Cellular Automata and Tilings II

Jarkko Kari

Department of Mathematics, University of Turku, Finland
TUCS(Turku Centre for Computer Science), Turku, Finland
(1) More on determinism
   - Robinson $\implies$ NW-deterministic aperiodic
   - Variants of expansivity

(2) Snake tiles
   - Robinson $\implies$ Snakes
   - Undecidability results from Snakes
   - Finite entropy 2D CA
   - The snake tiling problem
Robinson’s aperiodic tile set
Robinson’s aperiodic tile set consists of crosses and arms.

The arrows indicate the matching rules: arrows must continue uninterrupted across tile boundaries.
Robinson’s aperiodic tile set consists of crosses and arms.

All tiles may be rotated so each tile comes in four orientations. (28 variants in total.)
Robinson’s aperiodic tile set consists of crosses and arms.

Each tile has a black (incoming or outgoing) arrow on each side, and possibly some red side arrows.
Each cross has two side arrows and the direction of these side arrows determines the orientation of the cross.
The black arrow through an arm tile is its principal arrow and it indicates the orientation of the arm.

Any black arrow may be accompanied by a red side arrow, on either side, with the following exception:

- A pair of incoming side arrows cannot be towards the tail of the principal arrow:
The tiles are paired with parity tiles

in such a way that

- **1** is only paired with crosses,
- **2** is only paired with vertically oriented arms,
- **3** is only paired with horizontally oriented arms.
- **4** can be paired with anything.

Each Robinson tile has then two possible parities (4 and either 1,2 or 3), so the total tile count is 56.
Tiling forced by parity tiles
Odd-odd positions (parity tile 1) are forced to contain crosses.
Odd-odd positions (parity tile 1) are forced to contain crosses. Vertical and horizontal arms are forced between them. Tiles in even-even positions (parity tile 4) can be chosen freely.
For any $n \geq 1$, we recursively build **special squares** of size $(2^n - 1) \times (2^n - 1)$:

- The special squares have outward arrows along the sides.
- At the centers of two sides there are red side arrows.
- Corners are parity 1 crosses.
- The square can be rotated in four possible orientations.

We call $n$ the **level** of the square.
- Level 1 squares are crosses with parity tile 1. There are four possible orientations of the cross.
- Level $n + 1$ square consists of four level $n$ squares facing each other, with a cross in the middle.
- Level $n+1$ square consists of four level $n$ squares facing each other, with a cross in the middle.

- the cross in the center can be oriented arbitrarily: this determines the orientation of the square.
- Level $n + 1$ square consists of four level $n$ squares facing each other, with a cross in the middle.
- the cross in the center can be oriented arbitrarily: this determines the orientation of the square.
- rows of arms radiate out from the center.
The special square is correctly tiled. The only critical positions are the tiles where the side arrows of the smaller squares meet: The side arrows are towards the head of the principal arrow as required.
Since special squares are correctly tiled, and they are unbounded in size, we see that Robinson’s tiles admit a valid tiling of the plane.
Since special squares are correctly tiled, and they are unbounded in size, we see that Robinson’s tiles admit a valid tiling of the plane.

But not only are special squares possible: they are mandatory in any valid tiling.

**Lemma:** Consider a valid tiling of the plane. Every cross in odd-odd position (i.e., with parity tile 1) belongs to a unique special square of level $n$, for all $n = 1, 2, \ldots$
Since special squares are correctly tiled, and they are unbounded in size, we see that Robinson’s tiles admit a valid tiling of the plane.

But not only are special squares possible: they are mandatory in any valid tiling.

**Lemma:** Consider a valid tiling of the plane. Every cross in odd-odd position (i.e., with parity tile 1) belongs to a unique special square of level $n$, for all $n = 1, 2, \ldots$

**Proof:** Induction on level $n$.

- When $n = 1$ the cross itself is the unique level 1 square
Suppose then that the claim is true for level $n$, and consider level $n + 1$ and an arbitrary cross in an odd-odd position.
The cross belongs to a unique level $n$ square.
Suppose w.l.o.g. that the square is oriented north-east.
What is the tile outside the north-east corner of the square?
If it were an arm, then its west or south edge would have an incoming arrow.
A sequence of arms directed towards the tile would be forced.
But then an arm with incoming side arrows at the wrong end would result.
Hence the tile must be a cross. It is in even-even position (parity tile 4).
The tile to the north-west has parity 1, so it is a cross.

