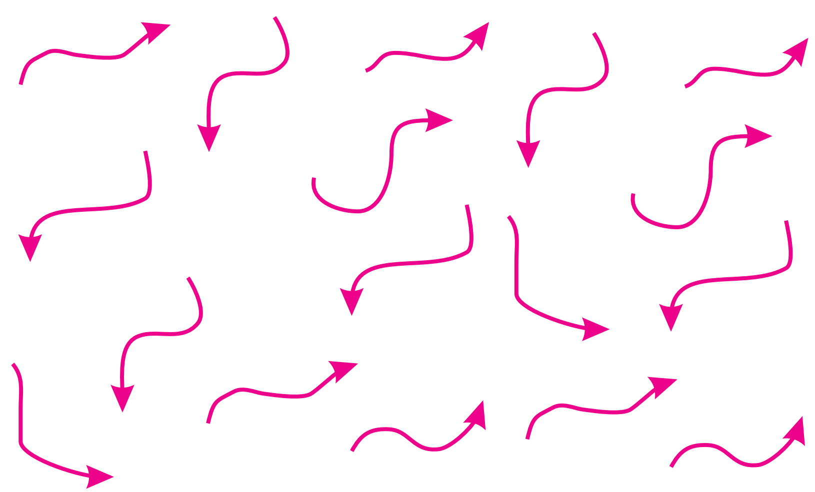


In the undecidability proof for reversibility we executed XOR along paths, and

- if tile set T does not admit a tiling then no infinite correctly tiled path exists.

By compactness then

- the lengths of valid paths in all configurations are bounded by some constant N.



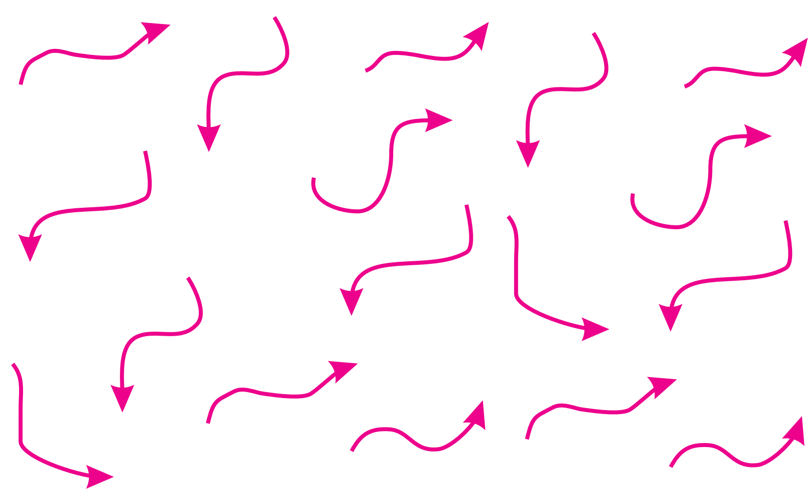
In the undecidability proof for reversibility we executed XOR along paths, and

- if tile set T does not admit a tiling then no infinite correctly tiled path exists.

By compactness then

- the lengths of valid paths in all configurations are bounded by some constant N.

$\Longrightarrow$ the CA is not only reversible but it is even periodic.



Hence we have (G.Theyssier M.Sablik)

Theorem: It is undecidable whether a given two-dimensional CA is periodic.

Or even

Theorem: 2D Periodic CA and non-reversible CA are recursively inseparable.

Hence we have (G.Theyssier M.Sablik)

Theorem: It is undecidable whether a given two-dimensional CA is periodic.

Or even

Theorem: 2D Periodic CA and non-reversible CA are recursively inseparable.

It turn out that periodicity is also undecidable among one-dimensional CA (Kari, Ollinger):

Theorem: It is undecidable whether a given one-dimensional CA is periodic.

Conclusions

	1D	2D
Nilpotency	**U**	**U**
Periodicity	**U**	**U**
$\exists$ fixed point	**D**	**U**
Injectivity	**D**	**U**
Surjectivity	**D**	**U**
(Positive) expansivity	?	N/A
One-sided expansivity	**U**	N/A