By inductive hypothesis, it belongs to some special square of level $n$. 
Two level $n$ squares cannot intersect (as this would contradict the uniqueness for the tiles in the intersection).
The tile to the east (marked "X") cannot have an arrow pointing south, so it cannot be on the south boundary of the level $n$ square.
The level $n$ square must therefore be located on top of the first one.
The side arrows force the two level $n$ squares to face each other.
The same reasoning implies four level $n$ squares facing each other.
The corridors between the squares can only be arms. Hence we have obtained a level $n + 1$ special square that contains the original green tile.
Uniqueness of the square is obvious since the orientation of the level $n$ square that contains the green tile determines where the center of the level $n+1$ square is.
**Theorem:** Robinson’s tile set is aperiodic.

**Proof:** Every valid tiling contains level $n$ squares for all $n$, and therefore crosses followed by arbitrarily long sequences of arms next to them. There can be no horizontal period.
Recall that a tile set $T$ is **NW-deterministic** if the north and west color determines tiles uniquely.

For example

```
A | B | C | D
```

is NW-deterministic.
Robinson’s tiles are almost NW-deterministic:
Robinson’s tiles are **almost** NW-deterministic:

- One can NW-deterministically recognize whether a tile is a cross or an arm,
Robinson’s tiles are almost NW-deterministic:

- One can NW-deterministically recognize whether a tile is a cross or an arm,
- The orientation of a cross is NW-deterministic,
Robinson’s tiles are almost NW-deterministic:

- One can NW-deterministically recognize whether a tile is a cross or an arm,
- The orientation of a cross is NW-deterministic,
- Parity-tiles are deterministic from all sides,
Robinson’s tiles are **almost** NW-deterministic:

- One can NW-deterministically recognize whether a tile is a cross or an arm,
- The orientation of a cross is NW-deterministic,
- Parity-tiles are deterministic from all sides,
- Red side arrows are NW-deterministic, once the black arrows are known.
The only source of non-determinism is in identifying the direction of an arm, when both the north and the west sides have an incoming arrow:
Observation: In the special squares, horizontal and vertical arms alternate on each diagonal.
**Observation:** In the special squares, horizontal and vertical arms alternate on each diagonal.

Hence, a layer of diagonal signals can identify whether an arm is horizontal or vertical.
Observation: In the special squares, horizontal and vertical arms alternate on each diagonal.

To use edge colors (instead of corner colors) the signal can be zigzagged.
Observation: In the special squares, horizontal and vertical arms alternate on each diagonal.

We easily obtained a NW-deterministic, aperiodic tile set.
**Remark:** The classical proof for the undecidability of the tiling problem uses a tile set such as Robinson’s tiles to define areas for Turing machine simulations.

It is straightforward to adapt this method to the NW-deterministic version of Robinson’s tiles to prove the undecidability of the tiling problem among NW-deterministic tile sets.
The 4-way deterministic case is trickier, but we have (V.Lukkarila)

**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.
The 4-way deterministic case is trickier, but we have (V. Lukkarila)

**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.

Like what??
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:

![Diagram](image)

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 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 paths that **Snakes** forces when no tiling error is encountered have the shape of the well known plane-filling Hilbert-curve
The paths that \textbf{SNAKES} forces when no tiling error is encountered have the shape of the well known plane-filling Hilbert-curve
The paths that \textbf{SNAKES} forces when no tiling error is encountered have the shape of the well known plane-filling Hilbert-curve
The paths that **Snakes** forces when no tiling error is encountered have the shape of the well known plane-filling Hilbert-curve.
The paths that SNAKES forces when no tiling error is encountered have the shape of the well known plane-filling Hilbert-curve
The paths that **Snakes** forces when no tiling error is encountered have the shape of the well known plane-filling Hilbert-curve.
The paths that Snakes forces when no tiling error is encountered have the shape of the well known plane-filling Hilbert-curve
The paths that **Snakes** forces when no tiling error is encountered have the shape of the well known plane-filling Hilbert-curve.
Snakes are built using Robinson’s tiles.

Directions are attached to crosses at odd-odd positions (parity tile 1). The actual tiles are then $2 \times 2$ blocks of Robinson’s tiles.

The Hilbert-curve is forced through special squares.
Hilbert-curve comes in four orientations, generated by substitutions
Hilbert-curve comes in four orientations, generated by substitutions
Hilbert-curve comes in four orientations, generated by substitutions
Hilbert-curve comes in four orientations, generated by substitutions
We want to force the Hilbert-curve through the special squares.
The center cross (and the arrows out of the cross) is labeled with a symbol that identifies the orientation of the curve through the square.
The four smaller squares at the corners should be labeled as given by the Hilbert substitution rule.
The four smaller squares at the corners should be labeled as given by the Hilbert substitution rule.
This is forced by limiting the allowed labels at the arms with incoming red side arrows.
This way the shapes are uniquely propagated to all crosses. The direction to be attached in each cross (in odd-odd position) can be uniquely deduced from these labels.
Our explanation was somewhat over-simplified:

It turns out that is more convenient to include in the labels the directions of entering and leaving the square, e.g.

\[ \Rightarrow \]

\[ \Rightarrow 12 \text{ labels instead of } 4. \]

Bonus: the direction of the path is directly given by the label.
Our explanation was somewhat over-simplified:

It turns out that is more convenient to include in the labels the directions of entering and leaving the square, e.g.

$$\Rightarrow 12\text{ labels instead of 4.}$$

Bonus: the direction of the path is directly given by the label.
Such construction clearly provides directed tiles such that

• in any correct tiling of the plane the directions define
  plane filling path(s).
Such construction clearly provides directed tiles such that

- in any correct tiling of the plane the directions define plane filling path(s).

**But this is not enough.** We want that even if there may be tiling errors outside a path, the path must be plane-filling as long as there are no tiling errors along the path.

It turns out that this can be obtained, using the facts that the Robinson’s tiles force the special squares locally, and the special squares get uniquely assembled along the Hilbert curve.
Some additional labeling is still needed (details skipped) because in the Hilbert curve the transition from one quadrant to the next does not happen always near the center cross:
Some additional labeling is still needed (details skipped) because in the Hilbert curve the transition from one quadrant to the next does not happen always near the center cross:
In the end we have **Snakes** with the desired plane-filling property:

- Any infinite path where colors match between all neighboring tiles along the path is plane filling: it covers arbitrarily large squares.

- There exists a valid tiling of the plane. (In fact, a tiling exists where a single bi-infinite path covers all cells.)
In the end we have **Snakes** with the desired plane-filling property:

- Any infinite path where colors match between all neighboring tiles along the path is plane filling: it covers arbitrarily large squares.

- There exists a valid tiling of the plane. (In fact, a tiling exists where a single bi-infinite path covers all cells.)

T. Meyerovitch also noticed the following:

- A configuration can contain at most 4 disjoint infinite paths without tiling errors.
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:

\[ G \text{ injective } \iff G_P \text{ injective}, \]
\[ G \text{ surjective } \iff G_P \text{ surjective}. \]
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:

\[ G \text{ injective} \implies G_P \text{ injective}, \]
\[ G \text{ surjective} \iff G_P \text{ surjective}. \]
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:

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

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

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

$$G \text{ injective } \not\iff G_P \text{ injective}.$$
The state set of the CA is

\[ S = \text{Snakes} \times \{0, 1\}. \]

(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 \textbf{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. □
**Open problem:** The implication

\[ G \text{ surjective } \implies G_P \text{ surjective} \]

is not known.

If every configuration has a pre-image, does every periodic configuration have a periodic pre-image?
Second application of \textbf{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:

$(\Rightarrow)$ 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.
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 $\vec{p}_1 \in \mathbb{Z}^2$. 
(←) 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 $\text{Snakes}$ 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 $\text{Snakes}$ 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 $d$. 
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 $\text{Snakes}$ forces the path to cover arbitrarily large squares.

Hence $T$ admits tilings of arbitrarily large squares, and consequently a tiling of the infinite plane. □
**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** $\implies$ A finite neighborhood is enough to determine the previous state of a cell.

**Computation theory** $\implies$ 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 (113 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 $N$.

$\implies$ 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 turns out that periodicity is also undecidable among one-dimensional CA (Kari, Ollinger):

**Theorem:** It is undecidable whether a given one-dimensional CA is periodic.
A CA $G$ is called **open** if $G(U)$ is open for all open $U$. Reversible CA are trivially open, and open CA have to be surjective (because in non-surjective CA Garden-Of-Eden configurations are dense.)
A CA $G$ is called **open** if $G(U)$ is open for all open $U$. Reversible CA are trivially open, and open CA have to be surjective (because in non-surjective CA Garden-Of-Eden configurations are dense.)

The following one-dimensional, radius-$\frac{1}{2}$, three state CA is not open even though it is **right permutive**:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

In our proofs we only needed the fact that XOR is right permutive. So we can do the same proof, with XOR replaced by the rule above.
Then:

- If $T$ does not admit a tiling then the CA is periodic.
- If $T$ admits a tiling the CA is not open. (Follows from the non-openness of the 1D CA, after a small proof.)

We have (C.Zinoviadis)

**Theorem**: It is undecidable whether a given 2D CA is open.

Or even

**Theorem**: 2D Periodic CA and non-open CA are recursively inseparable.
In 2D all classes are undecidable.

In 1D periodicity is undecidable, other classes are decidable.
Next up:

**Snake XOR** has finite but non-zero topological entropy (T. Meyerovitch), refuting an earlier conjecture that all 2D CA have either infinite entropy or zero entropy.
Snake XOR executes a one-dimensional XOR along valid paths. Any configuration by Snakes contains at most four infinite valid paths. All other paths are finite.
The tiling component and the finite paths contribute nothing to the entropy

\[ \implies H(\text{Snake XOR}) \leq 4 \times H(\text{XOR}). \]

But \textbf{Snakes} admits a tiling with a valid path

\[ \implies H(\text{Snake XOR}) \geq H(\text{XOR}). \]

\textbf{Theorem:} Snake XOR has nonzero finite entropy.
The **Snake tiling problem** was first raised in 1994 (Y Etzion-Petruschka, D.Harel, D.Myers), and later again in the context of Wang tile model of self-assembly by E.Winfree.
In Winfree’s model, tiles assemble one-by-one into a growing pattern of tiles.
In Winfree’s model, tiles assemble one-by-one into a growing pattern of tiles.
In Winfree’s model, tiles assemble one-by-one into a growing pattern of tiles.
In Winfree’s model, tiles assemble one-by-one into a growing pattern of tiles.
In Winfree’s model, tiles assemble one-by-one into a growing pattern of tiles.
In Winfree’s model, tiles assemble one-by-one into a growing pattern of tiles.
In Winfree’s model, tiles assemble one-by-one into a growing pattern of tiles.
In Winfree’s model, tiles assemble one-by-one into a growing pattern of tiles.
There are two natural choices of matching rules:

- Under **strong matching rules** each new tile must match in color with all its neighbors that have been assembled before it.
There are two natural choices of matching rules:

- Under **strong matching rules** each new tile must match in color with all its neighbors that have been assembled before it.

- Under **weak matching rules**, it is enough for a tile to match with one previous tile.
There are two natural choices of matching rules:

- Under **strong matching rules** each new tile must match in color with all its neighbors that have been assembled before it.

- Under **weak matching rules**, it is enough for a tile to match with one previous tile.
There are two natural choices of matching rules:

- Under **strong matching rules** each new tile must match in color with all its neighbors that have been assembled before it.

- Under **weak matching rules**, it is enough for a tile to match with one previous tile.

(More generally, Winfree associates strength values to colors, and allows an attachment of a tile if the sum of the strengths exceeds a given threshold.)
A pattern is a **terminal assembly** if no more tiles can be added.

A tile set admits an **unbounded assembly** if an infinite sequence of tile additions without ever reaching a terminal assembly is possible.
A pattern is a **terminal assembly** if no more tiles can be added.

A tile set admits an **unbounded assembly** if an infinite sequence of tile additions without ever reaching a terminal assembly is possible.

**Decision problem:** Does a given tile set admit an unbounded assembly (under strong/weak matching rules)?
A moment of thought...
A moment of thought... reveals that

- an unbounded assembly is possible if and only if the tiles admit a tiling of a (bi-infinite) snake.

A **snake** is an injective function \( s : \mathbb{Z} \rightarrow \mathbb{Z}^2 \) that is continuous in the sense that \( s(i) \) and \( s(i + 1) \) are neighbors for all \( i \).
If the assembly uses strong matching rules then the snake must be tiled so that any neighboring tiles of the snake must match. We call this a **strong snake**.
• If the assembly uses strong matching rules then the snake must be tiled so that any neighboring tiles of the snake must match. We call this a strong snake.

• If the assembly uses weak matching rules then it is enough that the consecutive tiles along the snake match. This is a weak snake.
• If the assembly uses strong matching rules then the snake must be tiled so that any neighboring tiles of the snake must match. We call this a **strong snake**.

• If the assembly uses weak matching rules then it is enough that the consecutive tiles along the snake match. This is a **weak snake**.

**Snake tiling problems:** Does a given finite set of Wang tiles admit a strong/weak snake?

**Theorem (Adleman et.al.):** The snake tiling problems (both variants) are undecidable.
In the proof we reduce the following undecidable problem:

**Directed snake tiling problem:** Does a given set of directed tiles admit a strong tiling of the snake in such a way that the direction of each tile points to the next tile along the snake.
In the proof we reduce the following undecidable problem:

**Directed snake tiling problem:** Does a given set of directed tiles admit a strong tiling of the snake in such a way that the direction of each tile points to the next tile along the snake.

Not ok
In the proof we reduce the following undecidable problem:

**Directed snake tiling problem:** Does a given set of directed tiles admit a strong tiling of the snake in such a way that the direction of each tile points to the next tile along the snake.

Undecidability of this problem is a direct application of the methods discussed earlier.
Let $D$ be a given set of directed tiles, called **macro-tiles**.

For each $d \in D$ we build a sequence of undirected mini-tiles. The mini-tiles are colored with unique colors that force them to form a motif: a finite snake that goes around the the four edges of the macro-tile $d$:

macro-tile $d$  
Corresponding motif of mini-tiles
Since the entry direction into a tile is not specified by the arrow, we need three motifs for each $d \in D$, one for every possible entry direction:

Left entry  Bottom entry  Right entry
Building a motif
Each mini-tile of a motif has two edges with a color that does not match any other tile, and the other two edges match a unique mini-tiles that force the formation of the motif.
Each motif matches other motifs only at the two "free" ends.
Motifs are glued to each other at their free ends. Motif can follow another motif, as long as its entry direction matches the exit direction of the previous motif.
Motifs are glued to each other at their free ends. Motif can follow another motif, as long as its entry direction matches the exit direction of the previous motif.

To prevent gluing two ends of entry, or two ends of exit together, we color each free end of the motif with a color that identifies the direction of that entry or exit (four colors for four possible directions N, E, S and W), e.g.
By gluing infinitely many motifs from their free ends we obtain infinite snakes that exactly correspond to infinite paths of macro-tiles specified by arrows:
By gluing infinitely many motifs from their free ends we obtain infinite snakes that exactly correspond to infinite paths of macro-tiles specified by arrows:
By gluing infinitely many motifs from their free ends we obtain infinite snakes that exactly correspond to infinite paths of macro-tiles specified by arrows:
So far the motifs have no constraints corresponding to edge coloring of the macro-tiles. To simulate the colors, the motifs are bent to form one **bump** or **dent** on each side of the motif.
The N and E side of each motif contains one bump, and the S and W side contains one dent:
The exact position of the bump or the dent encodes the color of the edge in the corresponding macro-tile. Each color corresponds to a unique position.
The exact position of the bump or the dent encodes the color of the edge in the corresponding macro-tile. Each color corresponds to a unique position.

If the colors of two adjacent macro-tiles match then the bump exactly fits inside the dent:
The exact position of the bump or the dent encodes the color of the edge in the corresponding macro-tile. Each color corresponds to a unique position.

But if the colors do not match then the two motifs would overlap, which is not possible:
We can make the following simple reasoning:

(⇒) Assume that the macro-tiles admit a strongly valid directed tiling of an infinite snake. If we replace each macro-tile by its motif we obtain an infinite snake of mini-tiles. The mini-tiles correctly tile this snake (even in the strong sense because the snake never returns to touch an earlier tile.)
We can make the following simple reasoning:

(⇒) Assume that the macro-tiles admit a strongly valid directed tiling of an infinite snake. If we replace each macro-tile by its motif we obtain an infinite snake of mini-tiles. The mini-tiles correctly tile this snake (even in the strong sense because the snake never returns to touch an earlier tile.)

(⇐) Conversely, assume that the mini-tiles admit a weak tiling of an infinite snake. The mini-tiles where designed in such a way that the snake must consist of a sequence of motifs. By replacing each motif by the macro-tile it represents, we obtain a directed tiling of an infinite snake by the macro-tiles. This tiling is valid even in the strong sense, because neighboring motifs must match even if they are not consecutive in the snake.
Hence we conclude that the strong and the weak snake tiling problems are undecidable.
Hence we conclude that the strong and the weak snake tiling problems are undecidable.

The motif construction was needed to translate the color constraints into purely geometric form, taking advantage of the fact that the snake is not allowed to overlap or intersect itself.
Hence we conclude that the strong and the weak snake tiling problems are undecidable.

The motif construction was needed to translate the color constraints into purely geometric form, taking advantage of the fact that the snake is not allowed to overlap or intersect itself.

The negative instances of the snake tiling problems have semi-algorithms: if a tile set does not admit a valid tiling of an infinite snake, then there is number $n$ so that no finite snake of length $n$ can be tiled. There are only a finite number of ways to tile finite snakes of length $n$, so we can exhaustively try all of them one-by-one.